HALMSTAD
UNIVERSITY

Lab 2

DT4015 - Data Communications

Haron Obaid and Fabian Henrysson

Part 1: Simple Packet Forwarding (20%)

Checkpoints 1

1. Transmissions and receptions for periodic message generation.

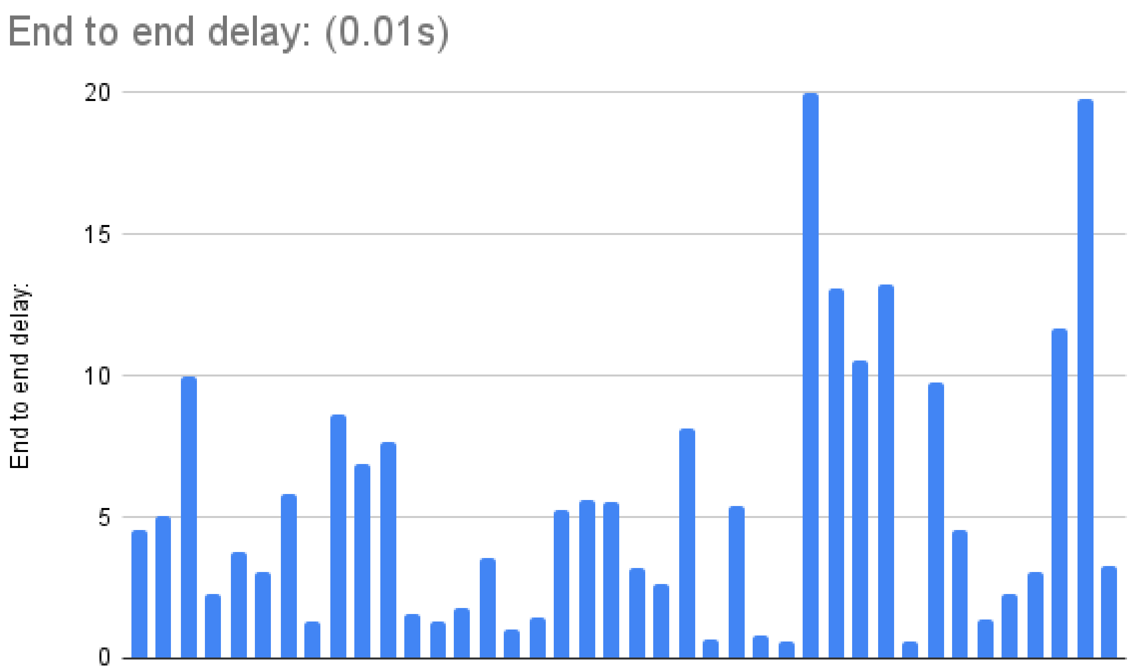
Node:	Sent:	Received:
0	201	0
1	200	200
2	200	199
3	199	199
4	199	199
5	0	199

We implemented the assignment using the as the lab sheet instructed. The message handler checks if the msg has arrived to either node 0 or 5. If it is node 0 we schedule a new message 1 second later, if it's at node 5 we increase and emit the reception signal and delete the message. For every other node other than 0 and 5 we forward the message to the next node and emit the reception/transmission signals.

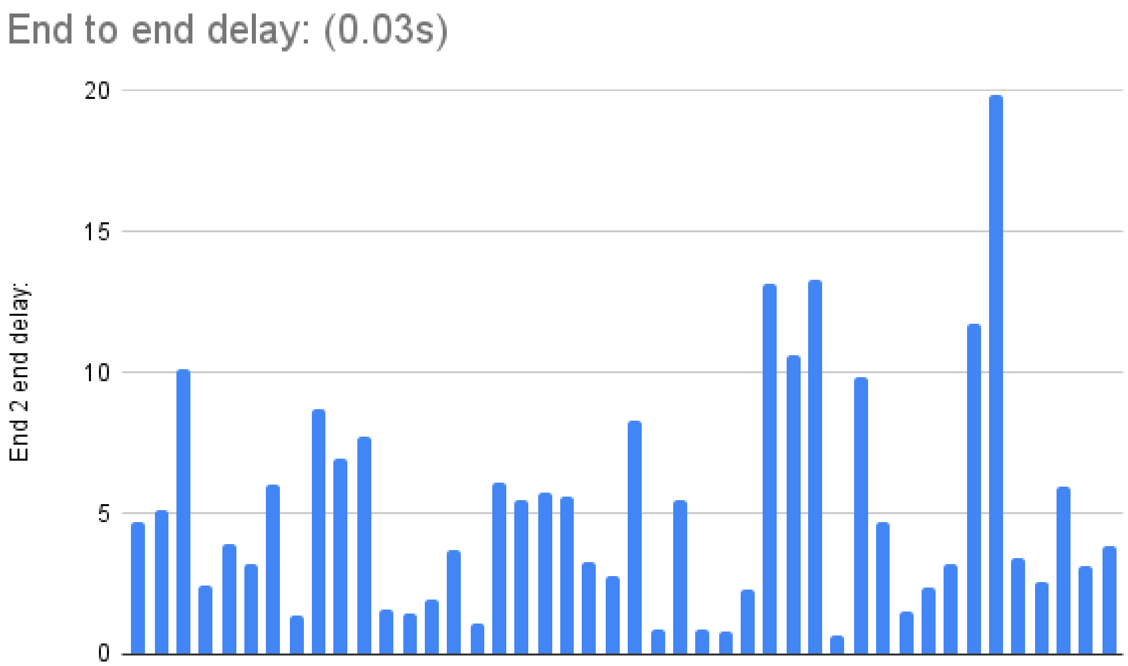
The test data shows that we send a message just about every second. The run time was set to 200s, and we were able to send 201 messages from node 0 and receive 199 messages in node 5 during that time. Thus, proving that our scheduling works as intended.

