

TSIN01 Information Networks

Slotted ALOHA Algorithm and Pseudo-Bayesian Stabilization

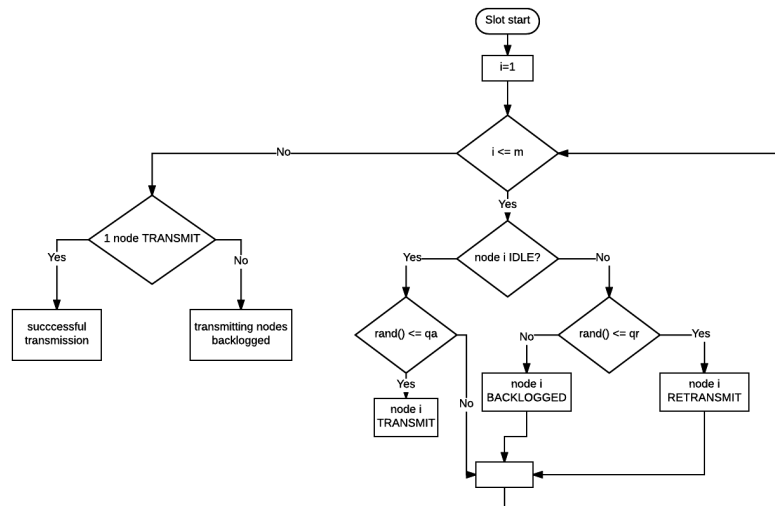
Rasmus Hedin
rashe877@student.liu.se
940524-3396

October 31, 2017

1 Assumptions

Consider a slotted multiaccess system with $m = 100$ nodes that have no-buffering. The packet arrivals are Poisson distributed with overall arrival rate $\lambda = 1/e$ packets per slot. The system runs for a duration of $t = 1000$ slots. Initially, all nodes are unbacklogged. The slotted ALOHA protocol, as well as a pseudo-bayesian stabilization version of the protocol have been implemented in Matlab.

2 Slotted ALOHA



2.1 $\lambda = 1/e, q_r = 0.01$

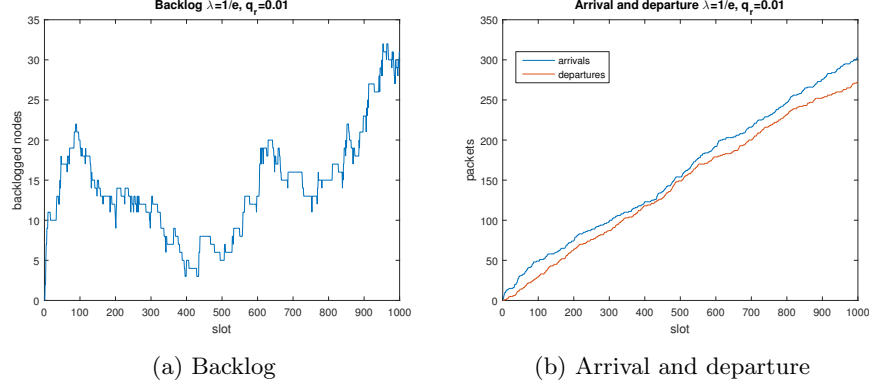
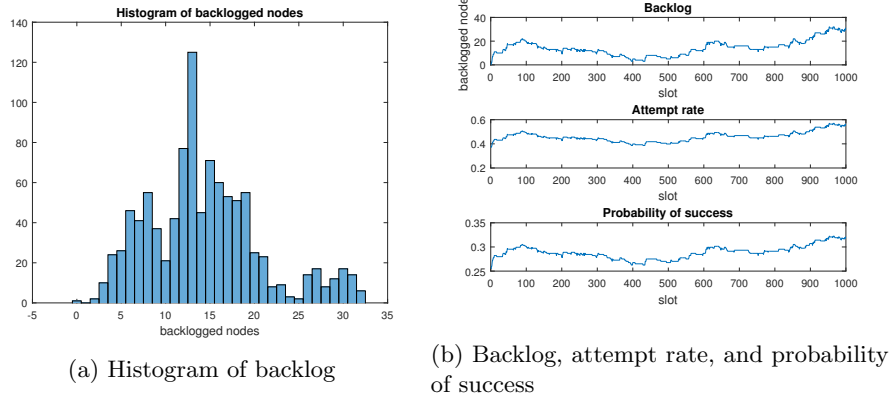


