Assignment No. 2 Report

Impact of Transmission Errors in the Performance of a Wireless Link

Desempenho e Dimensionamento de Redes
MIECT, DETI, UA

Gabriel Silva (85129) & Gonçalo Vítor (85119)

Task 1

a) Link state upon receiving a frame with errors

The first step in calculating this is knowing the probability of receiving a frame with errors both in normal and interference state. This is done by calculating the probability of receiving a frame with no errors at all, and then calculating the complementary. Knowing that the bit error rate in the normal state is 10^-7, the chance of a bit being correct is the complementary of that: 1 - 10^-7. For an entire frame to be correct, all bits have to be correct (in our case there are 128*8 bits). This is simply the probability of one bit being correct raised to the power of the number of bits in a frame. The same logic is applied to the interference state.

So now we have both the probability of a packet having any number of errors in the normal and interference state. Since the link being in normal or interference state are mutually exclusive events, we can use the Bayes Law to calculate the chance of the link being in normal/interference state knowing that a frame was received with errors.

For that we need to know the chance of a packet being received with errors (regardless of the state of the link), which is the sum of the probability of the packet being received with errors in the normal state times the probability of the link being in the normal state plus the same probability in the interference state.

Finally, to obtain the probability of being in a specific state knowing that a frame was received with errors, we need only to use the chance of a packet being received with errors in that same state times the chance of the link being in that state and divide it by the chance of receiving a packet with errors.

With this in mind we reached the following results:

	p(normal)	p(interference)
p = 99%	1.5568%	98.4432%
p = 99.9%	13.7615%	86.2385%
p = 99.99%	61.4969%	38.5031%
p = 99.999%	94.1084%	5.8916%

The results were produced with the following Matlab code:

```
en = 1e-7; % bit error rate normal state
ei = 1e-3; % bit error rate interference state
%1.a
p = [0.99 0.999 0.9999 0.99999];
package_oneplus_errors_normal = 1 -(1*(1-en)^(128*8)); % probability of having one or more errors in a frame in normal state
package_oneplus_errors_interf = 1- (1*(1-ei)^(128*8)); % probability of having one or more errors in a frame in interference state

normal = ((package_oneplus_errors_normal*p) ./ (package_oneplus_errors_normal*p + package_oneplus_errors_interf*(1-p)))*100
interf = ((package_oneplus_errors_interf*(1-p)) ./ (package_oneplus_errors_normal*p + package_oneplus_errors_interf*(1-p)))*100
```

Conclusions

We can conclude that, the probability of the link being in the normal state when receiving a frame with errors increases as the chance of being in the normal state increases (tending to 100%). For the interference state, the probability of the link being in the interference state when receiving a frame with errors increases as the chance of being in the interference state increases, or as seen in the table, as the chance of being in the normal state decreases.

This is easily understandable if we think about a link that as 100% or 0% chance of being in the normal state. For a link with 100% chance of being in the normal state the probability of it being in normal state when receiving a frame with errors has to be 100%. The same thought can be applied to the 0% example. These two examples match perfectly with our results.

b) False positives

A false positive occurs when the channel receives n consecutive frames with errors in the normal state. For this, we need to know the probability of receiving n frames with errors in a specific state and the probability of receiving n frames with errors in any state. In this case we use the probability of receiving a packet with errors to calculate the chance of receiving n consecutive packets with errors. This is done by raising the probability of receiving a frame with errors in a given state to the power of the number of consecutive frames we are considering. The probability of receiving n consecutive packets with errors regardless of state is calculated as in the previous example, with the sum of both events (receiving n frames with error in normal state and receiving n frames with error in interference state).

The probability of a false positive is the conditional probability of being the normal state given that all n frames were received with errors. This is calculated by using the chance of n packets being received with errors in normal state times the chance of the link being normal state and divided by the chance of receiving n packets with errors.

With this in mind we reached the results:

	n=2	n=3	n=4	n=5
p = 99%	0.00025%	4.03e-8%	6.45e-12%	1e-15%
p = 99.9%	0.0025%	4.07e-7%	6.5e-11%	1e-14%
p = 99.99%	0.0251%	4.08e-6%	6.51e-10%	1.04e-13%
p = 99.999%	0.2545%	4.08e-5%	6.51e-9%	1.04e-12%

The results were produced with the following Matlab code:

```
%1.b
en = 1e-7; % bit error rate normal state
ei = 1e-3; % bit error rate interference state
p = [0.99 0.999 0.9999 0.99999];
package_oneplus_errors_normal = 1 -(1*(1-en)^(128*8)); % probability of having one or more errors in a frame in normal state
package_oneplus_errors_interf = 1- (1*(1-ei)^(128*8)); % probability of having one or more errors in a frame in interference state
n = [2 3 4 5];
prob_n_errors_normal = package_oneplus_errors_normal .^ n;
prob_n_errors_inter = package_oneplus_errors_interf .^ n;

false_pos = ((prob_n_errors_normal'*p) ./ (prob_n_errors_normal'*p + (prob_n_errors_inter')*(1-p)))' * 100
```

c) False negatives

A false negative occurs when the channel receives at least one frame, within n consecutive frames, without errors in the interference state. The steps needed to calculate a false negative are the same for the false positives, except now we need the probability of at least one frame not having errors in n consecutive frames, instead of all n frames having errors. The first is the complementary of the second.