2. Statistics for end-to-end delay with random times between transmissions and random processing delays.

Transmission Delay (s):	Average End to end delay:	Standard Deviation:
0.01	5.520161225	4.87390357
0.03	5.167433894	4.090400021
0.05	5.324596231	4.105815097

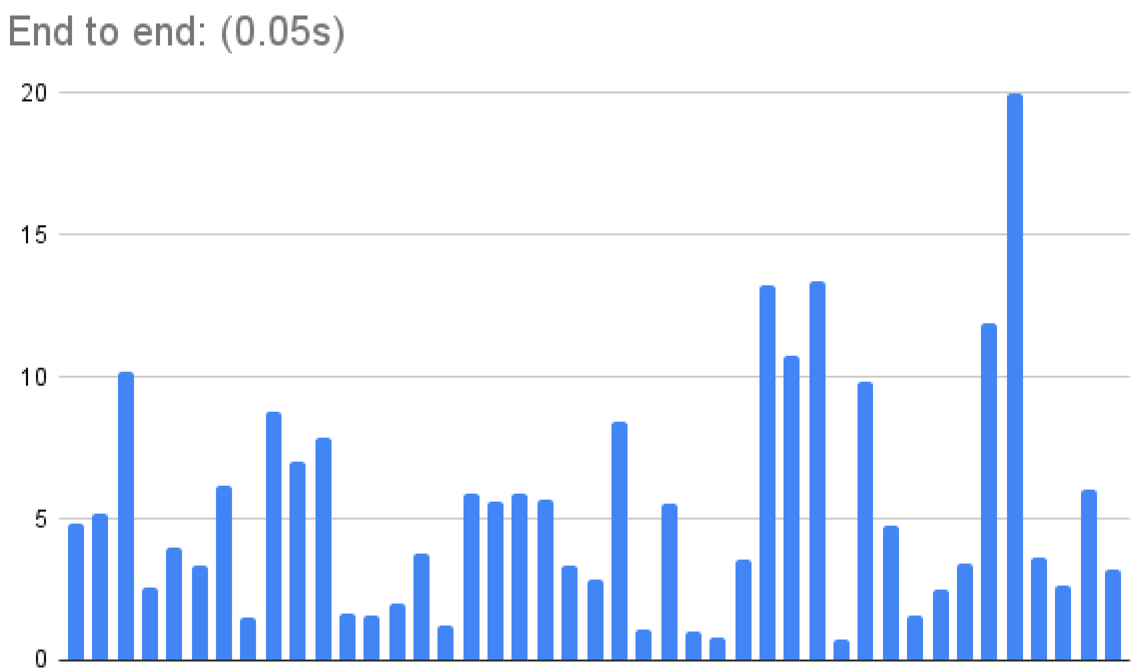


End to end delay: (0.01s)
End to end delay:
20
15
10
5
0



End to end delay: (0.03s)
End 2 end delay:
20
15
10
5
0

hh.se



From the grafs we can see that the end to end delay vary alot. This proves that the random timedelay is working as intended.

We implemented the randomised time using a random exponential with average 5s as the delayTime. When implementing forwardMessage we decided to use sendDelayed() to handle the transmission delay instead of implementing a Event based solution as we did in Lab 1.

