

## Network Engineering - Final exam - 2020/2021

**Preliminaries:**  $x_1 = 1$ st number of your NIA;  $x_2 = 2$ nd number of your NIA;  $x_3 = 3$ rd number of your NIA;  $x_4 = 4$ th number of your NIA, etc. If any number is zero, consider it equal to 1.

Consider a Home Wi-Fi network such as the depicted in Figure 1 where two Access Points (APs) are deployed in the same house to provide a good coverage everywhere. It consists of two APs, A and B, and three stations: C, D and E. The two APs are connected to Internet through a router/switch which forwards the traffic to the corresponding AP depending on the destination station. Station C is associated to AP A, and stations D and E are associated to AP B. AP A and B operate in different channels. Each station receives a traffic flow. The information related to each station can be found in Table 1. The buffer of AP A and AP B is large enough to be considered as infinite. The traffic arrival process in all cases is Poisson.

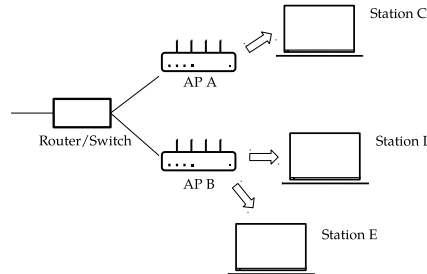


Figure 1: Network

Station	App description	Traffic Load	Packet Size	Link capacity (AP→ STA)
C	Video	see Problem 1	$L=4000 \cdot x_2$ bits, $CV_L = 1$	$R_{A \rightarrow C} = 18$ Mbps
D	Cloud gaming	$6 \cdot x_4$ Mbps	$L=500$ Bytes, $CV_L = 0$	$R_{B \rightarrow D} = 10 \cdot x_4$ Mbps
E	File transfer	$2 \cdot x_4$ Mbps	$L=10000 \cdot x_3$ bits, $CV_L = 0$	$R_{B \rightarrow E} = 7 \cdot x_4$ Mbps

Table 1: Data

### Problem 1 (40 mins) - 5 points

The load (in Mbps) send to Station C changes randomly every 4 seconds. Let us call  $X(t)$  the random variable modelling the load in Mbps. The set of values that  $X$  can take is given by  $\mathcal{X} = \{1, 4, 8, 16\}$  Mbps. Transitions happen only every 4 seconds, and are between consecutive states (i.e., from 4 we can only move to 1, to 8, or to stay in 4). The following transition probabilities are known:

- 1 Mbps  $\rightarrow$  4 Mbps: 0.4; 4 Mbps  $\rightarrow$  1 Mbps:  $\min(0.4, x_5/10)$ ;
- 4 Mbps  $\rightarrow$  8 Mbps: 0.2; 8 Mbps  $\rightarrow$  4 Mbps:  $\min(0.4, x_5/10)$ ;
- 8 Mbps  $\rightarrow$  16 Mbps: 0.1; 16 Mbps  $\rightarrow$  8 Mbps:  $\min(0.4, x_5/10)$ ;

- (1 point) Draw the Markov chain modelling the stochastic process  $X(t)$ .
- (1.5 point) Calculate the stationary probability of the stochastic process  $X(t)$ .
- (0.5 point) Calculate the expected load of the traffic flow directed to station C.
- (2 points) Calculate the packet delay,  $\mathbb{E}[D] = \mathbb{E}[D_s] + \mathbb{E}[D_q]$ , for each possible value of  $X(t)$ , and then do the average over all possible values. Compare it with the case where the delay is calculated using the expected load (from previous question). Which approach do you think is better?

### Problem 2 (40 mins) - 5 points

Station D and E share AP B.

- (1.5 points) Given that the packet error probability is  $p = \min(0.2, x_1/10)$ , and considering that the maximum number of retransmission per packet is  $R_{\max} = \min(3, x_3)$ , calculate the expected service time of the packets directed to station D and to station E. Calculate also the second moment of the service time for both cases, and the coefficient of variation.
- (0.5 points) Is AP B stable with the provided values? If not, propose new load values for station D and E that guarantee the stability of AP B.