Figure 1

3) In the first simulation the ALOHA protocol have been executed with retransmission rate $q_r = 0.01$ and arrival rate $\lambda = 1/e$. In figure 1a the number of packets in the backlog have been plotted. We can see that the backlog starts at zero and then increase. The backlog increases with one or more at a collision but it can only decrease by one step at a time since there can only be one succesful transmission at a time. This can be seen in the left graph. Figure 1b shows arrived packets (blue legend) and departed packets (red legend). The graph for arrived packets increase one step for each new arriving packet and the graph for departures increase one step at a time when a packet is successfully transmitted. The gap between these two graphs represents the delay of the system where a larger gap equals a longer delay. The horizontal distance between the two graphs at a $y = n$ is the delay for arriving packet number n .



4) By using the Matlab command `tabulate` we can obtain the steady-state

probabilities of the Markov chain from the backlog. The expected number of backlogged nodes N can then be calculated as $N = \sum_{n=0}^m np_n$, where n is the number of backlogged nodes and p_n is the probability of having n backlogged nodes. From the simulation results the expected number of backlogged nodes was around 11. The attempt rate $G(n) = (m - n)q_a + nq_r$, and probability of success $P_s = G(n)e^{-G(n)}$ can be seen in figure 2b. The rates depends on the backlog and have been calculated with the theoretical formulas for each slot. The average of the calculated probabilities of success is 0.28. Compared to the probability of success from the simulation which was 0.29, we may say that the estimation is good.

2.2 $\lambda = 1/2, q_r = 0.01$

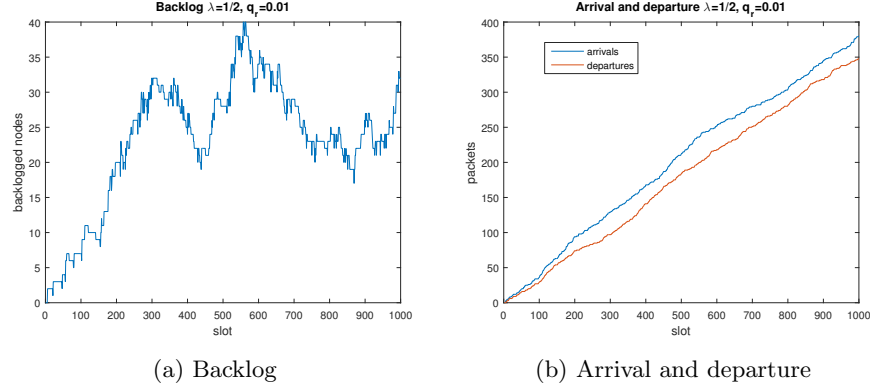


Figure 3

In this run of the simulation $\lambda = 1/2$ instead and q_r is kept at 0.01. this will increase the arrival rate and as we can see in figure 3a the backlog increases faster and the average number of backlogged nodes is also higher than in the previous plot. The delay have also increased since the gap between the graphs in figure 3b have increased. This is expected since now packets arrive at faster rate giving more collisions.

2.3 $\lambda = 1/e, q_r = 0.1$

In this run of the simulation $\lambda = 1/e$, and $q_r = 0.1$. This will make the backlog clear faster as we can see in figure 4a. Figure 4b shows that the delay is low and that packets depart very soon after arriving. However, sometimes we get unlucky where the backlog goes above a threshold and it will never clear and

give no succesful transmissions as seen in figure 5a. This is also very clear in figure 5b where we can see that there is almost no successful transmission when the backlog becomes large.

