

# TSIN01 Information Networks

## Slotted ALOHA Algorithm and Pseudo-Bayesian Stabilization

Rasmus Hedin  
*<rashe877@student.liu.se>*  
940524-3396

October 6, 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 $\lambda = 1/e, q_r = 0.01$

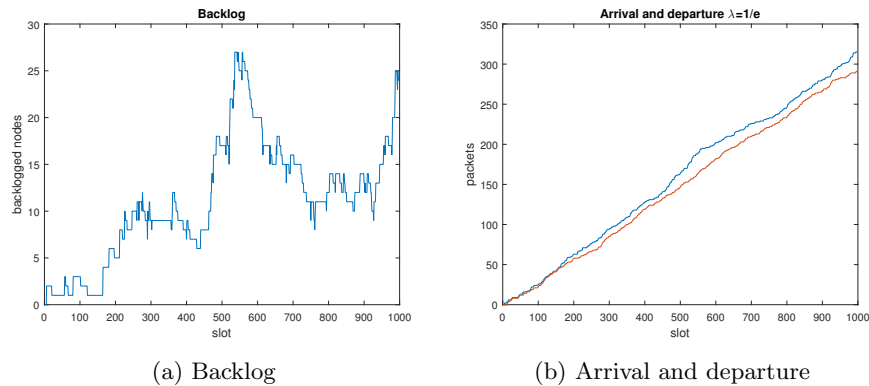
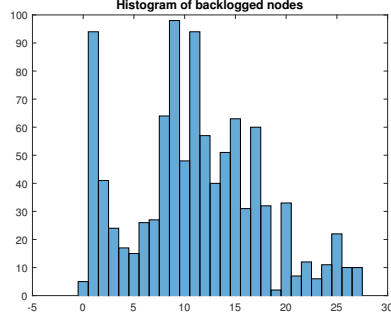


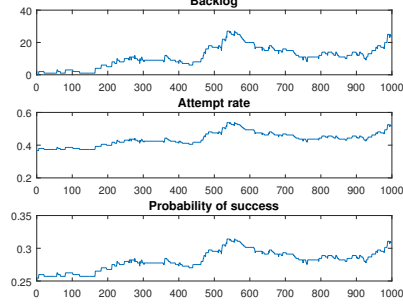
Figure 1

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 backlogged have been plotted. We can see that there is always some backlogged nodes. Figure 1b shows arrived packets and departed packets. The graph for arrived packets is always over the graph for departed packets which means that we have some delay in the system. A larger gap between the graphs would mean a longer delay.



(a) Histogram of backlog



(b) Backlog, attempt rate, and probability of success of success

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

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 has also increased since the gap between the graphs in figure 3b has increased. This is expected since now packets arrive at a faster rate giving more collisions.

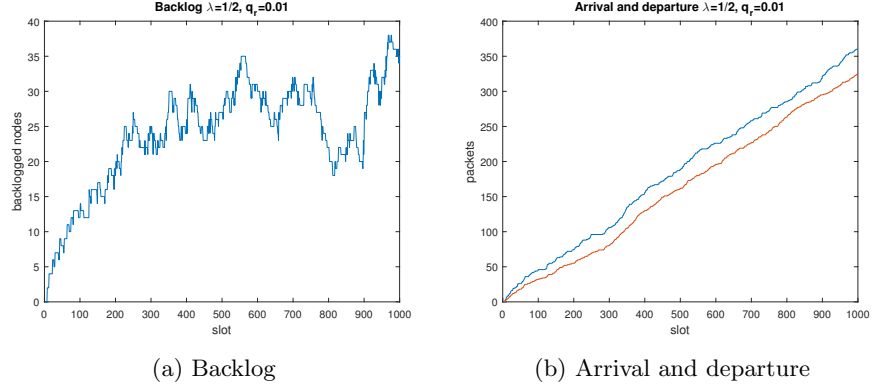


Figure 3

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

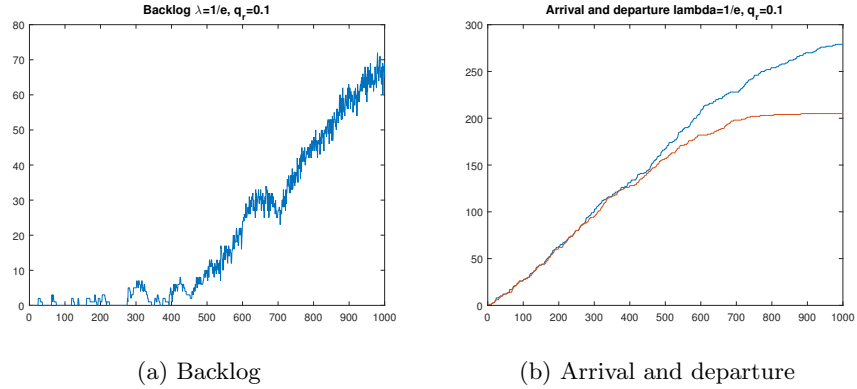


Figure 4

In this run of the simulation  $\lambda = 1/e$ , and  $q_r = 0.1$ . This will make the backlog clear faster, but as we can see in figure 4a, when the backlog goes above a threshold it will only increase and the packets can't be retransmitted. This is also very clear in figure 4b where we can see that there is almost no successful transmission when the backlog becomes large.

