TSIN01 Information Networks
Slotted ALOHA Algorithm and Pseudo-Bayesian Stabilization

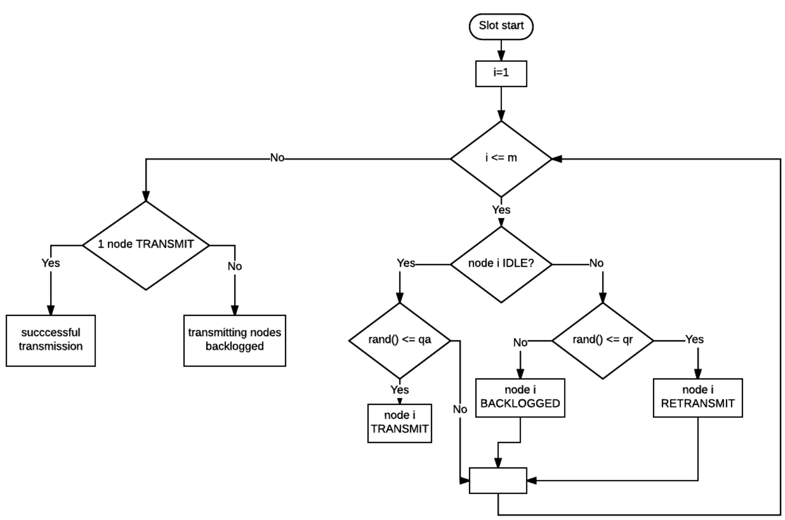
Rasmus Hedin
rashe877@student.liu.se
940524-3396

October 31, 2017

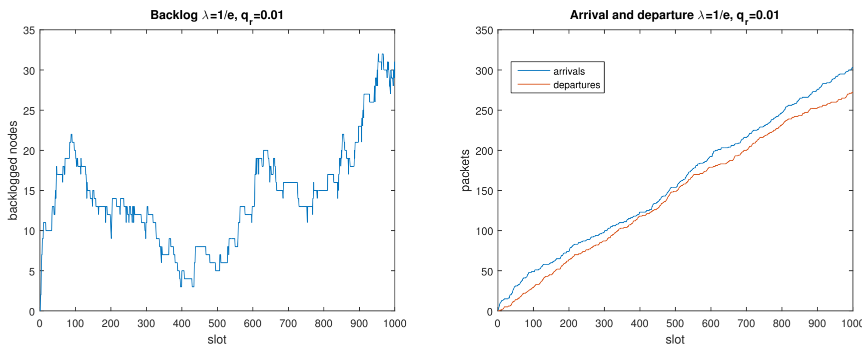
1 Assumptions

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

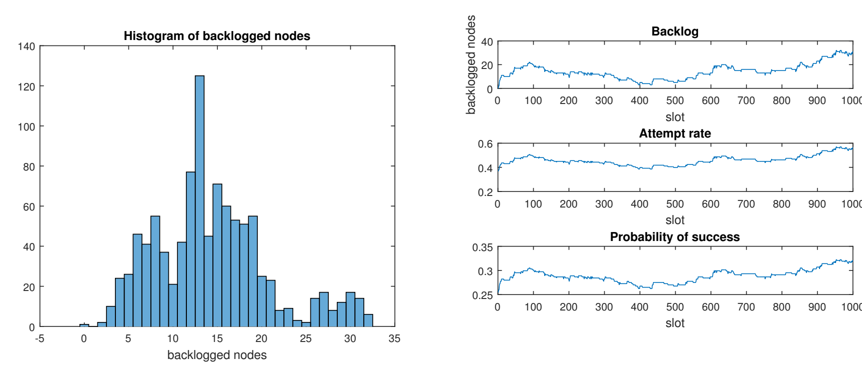


(a) Backlog

(b) Arrival and departure

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 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 departured 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.