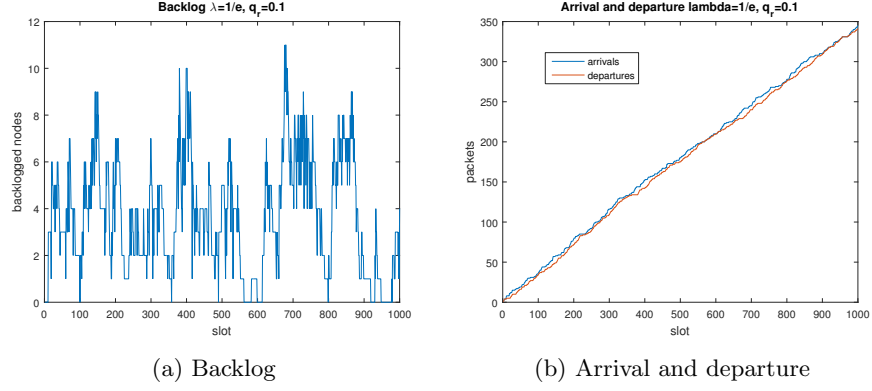


Figure 4

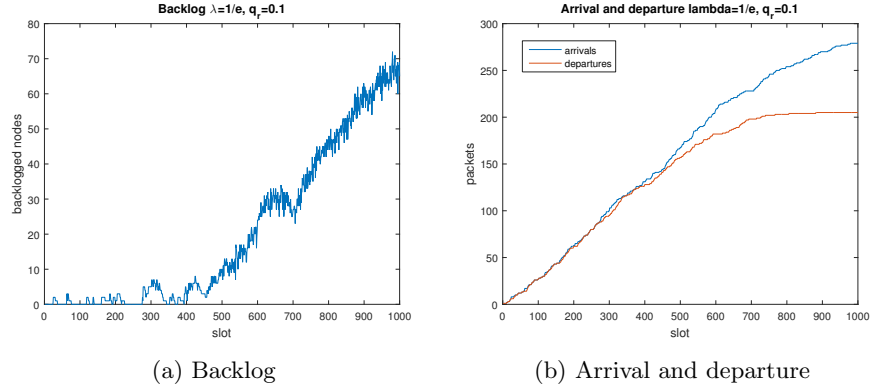
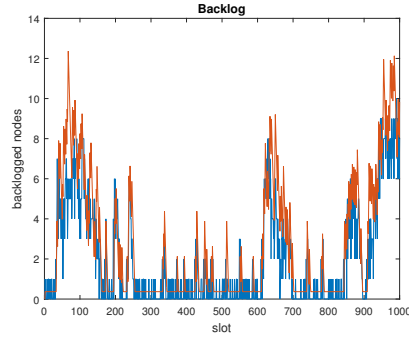
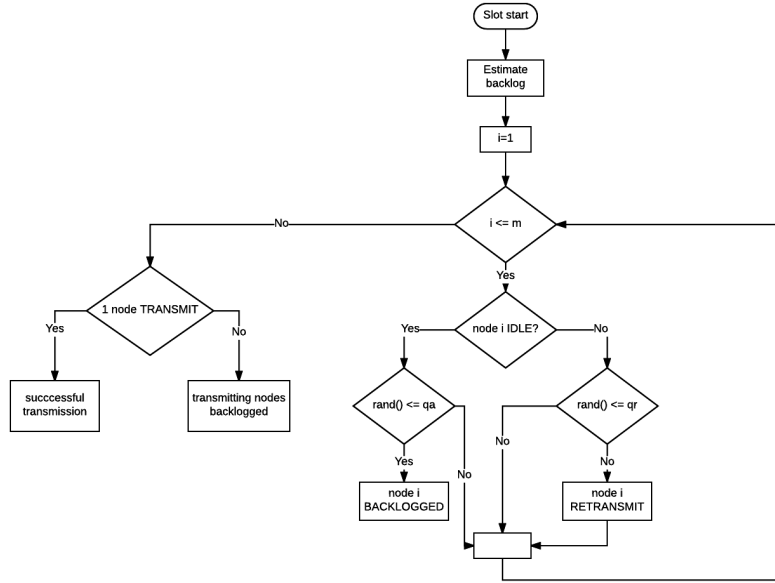