3. (1 point) Assuming no traffic differentiation is applied at AP B, calculate the expected packet delay for the packets directed to station D and to station E, i.e.,  $\mathbb{E}[D_D]$  and  $\mathbb{E}[D_E]$ .
4. (1 point) Assuming traffic differentiation is applied at AP B, giving priority to the cloud gaming traffic (as it requires low delay), calculate the expected packet delay for the packets directed to station D and to station E, i.e.,  $\mathbb{E}[D_D]$  and  $\mathbb{E}[D_E]$ . Explain the results.
5. (1 point) Let us assume station D equips two Wi-Fi interfaces, and so it could be 'associated' to AP A and B at the same time. Explain (no need to do any calculation) how and why that situation may help to improve or not the WLAN performance.

```
function NE_FinalExam_20202021()

%Problem1();
Problem2();

end

% -----
function Problem1()

disp('----- Exercise 1 -----');

x1=1;
x2=2;
x3=4;
x4=6;
x5=3;

% 1)

% (1) - (4) - (8) - (16)

% Transition Probabilities
P = [0.6 0.4 0 0;
     0.3 0.5 0.2 0;
     0 0.3 0.6 0.1;
     0 0 0.3 0.7;]

% 2)

% Local balance equations

auxpi4 = (0.4/0.3);
auxpi8 = (0.2/0.3)*(0.4/0.3);
auxpi16 = (0.1/0.3)*(0.2/0.3)*(0.4/0.3);

pi1 = 1 / (1 + auxpi4 + auxpi8 + auxpi16);
pi4 = auxpi4 * pi1;
pi8 = auxpi8 * pi1;
pi16 = auxpi16 * pi1;

disp('Stationary Probabilities (+ sum)');
disp([pi1 pi4 pi8 pi16 sum(pi1 + pi4 + pi8 + pi16)]);

% 3)

EloadC = 1E6*pi1 + 4E6*pi4 + 8E6*pi8 + 16E6*pi16;

disp('E[Load flow C]');
```

```

disp(EloadC);

% 4) The AP can be modelled as an MM1 queue

EL = 4000*x2;
mu = 18E6/EL;
Elambda = EloadC / EL;
lambda1 = 1E6/EL;
lambda4 = 4E6/EL;
lambda8 = 8E6/EL;
lambda16 = 16E6/EL;

% Delay average all loads

ED_Eloads = 1 / (mu-Elambda);

% Delays per load

ED1 = 1 / (mu-lambda1)
ED4 = 1 / (mu-lambda4)
ED8 = 1 / (mu-lambda8)
ED16 = 1 / (mu-lambda16)

EED = pi1*ED1 + pi4*ED4 + pi8*ED8 + pi16*ED16;
disp('Delays: Average | Average per load');
disp([ED_Eloads EED]);

% Considering the average load we are balancing the load over all the time,
% and that's why the results are optimistic (we are moving load from
% high-load periods to low-load ones).

end

% -----
function Problem2()

disp('----- Exercise 2 -----');

x1=1;
x2=2;
x3=4;
x4=6;
x5=3;

p=0.1;
Rmax=3;

LD = 500*8;
RD = 10*x4*1E6;
LoadD = 6*x4*1E6;

LE = 10000*x3;
RE = 7*x4*1E6;
LoadE = 2*x4*1E6;

% 1)

% Flow B --> D
DsD = LD/RD;
EDsDRet = (1-p)*DsD + p*(1-p)*2*DsD + p^2*(1-p)*3*DsD + p^3*4*DsD;
EDs2DRet = (1-p)*DsD^2 + p*(1-p)*(2*DsD)^2 + p^2*(1-p)*(3*DsD)^2 + p^3*(4*DsD)^2;

```

```

VarD = EDs2DRet - EDsDRet^2;
CVD = sqrt(VarD)/EDsDRet;

disp('Flow D: 1st moment | 2nd moment | CV');
disp([EDsDRet EDs2DRet CVD]);

% Flow B --> E
DsE = LE/RE;
EDsERet = (1-p)*DsE + p*(1-p)*2*DsE + p^2*(1-p)*3*DsE + p^3*4*DsE;
EDs2ERet = (1-p)*DsE^2 + p*(1-p)*(2*DsE)^2 + p^2*(1-p)*(3*DsE)^2 + p^3*(4*DsE)^2;

VarE = EDs2ERet - EDsERet^2;
CVE = sqrt(VarE)/EDsERet;

disp('Flow E: 1st moment | 2nd moment | CV');
disp([EDsERet EDs2ERet CVE]);

% 2)

% EDq = EDr/(1-a);

lambdaD = LoadD / LD;
lambdaE = LoadE / LE;

aD = lambdaD * EDsDRet;
aE = lambdaE * EDsERet;

a=aD+aE;

disp('Is stable? [aD aE a]');
disp([aD aE a]);

% For p=0.2, it is not stable. So, the solution is to 'reduce' the traffic
% load of station D or E.

% 3)

EDrD = aD * EDs2DRet/(2*EDsDRet);
EDrE = aE * EDs2ERet/(2*EDsERet);

EDr = EDrD + EDrE;

EDq = EDr / (1-a);

EDD = EDsDRet + EDq;
EDE = EDsERet + EDq;

disp('Delays without traffic diff. ');
disp([EDq EDD EDE]);

% 4) Priorities

EDqD = EDr / (1-aD);
EDqE = EDr / ((1-aD)*(1-aD-aE));

EDD = EDsDRet + EDqD;
EDE = EDsERet + EDqE;

disp('Delays Priorities');
disp([EDqD EDD EDqE EDE]);