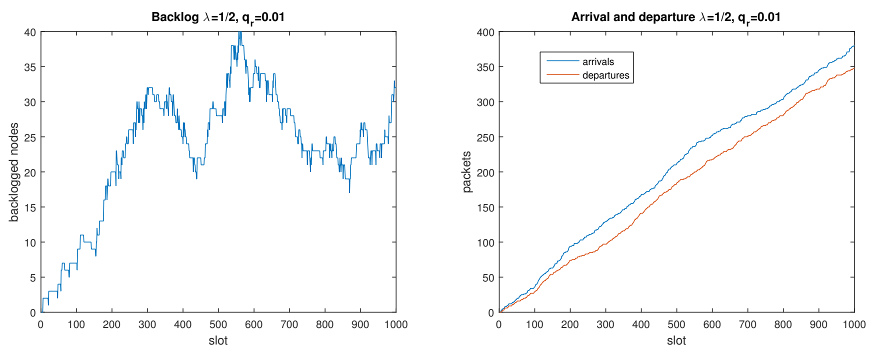
(a) Histogram of backlog

(b) Backlog, attempt rate, and probability 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 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 $\lambda = 1/2, q_r = 0.01$



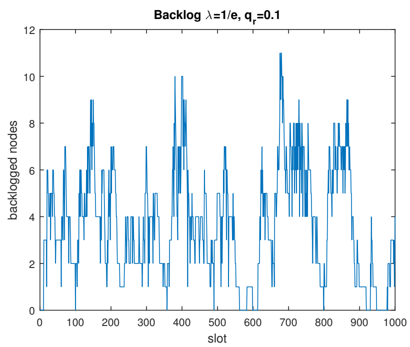
(a) Backlog (b) Arrival and departure

Figure 3

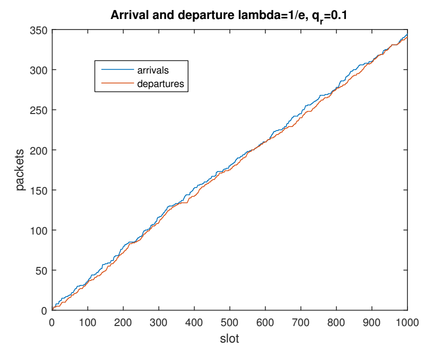
In this run of the simulation $\lambda = 1/2$ instead and q_r is kept at 0.01. this will increase the arrival rate and as we can see in figure 3a the backlog increases faster and the average number of backlogged nodes is also higher than in the previous plot. The delay have also increased since the gap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