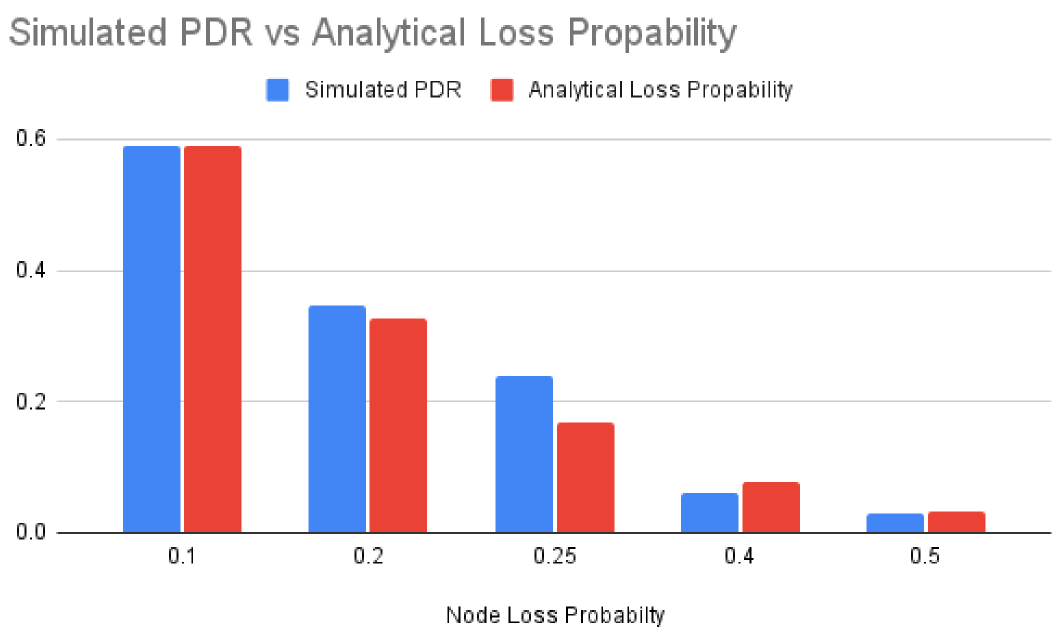
Part 2: Network Reliability with a Single Path (20%)

Checkpoints 2

1. Analytical calculation and simulation results for Packet-delivery Ratio.

P_{Success} = p_{AB} ∧ p_{BC} ∧ p_{CD} ∧ p_{DE} ∧ p_{EF}

Loss Probability:	Simulated PDR:	Analytical Loss Probability (P_{Success}):
0.1	0.5886366992	0.59049
0.2	0.3460724728	0.32768
0.25	0.2398465644	0.16807
0.4	0.06170421912	0.07776
0.5	0.02962991189	0.03125



For the most part we get a result of ±2%. We got a outlier result for loss probability = 0.25 where we got ±7.5%. This would be minimized if we ran the simulation for longer.

2. Network reliability metrics (PDR, IPG) for different probability combinations.

- Link Loss Probabilities: 0.5, 0.4, 0.35, 0.25, 0.2, 0.1

Loss Probability	Sent	Received	PDR	IGG(s)	IPG(s)
Run 0	162	23	0.1419753086	1.490495782	10.20686668
Run 1	167	21	0.125748503	1.438953237	11.39604942
Run 2	158	19	0.1202531646	1.525486324	10.53840244
Run 3	161	22	0.1366459627	1.490708327	9.907174206
Run 4	164	21	0.1280487805	1.465609718	11.85472933
Total/Average	812	106	0.1305343439	1.482250677	10.78064441

- Link Loss Probabilities: 0.1, 0.2, 0.1, 0.2, 0.1, 0.2

Loss Probability	Sent	Received	PDR	IGG	IPG
Run 0	160	80	0.5	1.500936687	2.991387047
Run 1	164	72	0.4390243902	1.470239108	3.146523433
Run 2	161	79	0.4906832298	1.495773748	3.053917867
Run 3	162	79	0.487654321	1.483413643	3.005804254
Run 5	163	73	0.4478527607	1.473913854	3.312343728
Total/Average	810	383	0.4730429404	1.484855408	3.101995266

- Link Loss Probabilities: 0.3, 0.2, 0.1, 0.3, 0.2, 0.1

Loss Probability:	Sent	Received	PDR	IGG	IPG
Run 0	160	45	0.28125	1.022582538	5.038073324
Run 1	159	41	0.2578616352	1.503779386	5.555090785
Run 2	164	56	0.3414634146	1.459278796	4.134780949
Run 3	163	44	0.2699386503	1.465167843	5.063732413
Run 4	162	46	0.2839506173	1.474028309	5.185561861
Total/Average	808	232	0.2868928635	1.384967374	4.995447866

- Link Loss Probabilities: 0.25, 0.25, 0.4, 0.1, 0.3, 0.01.

