Weining Bao weiningb@usc.edu 8287723653

Project 8: A simple switch

Consider a simple 3x3 (3 input, 3 output) switch that processes fixed length packets (blocks of data). Each packet has a destination address that determines which output port the packet is destined to. The switch fabric can be assumed to be non-blocking, i.e. it is able to deliver any arrangement of packets on the input ports to the associated output ports in a single clock cycle. Each output port is able to deliver one packet to the output line card in a clock cycle and has a buffer (of size *K* packets), so that when multiple packets are requesting the same output port it is able to buffer the packets that cannot be processed in this cycle. Assume that the probability that, in a clock cycle, a packet arrives to input port *i* (*i*=1,2,3) is $\lambda_i = \lambda$ (i.e. the same for all input ports) and the probability that a packet that arrives at port *i* is destined to port *j* is $r_{ij} = r = 1/3$ (i.e. all outputs ports are equally likely). If there is no buffer space available, the packet is dropped (packet loss). Find the probability of packet loss as a function of *K* and λ by simulation and validate your results by developing a discrete time model of the output buffer at one of the output ports. (Hint: focus on output port 1 (wlog) – there can be either 0, 1, 2, or 3 packets arriving to the port in a cycle. Let α_i be the probability of *i* packets arriving to this port. Find an expression for α_i in terms of λ (and r). Use this to develop your model.)