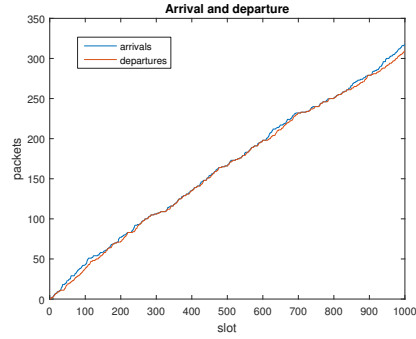
Figure 5

3 Slotted ALOHA with Pseudo-Bayesian Stabilization

In this section the simulations will be executed with the slotted ALOHA protocol, but now q_r is adapted by the pseudo-bayesian stabilization. $\lambda = 1/e$ is known.



(a) Backlog



(b) Arrival and departure

Figure 6

With pseudo-bayesian stabilization the backlog will be estimated. Figure 6a shows the simulated backlog (blue) and the estimated backlog (red). As we

can see the estimate follows the real backlog very good which will give a good retransmission rate. The delay is low as we can see in figure 6b since the gap between the graphs is very small. This is an indication that the system is stable.

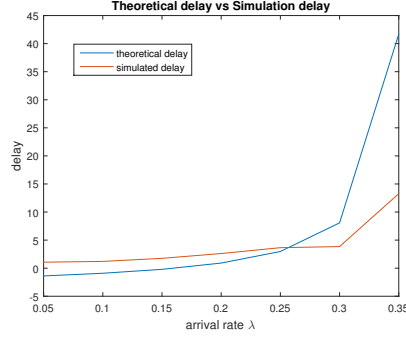


Figure 7: Approximate delay

The approximate delay of the system depends on the arrival rate λ . In figure 7 the approximated delay for successful packet transmission has been plotted. Both the delay from the formula (blue) and from the simulation results (red) can be seen for arrival rate $\lambda = 0.05 : 0.05 : 0.35$. The actual delay from the simulation result is almost the same as the theoretical delay for small arrival rates. But for larger arrival rates the actual simulation delay is smaller than the theoretical delay as we can see in figure 7.

A Slotted ALOHA

```
1 function [backlog, arrival, departure] = slotted_aloha(m, T, ...
    lambda, qr)
2
3 % State of a nodes
4 % 0: Idle
5 % 1: Transmitting
6 % 2: Backlogged
7 state = zeros(1,m);
8
9 transmission_slot = zeros(1,m);
10 backlog = zeros(1,T);
11 arrival = zeros(1,T);
12 departure = zeros(1,T);
13 arrival_slot = zeros(1,T);
14 departure_slot = zeros(1,T);
15 arrivedPkts = 0;
16 transnPkts = 0;
17
18 % Calculate the probability of arrival
19 qa = 1 - exp(-lambda/m);
20
21 % Simulate for T slots
22 t = 0;
23 while t < T
24     t = t + 1;
25
26     for i = 1:m
27         % Is node_i IDLE?
28         if state(i) == 0
29             % Packet arrive at node_i with probability qa
30             if rand() <= qa
31                 state(i) = 1;
32                 transmission_slot(i) = t;
33                 arrivedPkts = arrivedPkts + 1;
34                 arrival_slot(arrivedPkts) = t;
35             end
36         else
37             % Node i is backlogged
38             % Retransmit with probability qr
39             if rand() <= qr
40                 state(i) = 1;
41             else
42                 % Node_i did not retransmit
43                 % and stays backlogged
44                 state(i) = 2;
45                 backlog(t) = backlog(t) + 1;
46             end
47         end
48     end
49
50     % How many nodes are transmitting?
51     transmissions = 0;
52     transmitting_node = -1;
```

```

53     for i = 1:m
54         if state(i) == 1
55             transmissions = transmissions + 1;
56             transmitting_node = i;
57         end
58     end
59
60     % One node transmitting = SUCCESS
61     if transmissions == 1
62         state(transmitting_node) = 0;
63         transnPckts = transnPckts + 1;
64         departure_slot(transnPckts) = t;
65     end
66
67     % Store packet arrivals and departures
68     % at time slot t
69     arrival(t) = arrivedPckts;
70     departure(t) = transnPckts;
71 end
72
73 % Calculate mean delay
74 W = departure_slot(1:transnPckts) - arrival_slot(1:transnPckts);
75 W = mean(W);
76
77 fprintf('\n=====\\n')
78 fprintf('Simulation finished\\n')
79 fprintf('Nodes: %u\\n',m);
80 fprintf('Slots %u\\n',T);
81 fprintf('lambda: %.3f\\n',lambda);
82 fprintf('qr: %.3f\\n',qr);
83 fprintf('qa: %.3f\\n',qa);
84 fprintf('Arrived packets: %u\\n',arrivedPckts);
85 fprintf('Transmitted packets: %u\\n',transnPckts);
86 fprintf('Mean delay: %.3f\\n',W);
87 fprintf('=====\\n')