Loss Probability	Sent	Received	PDR	IGG	IPG
Run 0	161	34	0.2111801242	1.49016211	7.00978469
Run 1	160	34	0.2125	1.492876296	6.849590543
Run 2	164	35	0.2134146341	1.461788797	6.37064675
Run 3	167	46	0.2754491018	1.435964127	4.945164574
Run 4	159	27	0.1698113208	1.499362903	7.451132963
Total/Average	811	176	0.2164710362	1.476030847	6.525263904

Link Loss Probabilities: 0.5, 0.6, 0.7, 0.1, 0.2, 0.1

Loss Probability	Sent	Received	PDR	IGG	IPG
Run 0	162	9	0.05555555556	1.475519785	25.20279922
Run 1	162	1	0.006172839506	1.474170608	18.85166917
Run 2	160	7	0.04375	1.010326067	30.65380574
Run 3	160	6	0.0375	1.494131166	20.6747795
Run 4	165	7	0.04242424242	1.453318571	31.8569104
Total/Average	809	30	0.0370805275	1.381493239	25.4479928

For this part setting individual probabilities of loss was implemented. The IGGs stay consistent with an average of around 1.25 s and the PDR as seen is mostly predicated on the link with the most probability of loss. Together the different links add up to even more of a probability of loss.

Part 3: Network Reliability with Multiple Paths (20%)

Checkpoints 3.1

1. Analytical calculation and simulation results for Packet-delivery Ratio with equal probability of losses.

The divide of packets going to C vs D is coded to be set at a 50% split chance. We use this throughout the entire lab.

- The C route gives:

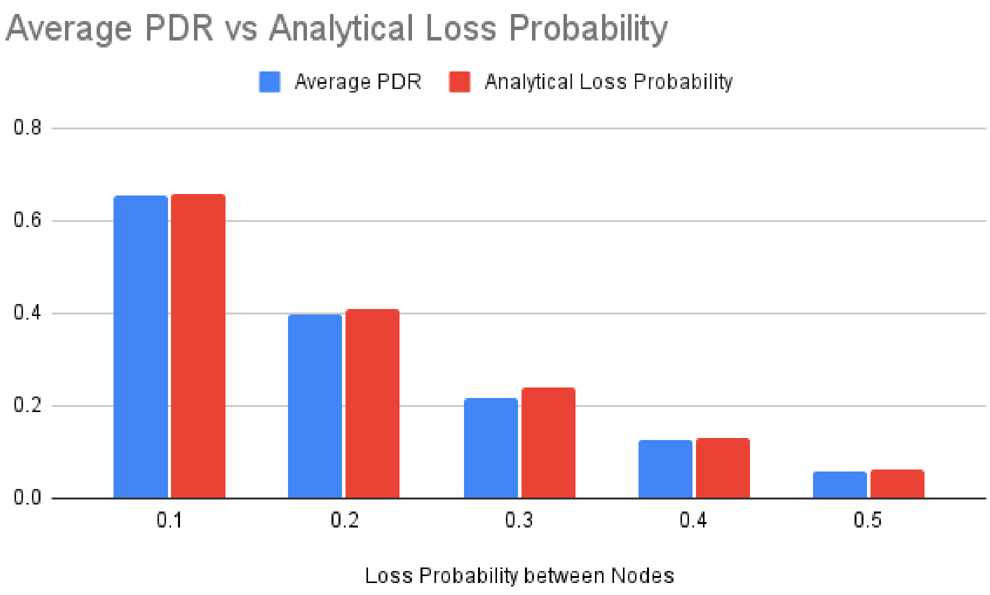
  Pr{C success} = (1-p_{AB}) ∧(1-p_{BC}) ∧(1-p_{CE}) ∧(1-p_{EF})

- The D route gives:

  Pr{D success} = (1-p_{AB}) ∧(1-p_{BD}) ∧(1-p_{DE}) ∧(1-p_{EF})

We only used one of the formulas to calculate the analytical loss probability of the network as the loss probability is the same in the entire network.

Link Loss Probability:	Average PDR:	Analytical Loss Probability (Pr{Success}):
0.1	0.6541568628	0.6561
0.2	0.3963834924	0.4096
0.3	0.2172660728	0.2401
0.4	0.1267700012	0.1296
0.5	0.05867973594	0.0625



We added a node in the chain letting the packets take multiple roads. And implemented a priority system for the packet so that they don't mindlessly get lost.

Our Simulations on the PDRs for the different loss probabilities were a bit shy of the analytical Average. This could be a result of giving the Packets a second option by jumping back and forth. Although it is miniscule as shown in the graph. The simulations was done on a 240s basis and could be improved by either longer runs or more runs to calculate the average on for a more precise simulation.