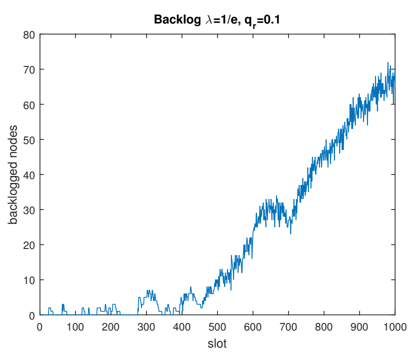


(a) Backlog

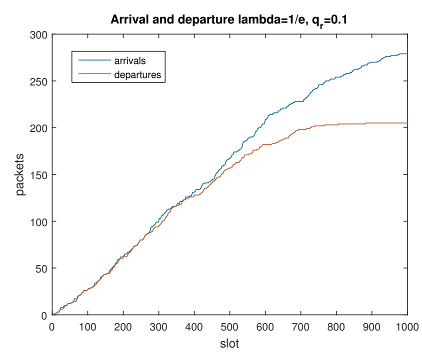


(b) Arrival and departure

Figure 4



(a) Backlog

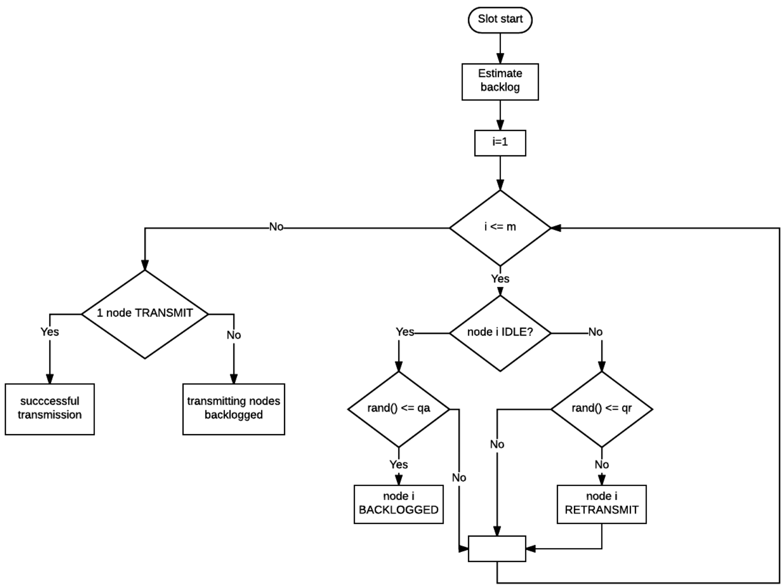


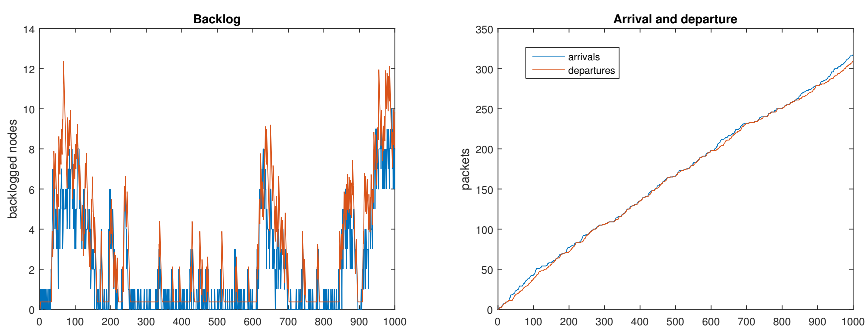
(b) Arrival and departure

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.

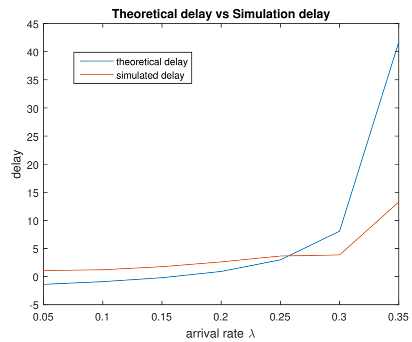


Figure 7: Approximate delay

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

```
function [backlog, arrival, departure] = slotted_aloha(m, T, ...
    lambda, qr)

% State of a nodes
% 0: Idle
% 1: Transmitting
% 2: Backlogged
state = zeros(1,m);

transmission_slot = zeros(1,m);
backlog = zeros(1,T);
arrival = zeros(1,T);
departure = zeros(1,T);
arrival_slot = zeros(1,T);
departure_slot = zeros(1,T);
arrivedPckts = 0;
transmPckts = 0;

% Calculate the probability of arrival
qa = 1 - exp(-lambda/m);

% Simulate for T slots
t = 0;
while t < T
    t = t + 1;

    for i = 1:m
        % Is node_i IDLE?
        if state(i) == 0
            % Packet arrive at node_i with probability qa
            if rand() ≤ qa
                state(i) = 1;
                transmission_slot(i) = t;
                arrivedPckts = arrivedPckts + 1;
                arrival_slot(arrivedPckts) = t;
            end
        else
            % Node i is backlogged
            % Retransmit with probability qr
            if rand() ≤ qr
                state(i) = 1;
            else
                % Node_i did not retransmit
                % and stays backlogged
                state(i) = 2;
                backlog(t) = backlog(t) + 1;
            end
        end
    end

    % How many nodes are transmitting?
    transmissions = 0;
    transmitting_node = -1;
```

```
    for i = 1:m
        if state(i) == 1
            transmissions = transmissions + 1;
            transmitting_node = i;
        end
    end

    % One node transmitting = SUCCESS
    if transmissions == 1
        state(transmitting_node) = 0;
        transmPckts = transmPckts + 1;
        departure_slot(transmPckts) = t;
    end

    % Store packet arrivals and departures
    % at time slot t
    arrival(t) = arrivedPckts;
    departure(t) = transmPckts;
end

% Calculate mean delay
W = departure_slot(1:transmPckts) - arrival_slot(1:transmPckts);
W = mean(W);

fprintf('\n=========================\n')
fprintf('Simulation finished\n')
fprintf('Nodes: %u\n',m);
fprintf('Slots %u\n',T);
fprintf('lambda: %.3f\n',lambda);
fprintf('qr: %.3f\n',qr);
fprintf('qa: %.3f\n',qa);
fprintf('Arrived packets: %u\n',arrivedPckts);
fprintf('Transmitted packets: %u\n',transmPckts);
fprintf('Mean delay: %.3f\n',W);
fprintf('=========================\n')
```