```


B Slotted ALOHA with Pseudo-Bayesian Stabilization

```
1 function [backlog, backlog_estimate, arrival, departure, W] = ...
    stabilized_slotted_aloha(m, T, lambda)
2
3 % State of a nodes
4 % 0: Idle
5 % 1: Transmitting
6 % 2: Backlogged
7 state = zeros(1,m);
8
9 transmission_slot = zeros(1,m);
10 backlog = zeros(1,T);
11 backlog_estimate = zeros(1,T+1);
12 arrival = zeros(1,T);
13 departure = zeros(1,T);
14 arrival_slot = zeros(1,T);
15 departure_slot = zeros(1,T);
16 arrivedPkts = 0;
17 transnPckts = 0;
18
19 % Calculate the probability of arrival
20 qa = 1 - exp(-lambda/m);
21
22 % Simulate for T slots
23 t = 0;
24 while t < T
25     t = t + 1;
26
27     % Update retransmission rate (qr)
28     if 0 ≤ backlog_estimate(t) && backlog_estimate(t) < 1
29         qr = 1;
30     else
31         qr = 1/backlog_estimate(t);
32     end
33
34     for i = 1:m
35         % Is node_i IDLE?
36         if state(i) == 0
37             % Packet arrive at node_i with probability qa
38             if rand() ≤ qa
39                 state(i) = 2;
40                 transmission_slot(i) = t;
41                 arrivedPkts = arrivedPkts + 1;
42                 arrival_slot(arrivedPkts) = t;
43                 backlog(t) = backlog(t) + 1;
44             end
45         else
46             % Node i is backlogged
47             % Retransmit with probability qr
48             if rand() ≤ qr
49                 state(i) = 1;
50             else
```

```

51         % Node_i did not retransmit
52         % and stays backlogged
53         state(i) = 2;
54         backlog(t) = backlog(t) + 1;
55     end
56 end
57 end
58
59 % How many nodes are transmitting?
60 transmissions = 0;
61 transmitting_node = -1;
62 for i = 1:m
63     if state(i) == 1
64         transmissions = transmissions + 1;
65         transmitting_node = i;
66     end
67 end
68
69 % One node transmitting = SUCCESS
70 if transmissions == 1
71     state(transmitting_node) = 0;
72     transmPckts = transmPckts + 1;
73     departure_slot(transmPckts) = t;
74 end
75
76 % Estimate backlog for slot t+1
77 if transmissions ≤ 1
78     backlog_estimate(t+1) = max(lambda, backlog_estimate(t) ...
79         + lambda - 1);
80 else
81     backlog_estimate(t+1) = backlog_estimate(t) + lambda + ...
82         1/(exp(1)-2);
83 end
84
85 % Store packet arrivals and departures
86 % at time slot t
87 arrival(t) = arrivedPckts;
88 departure(t) = transmPckts;
89 end
90
91 % Calculate mean delay
92 W = departure_slot(1:transmPckts) - arrival_slot(1:transmPckts);
93 W = mean(W);
94
95 fprintf('\n=====\\n')
96 fprintf('Simulation finished\\n')
97 fprintf('Nodes: %u\\n',m);
98 fprintf('Slots %u\\n',T);
99 fprintf('lambda: %.3f\\n',lambda);
100 fprintf('qa: %.3f\\n',qa);
101 fprintf('Arrived packets: %u\\n',arrivedPckts);
102 fprintf('Transmitted packets: %u\\n',transmPckts);
103 fprintf('Mean delay: %.3f\\n',W);
104 fprintf('=====\\n')