2. Analytical calculation and simulation results for Packet-delivery Ratio with different probability of losses.

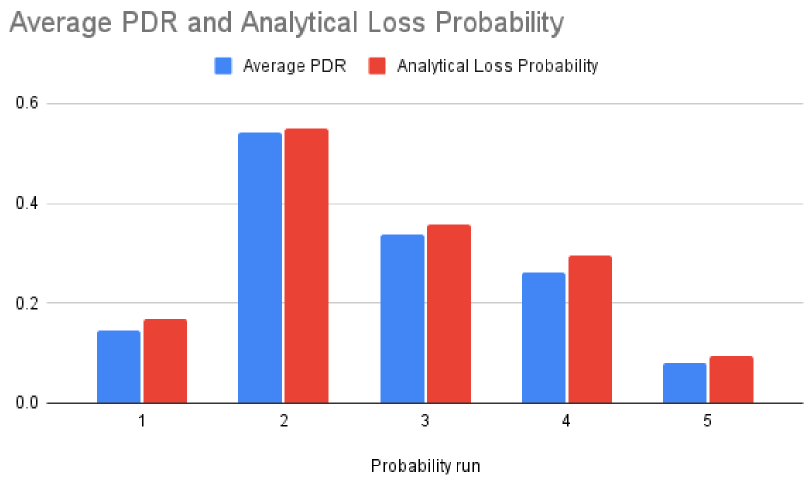
Link:	Run 1	Run 2	Run 3	Run 4	Run 5
p_{AB}	0.5	0.1	0.3	0.25	0.5
p_{BC}	0.4	0.2	0.2	0.25	0.6
p_{BD}	0.4	0.2	0.2	0.25	0.6
p_{CE}	0.35	0.1	0.1	0.4	0.7
p_{DE}	0.25	0.2	0.3	0.1	0.1
p_{EF}	0.2	0.1	0.2	0.3	0.2

We decided for the analytical loss probability here to use the average loss probability from going both routes as follows.

- Average of C and D routes:

  Pr{Average} = (Pr{C success} + Pr{D success}) / 2

Run:	Average PDR:	Analytical Loss Probability (Pr{Average}):
1	0.1441369684	0.168
2	0.5420264826	0.5508
3	0.3368965237	0.3584
4	0.2626448563	0.2953125
5	0.08175080409	0.096



Result wise we got within ±3.5% for the PDRs and the analytical loss probability.

As pointed out in the previous point this this could be because of the lower simulation time of 240s or lack of runs to gather information for a more precise average simulated values.

Checkpoints 3.2

1. Analytical calculation and simulation results for Simple Forwarding with equal probability of loss.

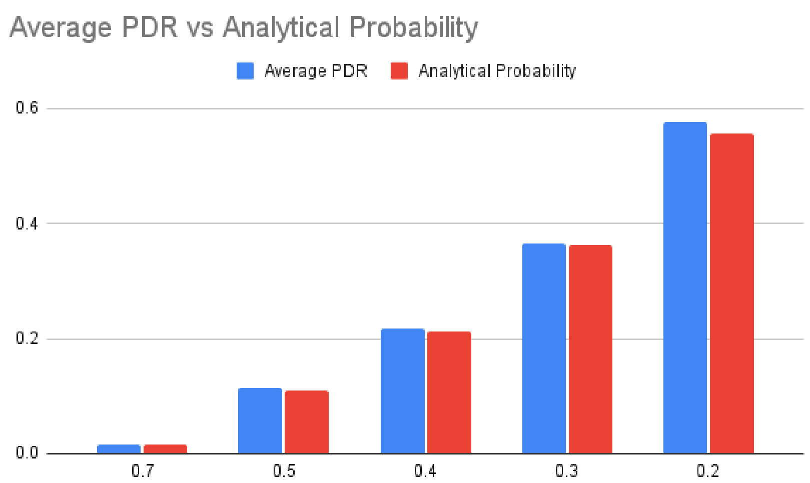
- Analytical probability of success calculation

  Pr{Success} = (1 - p_(AB)) ∧ (1 - p_(BE)) ∧(1 - p_(EF))

- Where (1-p_(BE)) is:

  (1-p_(BE)) = ((1-p_(BC))(1-p_(CE)) + (1-p_(BD))(1-p_(DE))) - ((1-p_(BC))(1-p_(CE))(1-p_(BD))(1-p_(DE)))

Loss probability	Average PDR	Analytical Probability
0.7	0.01495465022	0.0154771
0.5	0.1151294972	0.109375
0.4	0.2169945067	0.212544
0.3	0.3643492657	0.362551
0.2	0.5755201863	0.557056