B Slotted ALOHA with Pseudo-Bayesian Stabilization

```
function [backlog, backlog_estimate, arrival, departure, W] = ...
    stabilized_slotted_aloha(m, T, lambda)

% State of a nodes
% 0: Idle
% 1: Transmitting
% 2: Backlogged
state = zeros(1,m);

transmission_slot = zeros(1,m);
backlog = zeros(1,T);
backlog_estimate = zeros(1,T+1);
arrival = zeros(1,T);
departure = zeros(1,T);
arrival_slot = zeros(1,T);
departure_slot = zeros(1,T);
arrivedPckts = 0;
transmPckts = 0;

% Calculate the probability of arrival
qa = 1 - exp(-lambda/m);

% Simulate for T slots
t = 0;
while t < T
    t = t + 1;

    % Update retransmission rate (qr)
    if 0 ≤ backlog_estimate(t) && backlog_estimate(t) < 1
        qr = 1;
    else
        qr = 1/backlog_estimate(t);
    end

    for i = 1:m
        % Is node_i IDLE?
        if state(i) == 0
            % Packet arrive at node_i with probability qa
            if rand() ≤ qa
                state(i) = 2;
                transmission_slot(i) = t;
                arrivedPckts = arrivedPckts + 1;
                arrival_slot(arrivedPckts) = t;
                backlog(t) = backlog(t) + 1;
            end
        else
            % Node i is backlogged
            % Retransmit with probability qr
            if rand() ≤ qr
                state(i) = 1;
            else
```

```
                % Node_i did not retransmit
                % and stays backlogged
                state(i) = 2;
                backlog(t) = backlog(t) + 1;
            end
        end
    end

    % How many nodes are transmitting?
    transmissions = 0;
    transmitting_node = -1;
    for i = 1:m
        if state(i) == 1
            transmissions = transmissions + 1;
            transmitting_node = i;
        end
    end

    % One node transmitting = SUCCESS
    if transmissions == 1
        state(transmitting_node) = 0;
        transmPckts = transmPckts + 1;
        departure_slot(transmPckts) = t;
    end

    % Estimate backlog for slot t+1
    if transmissions ≤ 1
        backlog_estimate(t+1) = max(lambda, backlog_estimate(t) ...
            + lambda - 1);
    else
        backlog_estimate(t+1) = backlog_estimate(t) + lambda + ...
            1/(exp(1)-2);
    end

    % Store packet arrivals and departures
    % at time slot t
    arrival(t) = arrivedPckts;
    departure(t) = transmPckts;
end

% Calculate mean delay
W = departure_slot(1:transmPckts) - arrival_slot(1:transmPckts);
W = mean(W);

fprintf('\n=========================\n')
fprintf('Simulation finished\n')
fprintf('Nodes: %u\n',m);
fprintf('Slots %u\n',T);
fprintf('lambda: %.3f\n',lambda);
fprintf('qa: %.3f\n',qa);
fprintf('Arrived packets: %u\n',arrivedPckts);
fprintf('Transmitted packets: %u\n',transmPckts);
fprintf('Mean delay: %.3f\n',W);
fprintf('=========================\n')
```