Matlab Simulation

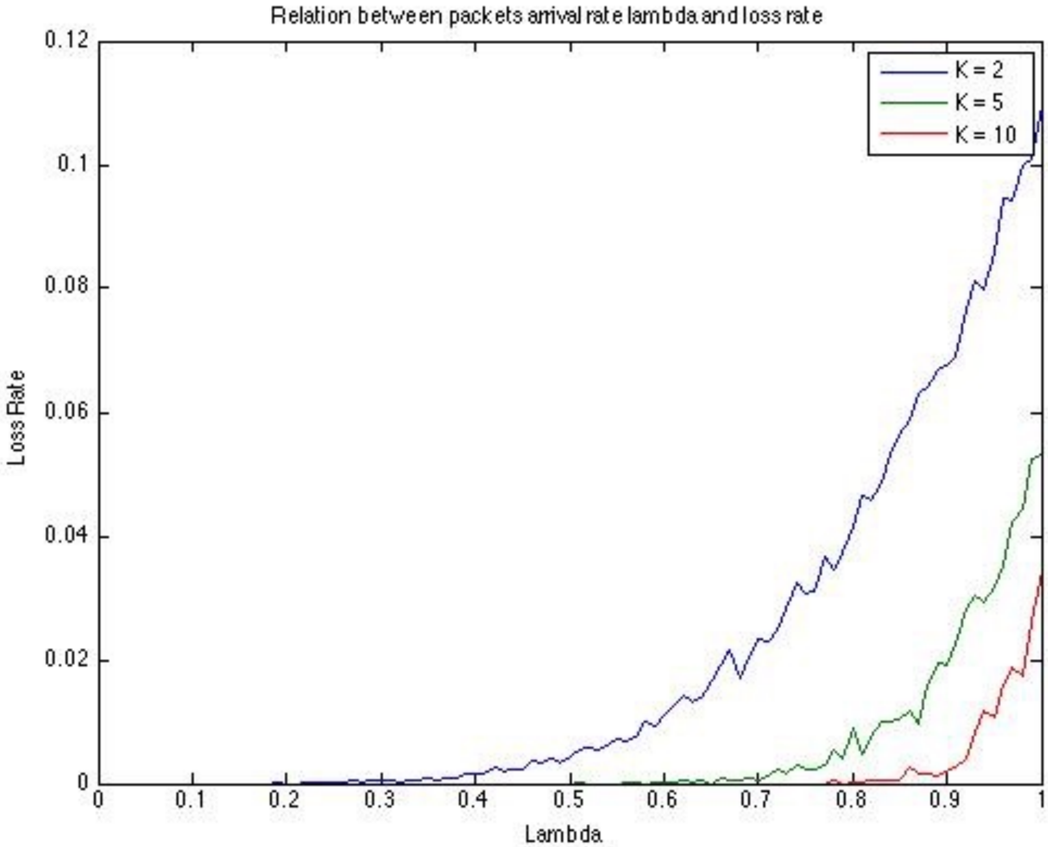
with code at last

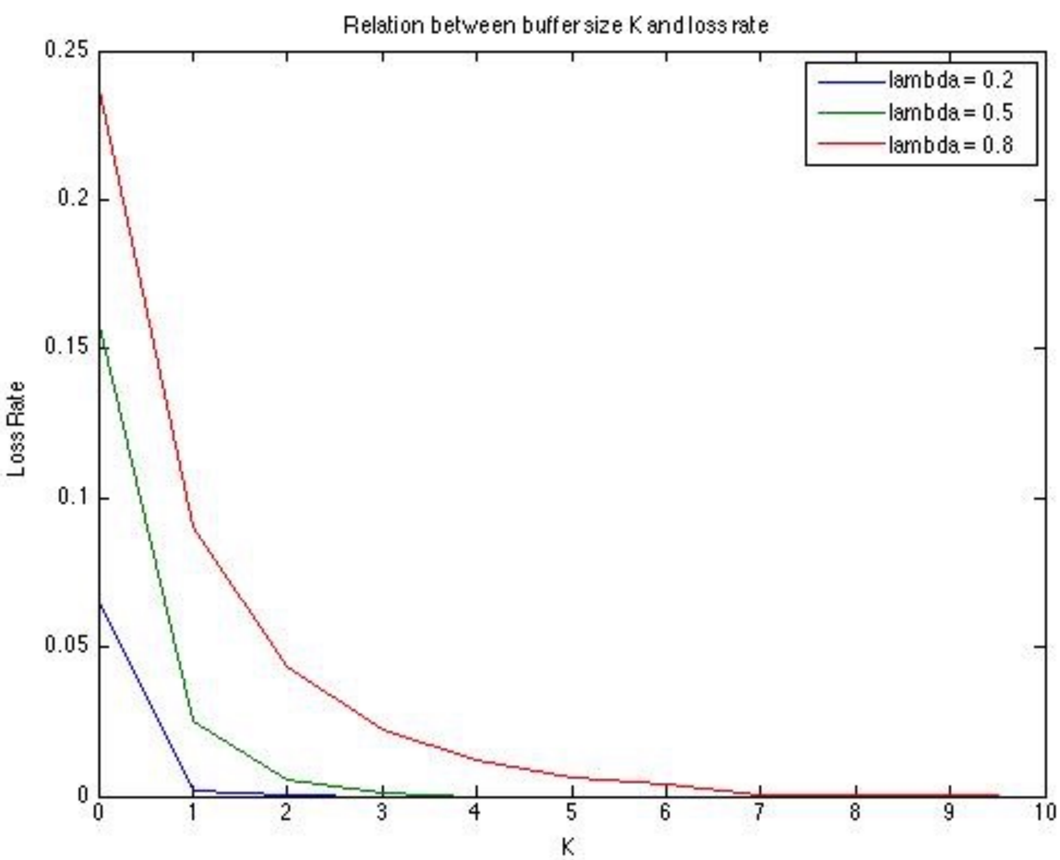
To get the relation between packet loss rate with packet arrival rate λ and buffer size K, two separate simulations are done, while one fixes λ and changes K, another vice versa. Number of packets in 3 servers and 3 buffers are all recorded. Number of packets arrived are also recorded. Every clock cycle number of packets in servers and buffers will be updated first, then packets will arrive at different input ports according to λ and change the number of packets in buffers only. If the number of packets in buffers already reaches buffer size K, the new arriving packets will be dropped and the number of lost packets is recorded. The resulting graphs are in next page.

One interesting facts: If $\lambda = 0$, the simulation will never stop because no packet comes in this event-driven simulation;

One thing to note: loss rate is not always 1 when K = 0 because if there is no packet in server, the arriving packet can directly go there and not got dropped.

Discrete time model of one output buffer is developed at the last handwriting page.