## 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.

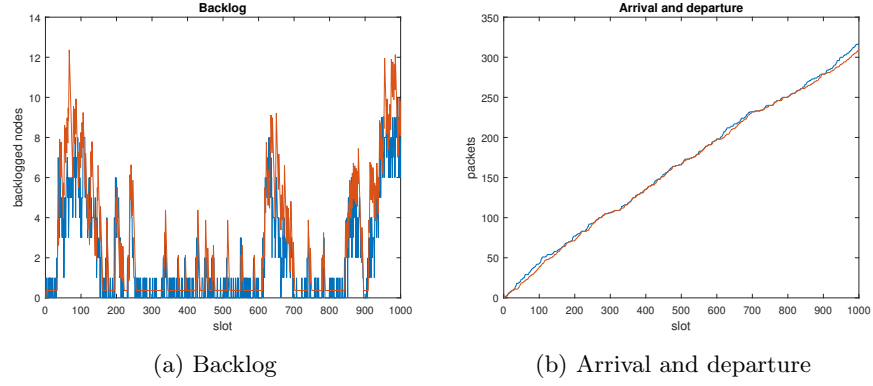


Figure 5

With pseudo-bayesian stabilization the backlog will be estimated. Figure 5a 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 5b since the gap between the graphs is very small. This is an indication that the system is stable.

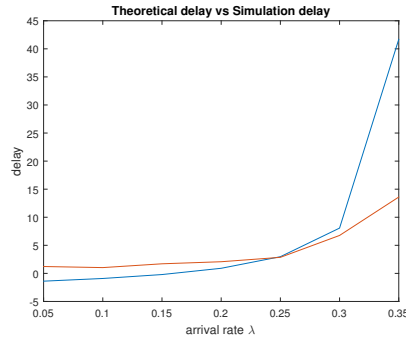


Figure 6: Approximate delay

The approximate delay of the system depends on the arrival rate  $\lambda$ . In figure 6 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 delay is smaller.

## A Slotted ALOHA

```
1 function [backlog, arrival, departure] = slotted_aloha(m, T, ...
    lambda, qr)
2
3 % State of the nodes
4 % 0: Idle
5 % 1: Transmitting
6 % 2: Backlogged
7 state = zeros(1,m);
8 transmission_slot = zeros(1,m);
9 backlog = zeros(1,T);
10 arrival = zeros(1,T);
11 departure = zeros(1,T);
12 meanDelay = 0;
13 arrivedPkts = 0;
14 transnPckts = 0;
15 qa = 1 - exp(-lambda/m);
16
17 t = 0;
18 while t < T
19     t = t + 1;
20
21     for i = 1:m
22         if state(i) == 0
23             % Arrival at node i?
24             if rand() <= qa
25                 state(i) = 1;
26                 transmission_slot(i) = t;
27                 arrivedPkts = arrivedPkts + 1;
28             end
29         else
30             % Retransmission at node i?
31             if rand() <= qr
32                 state(i) = 1;
33             else
34                 state(i) = 2;
35                 backlog(t) = backlog(t) + 1;
36             end
37         end
38     end
39
40     % How many nodes are transmitting?
41     transmissions = 0;
42     transmitting_node = -1;
43     for i = 1:m
44         if state(i) == 1
45             transmissions = transmissions + 1;
46             transmitting_node = i;
47         end
48     end
49
50     % One node transmitting = SUCCESS
51     if transmissions == 1
52         state(transmitting_node) = 0;
```

```

53         transnPckts = transnPckts + 1;
54         delay = t - transmission_slot(transmitting_node);
55         meanDelay = meanDelay + (1/transnPckts)*(delay - meanDelay);
56     end
57
58     arrival(t) = arrivedPckts;
59     departure(t) = transnPckts;
60 end
61
62 fprintf('\nqa: %.3f,\nTransmitted packets: %u,\nMean delay: ...
        %.0f\n',qa,transnPckts,meanDelay);