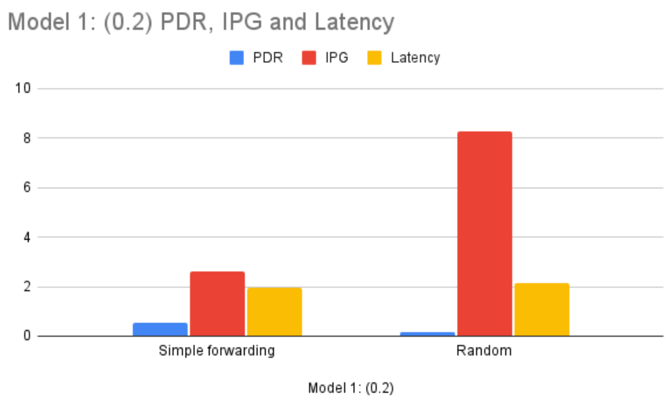
On average we got simulated results that were within ±1% of the analytical probability for our implementation of simple forwarding.

Part 4: Network Reliability for different network models (40%)

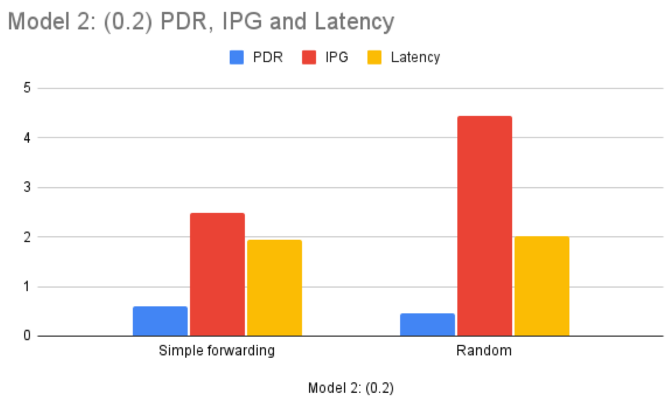
Checkpoints 4

1. Simulation results for random next hop and Simple Forwarding in a single probability of losses. Metrics: PDR, IPG, latency. <u>3 networks.</u>

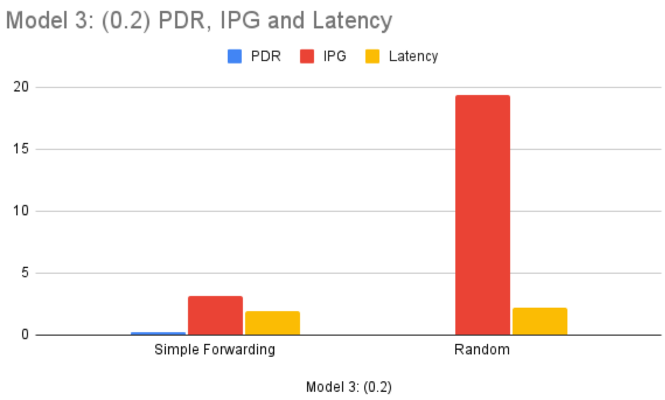
For this simulation we used a loss probability of 0.2 with all nodes. For the PDR it is recommended to look at the table.



Model 1: (0.2)	PDR	IPG	Latency
Simple forwarding	0.5628752635	2.620354	1.935236
Random	0.1818435779	8.25966	2.145588



Model 2:	PDR	IPG	Latency
Simple forwarding	0.6010364442	2.47355	1.932686
Random	0.4485850952	4.451188	2.004503333



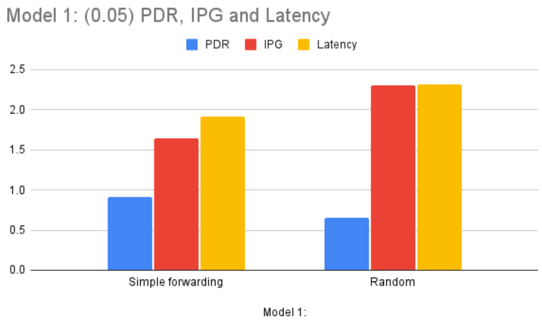
Model 3:	PDR	IPG	Latency
Simple Forwarding	0.2326953567	3.194618	1.934512
Random	0.07594835519	19.34674	2.183512

For the Random-hop we tried to priorities for it to go towards the larger nodes by initializing the different gates in a specific way. Although given more options it will naturally take longer time to complete a single run.

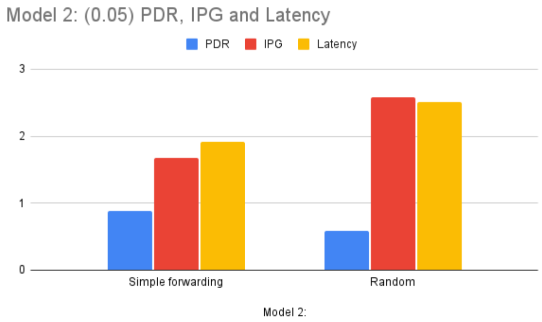
2. Simulation results for random next hop and Simple Forwarding for different levels of probability. Metrics: PDR, IPG, latency. 3 networks.