```

C Plots

```
1  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
2  %% Slotted ALOHA  %%
3  %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
4  m = 100;
5  T = 1000;
6  lambda = exp(-1);
7  qr = 0.01;
8  [backlog, arrival, departure] = slotted_aloha(m,T,lambda,qr);
9
10 %% Backlog
11 figure
12 plot(backlog);
13 title('Backlog {\lambda=1/e, q_r=0.01}');
14 xlabel('slot');
15 ylabel('backlogged nodes');
16 x = 1:T;
17
18 %% Arrival and departure of packets
19 figure
20 plot(x,arrival,x,departure);
21 title('Arrival and departure {\lambda=1/e, q_r=0.01}');
22 xlabel('slot');
23 ylabel('packets');
24 legend('arrivals','departures');
25
26 %% Histogram of backlog
27 figure
28 histogram(backlog);
29 title('Histogram of backlogged nodes');
30 xlabel('backlogged nodes');
31
32 %% Steady-states of the Markov chain
33 tbl = tabulate(backlog);
34 N = sum(tbl(1:end,1).*(tbl(1:end,3)/100));
35 D = N/lambda;
36
37 %% Attempt rate
38 qa = 1 - exp(-lambda/m);
39 G = attempt_rate(m, backlog, qa, qr);
40 figure
41 subplot(3,1,1);
42 plot(x,backlog);
43 title('Backlog');
44 xlabel('slot');
45 ylabel('backlogged nodes');
46 subplot(3,1,2);
47 plot(x,G);
48 title('Attempt rate');
49 xlabel('slot');
50
51 % Probability of success
52 G = attempt_rate(m, backlog, qa, qr);
53 Ps = G.*exp(-G);
```

```

54 Ps_avg = mean(Ps);
55 subplot(3,1,3);
56 plot(x,Ps);
57 title('Probability of success');
58 xlabel('slot');
59
60 %%
61 Pnew = (m-backlog)*qa;
62 figure
63 plot(x,Ps,x,Pnew);
64 title('Probability of new arrivals');
65
66 %%
67 G_theory = attempt_rate(m,0:m,qa,qr);
68 Ps_theory = G_theory.*exp(-G_theory);
69 Pnew_theory = (m-(0:m))*qa;
70 figure
71 plot(0:m,Ps_theory,0:m,Pnew_theory);
72 Ps_avg_theory = mean(Ps_theory);
73
74 %%
75 Dn_theory = Pnew_theory - Ps_theory;
76 figure
77 subplot(2,1,1);
78 plot(0:m,Ps_theory,0:m,Pnew_theory);
79 subplot(2,1,2);
80 plot(0:m,Dn_theory,0:m,zeros(1,m+1));
81
82 %% Assignment 5
83 m = 100;
84 T = 1000;
85 lambda = 1/2;
86 qr = 0.01;
87 [backlog, arrival, departure] = slotted_aloha(m,T,lambda,qr);
88
89 figure
90 plot(backlog);
91 title('Backlog {\lambda=1/2, q_r=0.01}');
92 xlabel('slot');
93 ylabel('backlogged nodes');
94 x = 1:T;
95 figure
96 plot(x,arrival,x,departure);
97 title('Arrival and departure {\lambda=1/2, q_r=0.01}');
98 xlabel('slot');
99 ylabel('packets');
100 legend('arrivals','departures');
101
102 %% Assignment 6
103 m = 100;
104 T = 1000;
105 lambda = exp(-1);
106 qr = 0.1;
107 [backlog, arrival, departure] = slotted_aloha(m,T,lambda,qr);
108
109 figure
110 plot(backlog);