```

## 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 the nodes
4  % 0: Idle
5  % 1: Transmitting
6  % 2: Backlogged
7  state = zeros(1,m);
8  transmission_slot = zeros(1,m);
9  backlog = zeros(1,T);
10 backlog_estimate = zeros(1,T+1);
11 arrival = zeros(1,T);
12 departure = zeros(1,T);
13 arrival_slot = zeros(1,T);
14 departure_slot = zeros(1,T);
15 meanDelay = 0;
16 arrivedPckts = 0;
17 transnPckts = 0;
18 qa = 1 - exp(-lambda/m);
19
20
21 t = 0;
22 while t < T
23     t = t + 1;
24
25     % Update retransmission rate (qr)
26     if 0 ≤ backlog_estimate(t) && backlog_estimate(t) < 1
27         qr = 1;
28     else
29         qr = 1/backlog_estimate(t);
30     end
31
32     for i = 1:m
33         if state(i) == 0
34             % Arrival at node i?
35             if rand() ≤ qa
36                 state(i) = 2;

```

```

37         transmission_slot(i) = t;
38         arrivedPkts = arrivedPkts + 1;
39         arrival_slot(arrivedPkts) = t;
40         backlog(t) = backlog(t) + 1;
41     end
42 else
43     % Retransmission at node i?
44     if rand() ≤ qr
45         state(i) = 1;
46     else
47         state(i) = 2;
48         backlog(t) = backlog(t) + 1;
49     end
50 end
51 end
52
53 % How many nodes are transmitting?
54 transmissions = 0;
55 transmitting_node = -1;
56 for i = 1:m
57     if state(i) == 1
58         transmissions = transmissions + 1;
59         transmitting_node = i;
60     end
61 end
62
63 % One node transmitting = SUCCESS
64 if transmissions == 1
65     state(transmitting_node) = 0;
66     transnPckts = transnPckts + 1;
67     departure_slot(transnPckts) = t;
68     delay = t - arrival_slot(transnPckts);
69     meanDelay = meanDelay + (1/transnPckts)*(delay - meanDelay);
70 end
71
72 % Estimate backlog for slot t+1
73 if transmissions ≤ 1
74     backlog_estimate(t+1) = max(lambda, backlog_estimate(t) ...
75         + lambda - 1);
76 else
77     backlog_estimate(t+1) = backlog_estimate(t) + lambda + ...
78         1/(exp(1)-2);
79 end
80
81 arrival(t) = arrivedPkts;
82 departure(t) = transnPckts;
83 end
84
85 W = departure_slot(1:transnPckts) - arrival_slot(1:transnPckts);
86 W = mean(W);
87
88 fprintf('\nqa: %.3f,\nTransmitted packets: %u,\nMean delay: ...
89     %.0f\n',qa,transnPckts,W);

```

## 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
25 %% Histogram of backlog
26 figure
27 histogram(backlog);
28 title('Histogram of backlogged nodes');
29
30 %% Steady-states of the Markov chain
31 tbl = tabulate(backlog);
32 N = sum(tbl(1:end,1).*(tbl(1:end,3)/100));
33 D = N/lambda;
34
35 %% Attempt rate
36 qa = 1 - exp(-lambda/m);
37 G = attempt_rate(m, backlog, qa, qr);
38 figure
39 subplot(3,1,1);
40 plot(x,backlog);
41 title('Backlog');
42 subplot(3,1,2);
43 plot(x,G);
44 title('Attempt rate');
45
46 % Probability of success
47 G = attempt_rate(m, backlog, qa, qr);
48 Ps = G.*exp(-G);
49 Ps_avg = mean(Ps);
50 subplot(3,1,3);
51 plot(x,Ps);
52 title('Probability of success');
53
```



```

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

```

```

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

```

```

167
168 %% Theoretical calculations
169 G_theory = attempt_rate(m,0:m,qa,qr);
170 Ps_theory = G_theory.*exp(-G_theory);
171 Pnew_theory = (m-(0:m))*qa;
172 figure
173 plot(0:m,Ps_theory,0:m,Pnew_theory);
174 Ps_avg_theory = mean(Ps_theory);
175
176 %% Approximate delay analysis
177 lambda = 0.05:0.05:0.35;
178 W_theory = average_delay(lambda);
179 figure
180 plot(lambda,W_theory);
181 title('');
182 xlabel('arrival rate {\lambda}')
183 ylabel('delay')
184
185 %% Average delay from simulation
186 lambda = 0.05:0.05:0.35;
187 W = zeros(1,length(lambda));
188 for i = 1:length(lambda)
189     [backlog, backlog_estimate, arrival, departure, Wi] = ...
        stabilized_slotted_aloha(m,T,lambda(i));
190     W(i) = Wi;
191 end
192 plot(lambda,W_theory, lambda,W);
193 title('Theoretical delay vs Simulation delay');
194 xlabel('arrival rate {\lambda}')
195 ylabel('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)));

```