C Plots

```
%%%%%%%%%%%%%%%%%%%%
%% Slotted ALOHA  %%
%%%%%%%%%%%%%%%%%%%%
m = 100;
T = 1000;
lambda = exp(-1);
qr = 0.01;
[backlog, arrival, departure] = slotted_aloha(m,T,lambda,qr);

%% Backlog
figure
plot(backlog);
title('Backlog {\lambda=1/e, q_r=0.01}');
xlabel('slot');
ylabel('backlogged nodes');
x = 1:T;

%% Arrival and departure of packets
figure
plot(x,arrival,x,departure);
title('Arrival and departure {\lambda=1/e, q_r=0.01}');
xlabel('slot');
ylabel('packets');
legend('arrivals','departures');

%% Histogram of backlog
figure
histogram(backlog);
title('Histogram of backlogged nodes');
xlabel('backlogged nodes');

%% Steady-states of the Markov chain
tbl = tabulate(backlog);
N = sum(tbl(1:end,1).*(tbl(1:end,3)/100));
D = N/lambda;

%% Attempt rate
qa = 1 - exp(-lambda/m);
G = attempt_rate(m, backlog, qa, qr);
figure
subplot(3,1,1);
plot(x,backlog);
title('Backlog');
xlabel('slot');
ylabel('backlogged nodes');
subplot(3,1,2);
plot(x,G);
title('Attempt rate');
xlabel('slot');

% Probability of success
G = attempt_rate(m, backlog, qa, qr);
Ps = G.*exp(-G);
```

```
Ps_avg = mean(Ps);
subplot(3,1,3);
plot(x,Ps);
title('Probability of success');
xlabel('slot');

%%
Pnew = (m-backlog)*qa;
figure
plot(x,Ps,x,Pnew);
title('Probability of new arrivals');

%%
G_theory = attempt_rate(m,0:m,qa,qr);
Ps_theory = G_theory.*exp(-G_theory);
Pnew_theory = (m-(0:m))*qa;
figure
plot(0:m,Ps_theory,0:m,Pnew_theory);
Ps_avg_theory = mean(Ps_theory);

%%
Dn_theory = Pnew_theory - Ps_theory;
figure
subplot(2,1,1);
plot(0:m,Ps_theory,0:m,Pnew_theory);
subplot(2,1,2);
plot(0:m,Dn_theory,0:m,zeros(1,m+1));

%% Assignment 5
m = 100;
T = 1000;
lambda = 1/2;
qr = 0.01;
[backlog, arrival, departure] = slotted_aloha(m,T,lambda,qr);

figure
plot(backlog);
title('Backlog {\lambda=1/2, q_r=0.01}');
xlabel('slot');
ylabel('backlogged nodes');
x = 1:T;
figure
plot(x,arrival,x,departure);
title('Arrival and departure {\lambda=1/2, q_r=0.01}');
xlabel('slot');
ylabel('packets');
legend('arrivals','departures');

%% Assignment 6
m = 100;
T = 1000;
lambda = exp(-1);
qr = 0.1;
[backlog, arrival, departure] = slotted_aloha(m,T,lambda,qr);

figure
plot(backlog);
```

```
title('Backlog {\lambda=1/e, q_r=0.1}');
xlabel('slot');
ylabel('backlogged nodes');
figure
plot(x,arrival,x,departure);
title('Arrival and departure {lambda=1/e, q_r=0.1}');
xlabel('slot');
ylabel('packets');
legend('arrivals','departures');


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%% Pseudo-Bayesian Stabilization  %%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
m = 100;
T = 1000;
x = 1:T;
lambda = exp(-1);
[backlog, backlog_estimate, arrival, departure, W] = ...
    stabilized_slotted_aloha(m,T,lambda);

%% Backlog and backlog estimate
figure
plot(x,backlog,x,backlog_estimate(1:1000));
title('Backlog');
xlabel('slot');
ylabel('backlogged nodes');

%% Arrival and departure
figure
plot(x,arrival,x,departure);
title('Arrival and departure');
xlabel('slot');
ylabel('packets');
legend('arrivals','departures');

%% Histogram of backlog
figure
histogram(backlog);
title('Histogram of backlogged nodes');

%% Steady-states of the Markov chain
tbl = tabulate(backlog);
N = sum(tbl(1:end,1).*(tbl(1:end,3)/100));
D = N/lambda;

%% Backlog, Attempt rate, and Probabiliy of success
qa = 1 - exp(-lambda/m);
G = attempt_rate(m, backlog, qa, qr);
figure
subplot(3,1,1);
plot(x,backlog);
title('Backlog');
xlabel('slot');
ylabel('backlogged nodes');
subplot(3,1,2);
plot(x,G);
```

```
title('Attempt rate');

% Probability of success
G = attempt_rate(m, backlog, qa, qr);
Ps = G.*exp(-G);
Ps_avg = mean(Ps);
subplot(3,1,3);
plot(x,Ps);
title('Probability of success');

%% Probability of new arrivals
Pnew = (m-backlog)*qa;
figure
plot(x,Ps,x,Pnew);
title('Probability of new arrivals');

%% Theoretical calculations
G_theory = attempt_rate(m,0:m,qa,qr);
Ps_theory = G_theory.*exp(-G_theory);
Pnew_theory = (m-(0:m))*qa;
figure
plot(0:m,Ps_theory,0:m,Pnew_theory);
Ps_avg_theory = mean(Ps_theory);

%% Approximate delay analysis
lambda = 0.05:0.05:0.35;
W_theory = average_delay(lambda);
figure
plot(lambda,W_theory);
title('');
xlabel('arrival rate {\lambda}')
ylabel('delay')

%% Average delay from simulation
lambda = 0.05:0.05:0.35;
W = zeros(1,length(lambda));
for i = 1:length(lambda)
    [backlog, backlog_estimate, arrival, departure, Wi] = ...
        stabilized_slotted_aloha(m,T,lambda(i));
    W(i) = Wi;
end
plot(lambda,W_theory, lambda,W);
title('Theoretical delay vs Simulation delay');
xlabel('arrival rate {\lambda}')
ylabel('delay')
legend('theoretical delay', 'simulated delay');
```

D Utility functions

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

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