```

```

function NE_FinalExam_20202021()

%Problem1();
Problem2();

end

% -----
function Problem1()

disp('----- Exercise 1 -----');

x1=1;
x2=2;
x3=4;
x4=6;
x5=3;

% 1)

% (1) - (4) - (8) - (16)

% Transition Probabilities
P = [0.6 0.4 0 0;
     0.3 0.5 0.2 0;
     0 0.3 0.6 0.1;
     0 0 0.3 0.7;]

% 2)

% Local balance equations

auxpi4 = (0.4/0.3);
auxpi8 = (0.2/0.3)*(0.4/0.3);
auxpi16 = (0.1/0.3)*(0.2/0.3)*(0.4/0.3);

pi1 = 1 / (1 + auxpi4 + auxpi8 + auxpi16);
pi4 = auxpi4 * pi1;
pi8 = auxpi8 * pi1;
pi16 = auxpi16 * pi1;

disp('Stationary Probabilities (+ sum)');
disp([pi1 pi4 pi8 pi16 sum(pi1 + pi4 + pi8 + pi16)]);

% 3)

EloadC = 1E6*pi1 + 4E6*pi4 + 8E6*pi8 + 16E6*pi16;

disp('E[Load flow C]');
disp(EloadC);

% 4) The AP can be modelled as an MM1 queue

EL = 4000*x2;
mu = 18E6/EL;
Elambda = EloadC / EL;
lambda1 = 1E6/EL;
lambda4 = 4E6/EL;
lambda8 = 8E6/EL;
lambda16 = 16E6/EL;

```

```

% Delay average all loads

ED_Eloads = 1 / (mu-Elambda);

% Delays per load

ED1 = 1 / (mu-lambda1)
ED4 = 1 / (mu-lambda4)
ED8 = 1 / (mu-lambda8)
ED16 = 1 / (mu-lambda16)

EED = pi1*ED1 + pi4*ED4 + pi8*ED8 + pi16*ED16;
disp('Delays: Average | Average per load');
disp([ED_Eloads EED]);

% Considering the average load we are balancing the load over all the time,
% and that's why the results are optimistic (we are moving load from
% high-load periods to low-load ones).

end

% -----
function Problem2()

disp('----- Exercise 2 -----');

x1=1;
x2=2;
x3=4;
x4=6;
x5=3;

p=0.1;
Rmax=3;

LD = 500*8;
RD = 10*x4*1E6;
LoadD = 6*x4*1E6;

LE = 10000*x3;
RE = 7*x4*1E6;
LoadE = 2*x4*1E6;

% 1)

% Flow B --> D
DsD = LD/RD;
EDsDRet = (1-p)*DsD + p*(1-p)*2*DsD + p^2*(1-p)*3*DsD + p^3*4*DsD;
EDs2DRet = (1-p)*DsD^2 + p*(1-p)*(2*DsD)^2 + p^2*(1-p)*(3*DsD)^2 + p^3*(4*DsD)^2;

VarD = EDs2DRet - EDsDRet^2;
CVD = sqrt(VarD)/EDsDRet;

disp('Flow D: 1st moment | 2nd moment | CV');
disp([EDsDRet EDs2DRet CVD]);

% Flow B --> E
DsE = LE/RE;
EDsERet = (1-p)*DsE + p*(1-p)*2*DsE + p^2*(1-p)*3*DsE + p^3*4*DsE;
EDs2ERet = (1-p)*DsE^2 + p*(1-p)*(2*DsE)^2 + p^2*(1-p)*(3*DsE)^2 + p^3*(4*DsE)^2;

VarE = EDs2ERet - EDsERet^2;

```

```

CVE = sqrt(VarE)/EDsERet;

disp('Flow E: 1st moment | 2nd moment | CV');
disp([EDsERet EDs2ERet CVE]);

% 2)

% EDq = EDr/(1-a);

lambdaD = LoadD / LD;
lambdaE = LoadE / LE;

aD = lambdaD * EDsDRet;
aE = lambdaE * EDsERet;

a=aD+aE;

disp('Is stable? [aD aE a]');
disp([aD aE a]);

% For p=0.2, it is not stable. So, the solution is to 'reduce' the traffic
% load of station D or E.

% 3)

EDrD = aD * EDs2DRet/(2*EDsDRet);
EDrE = aE * EDs2ERet/(2*EDsERet);

EDr = EDrD + EDrE;

EDq = EDr / (1-a);

EDD = EDsDRet + EDq;
EDE = EDsERet + EDq;

disp('Delays without traffic diff. ');
disp([EDq EDD EDE]);

% 4) Priorities

EDqD = EDr / (1-aD);
EDqE = EDr / ((1-aD)*(1-aD-aE));

EDD = EDsDRet + EDqD;
EDE = EDsERet + EDqE;

disp('Delays Priorities');
disp([EDqD EDD EDqE EDE]);

% 5) Balancing the traffic to D through AP A and B would improve the
% network operation, as both links will deal with a lower load, resulting
% in overall less delay. Observe also that the occupation of AP A is low
% (a_A = 0.317) which will allow to move most of the cloud gaming traffic
% to AP A. It will depend in any case also on the capacity of the link between AP A and D.

end

```