For this assignment we choose to use probabilities: 0.05, 0.1, 0.15 and 0.30 to have a higher chance of a packet finding it's way to the final node. We also increased the time to 1000s to improve the result data.

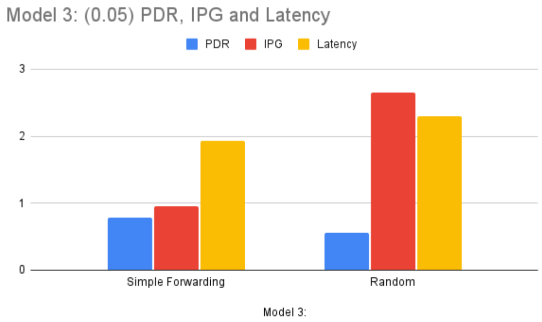
- **0.05 probability:**



Model 1:	PDR	IPG	Latency
Simple forwarding	0.9095435178	1.645482	1.9173838
Random	0.6483478752	2.305884	2.320046

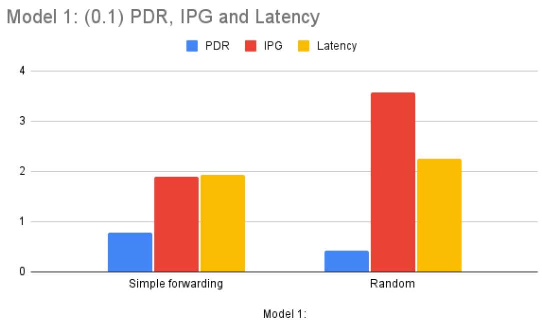


Model 2:	PDR	IPG	Latency
Simple forwarding	0.8883402081	1.674628	1.921274
Random	0.5804754654	2.582532	2.511822

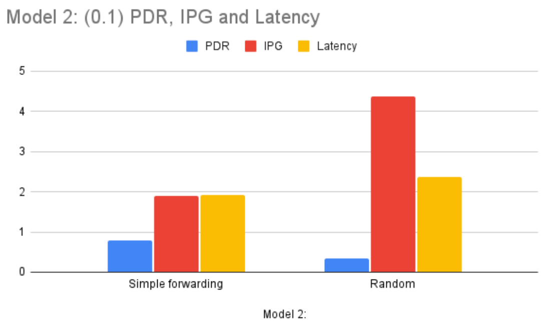


Model 3:	PDR	IPG	Latency
Simple Forwarding	0.7855517541	0.9520234	1.922416
Random	0.564300289	2.652792	2.297238

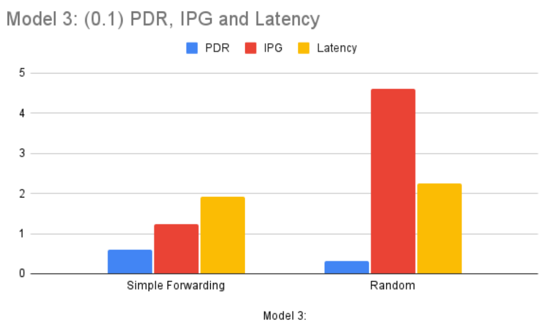
- **0.10 probability:**



Model 1:	PDR	IPG	Latency
Simple forwarding	0.7876905387	1.900198	1.927998
Random	0.4200613188	3.570736	2.261082



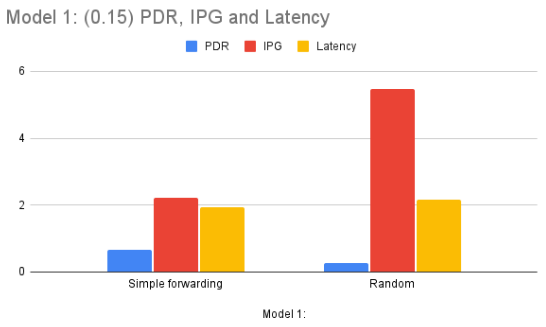
Model 2:	PDR	IPG	Latency
Simple forwarding	0.7908898219	1.884552	1.9135576
Random	0.3453621834	4.360714	2.374076