Relation between packets arrival rate lambda and loss rate
K = 2
K = 5
K = 10
Loss Rate
Lambda
0.12
0.1
0.08
0.06
0.04
0.02
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1


Relation between buffer size K and loss rate
lambda = 0.2
lambda = 0.5
lambda = 0.8
Loss Rate
K
0.25
0.2
0.15
0.1
0.05
0
1
2
3
4
5
6
7
8
9
10

Code:

```
number = 10000; % number of packets
step = 100; % number of step to test for both k and lambda

% k fixed

klist = [2, 5, 10]; % fixed k value to be test when lambda as x-axis
lambda = (1/step):(1/step):1; % lambda value to be test, lambda not zero
figure;

for i = 1:3
    k = klist(i);
    l = zeros(1, step); % loss rate for corresponding lambda
    for j = 1:step
        loss = 0; % number of packets lost
        b = zeros(1, 3); % number of packets in buffers
        s = zeros(1, 3); % number of packets in servers
        p = 0; % number of packets arrived
        lam = lambda(j);
        while p < number % one clock cycle
            for m = 1:3 % change in 3 buffers and servers
                if b(m) > 0
                    b(m) = b(m) - 1;
                    s(m) = 1;
                else
                    s(m) = 0;
                end
            end
            for m = 1:3 % 3 input ports
                if rand() < lam
                    p = p + 1;
                    curr = rand(); % decide to which out port
                    if curr < 1/3 % r = 1/3
                        if b(1) == 0 && s(1) == 0
                            s(1) = 1;
                        elseif b(1) == k
                            loss = loss + 1;
                        else
                            b(1) = b(1) + 1;
                        end
                    elseif curr < 2/3
                        if b(2) == 0 && s(2) == 0
                            s(2) = 1;
                        elseif b(2) == k
                            loss = loss + 1;
                        else
                            b(2) = b(2) + 1;
                        end
```

```
                else
                    if b(3) == 0 && s(3) == 0
                        s(3) = 1;
                    elseif b(3) == k
                        loss = loss + 1;
                    else
                        b(3) = b(3) + 1;
                    end
                end
            end
        end
    end
    l(j) = loss / p;
  end
  plot(lambda, l); hold all;
end
xlabel('Lambda');
ylabel('Loss Rate');
title('Relation between packets arrival rate lambda and loss rate');
legend('K = 2', 'K = 5', 'K = 10');
hold off;

% lambda fixed
step = 10;

llist = [0.2, 0.5, 0.8]; % fixed lambda value to be test when k as x-axis
kl = 0:step; % k value to be test, loss always = 1 when k = 0
figure;

for i = 1:3
  lam = llist(i);
  l = zeros(1, step+1); % loss rate for corresponding lambda
  for j = 1:step+1
    loss = 0; % number of packets lost
    b = zeros(1, 3); % number of packets in buffers
    s = zeros(1, 3); % number of packets in servers
    p = 0; % number of packets arrived
    k = kl(j);
    while p < number % one clock cycle
      for m = 1:3 % change in 3 buffers and servers
        if b(m) > 0
          b(m) = b(m) - 1;
          s(m) = 1;
        else
          s(m) = 0;
        end
      end
      for m = 1:3 % 3 input ports
        if rand() < lam
```

```
                p = p + 1;
                curr = rand(); % decide to which out port
                if curr < 1/3 % r = 1/3
                    if b(1) == 0 && s(1) == 0
                        s(1) = 1;
                    elseif b(1) == k
                        loss = loss + 1;
                    else
                        b(1) = b(1) + 1;
                    end
                elseif curr < 2/3
                    if b(2) == 0 && s(2) == 0
                        s(2) = 1;
                    elseif b(2) == k
                        loss = loss + 1;
                    else
                        b(2) = b(2) + 1;
                    end
                else
                    if b(3) == 0 && s(3) == 0
                        s(3) = 1;
                    elseif b(3) == k
                        loss = loss + 1;
                    else
                        b(3) = b(3) + 1;
                    end
                end
            end
        end
    end
    l(j) = loss / p;
  end
  plot(kl, l); hold all;
end
xlabel('K');
ylabel('Loss Rate');
title('Relation between buffer size K and loss rate');
legend('lambda = 0.2', 'lambda = 0.5', 'lambda = 0.8');
hold off;
```