```

```

111 title('Backlog {\lambda=1/e, q_r=0.1}');
112 xlabel('slot');
113 ylabel('backlogged nodes');
114 figure
115 plot(x,arrival,x,departure);
116 title('Arrival and departure {\lambda=1/e, q_r=0.1}');
117 xlabel('slot');
118 ylabel('packets');
119 legend('arrivals','departures');
120
121
122 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
123 %% Pseudo-Bayesian Stabilization %%
124 %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
125 m = 100;
126 T = 1000;
127 x = 1:T;
128 lambda = exp(-1);
129 [backlog, backlog_estimate, arrival, departure, W] = ...
    stabilized.slotted.aloha(m,T,lambda);
130
131 %% Backlog and backlog estimate
132 figure
133 plot(x,backlog,x,backlog_estimate(1:1000));
134 title('Backlog');
135 xlabel('slot');
136 ylabel('backlogged nodes');
137
138 %% Arrival and departure
139 figure
140 plot(x,arrival,x,departure);
141 title('Arrival and departure');
142 xlabel('slot');
143 ylabel('packets');
144 legend('arrivals','departures');
145
146 %% Histogram of backlog
147 figure
148 histogram(backlog);
149 title('Histogram of backlogged nodes');
150
151 %% Steady-states of the Markov chain
152 tbl = tabulate(backlog);
153 N = sum(tbl(1:end,1).*(tbl(1:end,3)/100));
154 D = N/lambda;
155
156 %% Backlog, Attempt rate, and Probabiliy of success
157 qa = 1 - exp(-lambda/m);
158 G = attempt_rate(m, backlog, qa, qr);
159 figure
160 subplot(3,1,1);
161 plot(x,backlog);
162 title('Backlog');
163 xlabel('slot');
164 ylabel('backlogged nodes');
165 subplot(3,1,2);
166 plot(x,G);

```

```

167 title('Attempt rate');
168
169 % Probability of success
170 G = attempt_rate(m, backlog, qa, qr);
171 Ps = G.*exp(-G);
172 Ps_avg = mean(Ps);
173 subplot(3,1,3);
174 plot(x,Ps);
175 title('Probability of success');
176
177 %% Probability of new arrivals
178 Pnew = (m-backlog)*qa;
179 figure
180 plot(x,Ps,x,Pnew);
181 title('Probability of new arrivals');
182
183 %% Theoretical calculations
184 G_theory = attempt_rate(m,0:m,qa,qr);
185 Ps_theory = G_theory.*exp(-G_theory);
186 Pnew_theory = (m-(0:m))*qa;
187 figure
188 plot(0:m,Ps_theory,0:m,Pnew_theory);
189 Ps_avg_theory = mean(Ps_theory);
190
191 %% Approximate delay analysis
192 lambda = 0.05:0.05:0.35;
193 W_theory = average_delay(lambda);
194 figure
195 plot(lambda,W_theory);
196 title('');
197 xlabel('arrival rate {\lambda}')
198 ylabel('delay')
199
200 %% Average delay from simulation
201 lambda = 0.05:0.05:0.35;
202 W = zeros(1,length(lambda));
203 for i = 1:length(lambda)
204     [backlog, backlog_estimate, arrival, departure, Wi] = ...
        stabilized_slotted_aloha(m,T,lambda(i));
205     W(i) = Wi;
206 end
207 plot(lambda,W_theory, lambda,W);
208 title('Theoretical delay vs Simulation delay');
209 xlabel('arrival rate {\lambda}')
210 ylabel('delay')
211 legend('theoretical delay', 'simulated delay');

```

D Utility functions

```

1 function G = attempt_rate(m, n, qa, qr)
2     G = (m-n)*qa + n*qr;

```

```
1 function W = average_delay(lambda)
2     e = exp(1);
3     W = (e - (1/2)) ./ (1-lambda.*e) - ...
        (e-1)*(exp(lambda)-1)/(lambda.*(1-(e - 1)*(exp(lambda)-1)));
```