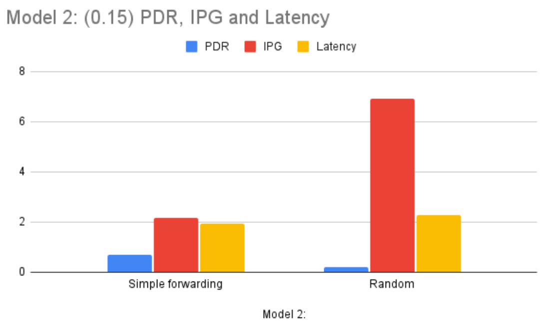
Model 3:	PDR	IPG	Latency
Simple Forwarding	0.5926502849	1.23948	1.9197212

Random	0.324599085	4.612518	2.240006

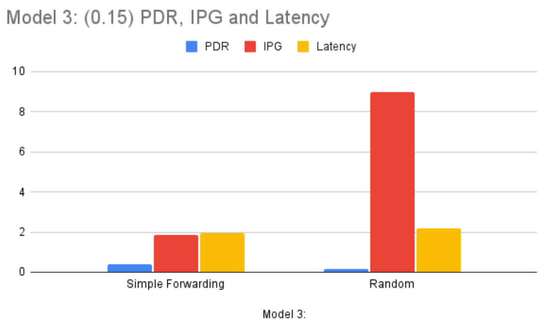
- **0.15:**



Model 1:	PDR	IPG	Latency
Simple forwarding	0.6755601441	2.216598	1.930094
Random	0.2741357927	5.484598	2.175432

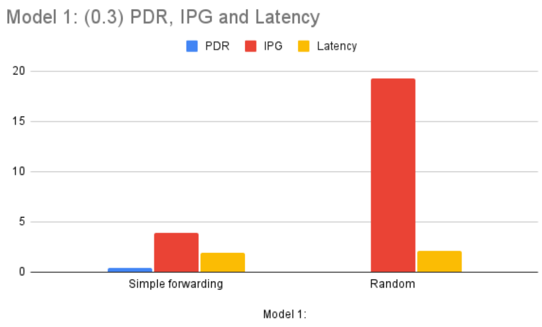


Model 2:	PDR	IPG	Latency
Simple forwarding	0.6926299045	2.14958	1.9269
Random	0.2178826415	6.898604	2.285482

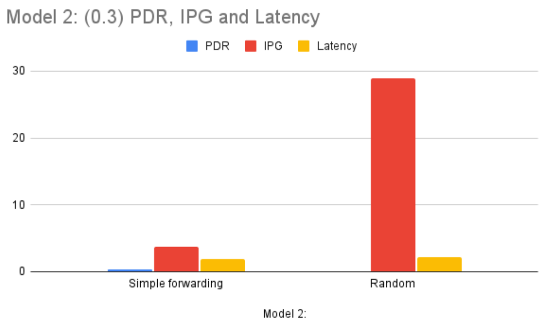


Model 3:	PDR	IPG	Latency
Simple Forwarding	0.4010390339	1.865334	1.958978
Random	0.1673013555	8.964214	2.197596

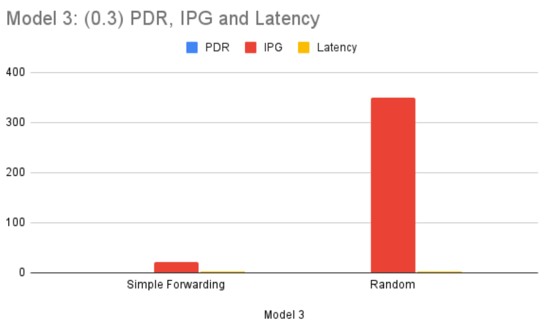
- **0.3:**



Model 1:	PDR	IPG	Latency
Simple forwarding	0.3840540527	3.903732	1.957224
Random	0.07820955714	19.26434	2.078133



Model 2:	PDR	IPG	Latency
Simple forwarding	0.4012456369	3.741472	1.9618
Random	0.05116459645	28.8795	2.200862



Model 3:	PDR	IPG	Latency
Simple Forwarding	0.03646560879	20.87636	1.992348
Random	0.00720503757	350.55668	2.287584

For this exercise we used the implementation of Simple forwarding we used in Part 3 and the Random Hop form Part 2-3. For network model 3 we took extra care when setting it up in the .ned file to assure that the gates got implemented in a growing manner.

Given the specific way that model 3 handles loss probabilities, multiplying it by 3 at the last node we couldn't go for any value above 0.33 as this would result in no packets being received.

Conclusion wise you can derive from the graphs that inter packet gaps becomes the leading metric as probability of loss grows, especially when it comes to the random next hop implementation.

Group members and work performed.

Haron Obaid

- Coding Part 1 – 4
- Debugging of Memory Leak
- Data Gathering Part 2 - 4
- Overview of report

Fabian Henrysson

- Coding Part 1-3.1
- Data gathering part 1
- Data handling in google sheets Part 1-4
- Graphs
- Lab report writing