With this in mind we reached the results:

	n=2	n=3	n=4	n=5
p = 99%	0.5915%	0.738%	0.832%	0.892%
p = 99.9%	0.0589%	0.0736%	0.0831%	0.0892%
p = 99.99%	0.00589%	0.00736%	0.00831%	0.00892%
p = 99.999%	0.000589%	0.000736%	0.000831%	0.000892%

The results were produced with the following Matlab code:

```
%1.c
en = 1e-7; % bit error rate normal state
ei = 1e-3; % bit error rate interference state
p = [0.99 0.999 0.9999 0.99999];
package_oneplus_errors_normal = 1 -(1*(1-en)^(128*8)); % probability of having one or more errors in a frame in normal state
package_oneplus_errors_interf = 1- (1*(1-ei)^(128*8)); % probability of having one or more errors in a frame in interference state
n = [2 3 4 5];
p_n = 1 - (package_oneplus_errors_normal .^ n);
p_i = 1 - (package_oneplus_errors_interf .^ n);

false_negs = ((p_i'*(1-p)) ./ ((p_n')*p + p_i'*(1-p)))' * 100
```

d) Influence of p and n on results

By looking at the results we notice that as n increases the chance of false positives decreases. This makes perfect sense, since the more packets we're expecting the harder it is for all of the them to have errors. We see the opposite effect in false negatives: the larger n is the higher the chance of false negatives is. This because, the more packets we're expecting, more likely it becomes that at least one of them does not have errors.

For p we notice that as p increases the chance to have false positives increases and the chance to have false negatives decreases. Since to have false positives the link has to be in the normal state, it is only natural that the higher the chance to be in the normal state the higher the chance to have false positives. As the chance to be in normal state increases the chance to be in interference state decreases, which by a similar rationale explains why the false negative chance decreases as p increases.

e) Best value of n for p = 99.999%

First we need to consider that the chance of being in normal state is much higher than to be in interference state. Having this said, there should be a higher focus in avoiding false positives than false negatives. For the false negatives, all values of n satisfy the required condition (having false positives/negatives < 0.1%). For the false positives only 3,4 and 5 satisfy this condition. Since the probability of false positives decreases as n increases, the best n to use in this situation would be 5. It is also worth noticing that for the false negatives (for n equal to 3,4 and 5) all percentages are of the same order and very close to each other.

Task 2

a) Average percentage of time for each state

This can be calculated by using the birth-death Markov chain formulas. We identified the states with numbers from 0 to 4, starting from the left.

The results:

state	percentage
0	98.7%
1	0.55%
2	0.27%
3	0.14%
4	0.34%

The results were produced with the following Matlab code (in the code the results are not in percentage for later use):

```
%2.a
p0 = 1/(1 + (1/180 + 1/180*20/40 + 1/180*20/40*10/20 + 1/180*20/40*10/20*5/2))
p1 = p0*1/180
p2 = p0*(1/180*20/40)
p3 = p0*(1/180*20/40*10/20)
p4 = p0*(1/180*20/40*10/20*5/2)
```

b) Bit error rate of the link

The bit error rate is given by a weighted mean in which the values are the bit error rate of each state and the weights are the probability of being in each state (calculated in the previous exercise). In this particular case, the sum of all weights is 1.

The result is 3.6957e-05.

The result was produced with the following Matlab code:

```
%2.b
ps = [p0 p1 p2 p3 p4];
errors = [1e-6 1e-5 1e-4 1e-3 1e-2];
avg_bit_error = sum(errors .* ps) / sum(ps) %sum(ps) is 1 in this case
```

c) Average time duration (minutes) of each state of the link

The time duration of a state can be calculated by inverting the sum of all transition rates from that state to other states.

The results are:

state	minutes
0	60
1	0.3
2	1.2
3	2.4
4	30

The results were produced with the following Matlab code:

```
%2.c
t0 = 1/1 * 60
t1 = 1/(180+20) * 60
t2 = 1/(40+10) * 60
t3 = 1/(5+20) * 60
t4 = 1/2 * 60
```

d) Probability of link being in interference state

The probability of the link being in interference state is simply the sum of the probabilities of all states that are a part of the interference state, in our case, state 3 and state 4.

The result is then 0.48%.

The result was produced with the following Matlab code:

```
%2.d
pint = p3 + p4
```

e) Bit error rate in interference state

The way to calculate the bit error rate in the interference state is equal to the way we used before. The difference is that now we can only consider the bit error rates for states 3 and 4 (the interference states) and their corresponding probabilities. In this case, unlike before, the sum of all weights will not be 1.

The result is 0.0074.

The result was produced with the following Matlab code:

```
%2.e
ps = [p3 p4];
errors = [1e-3 1e-2];
avg_bit_error_int = sum(errors.*ps) / sum(ps) %sum(ps) is not 1 in this case
```

f) Average time duration (minutes) of interference state

To calculate the average time of the interference state, first we need to know the probability of going from state 3 to state 2 (exit interference state), of going from state 3 to state 4 and from state 4 back to state 3 (both are staying in interference state). These probabilities are the rate at which a state transitions to a second state divided by the sum of all transitions that leave the first state.

Then a loop is needed to calculate the time spent from going from state 3 to state 4, then back to state 3 and back to state 4, and so on n times. After not even 10 times this time value stagnates. The way to calculate this is to raise the product between the probability to go from state 3 to state 4 and from state 4 to state 3 to the power of n, where n is how many times this cycle happens. Then we need to multiply that by the probability of leaving the interference state, giving us a probability for each scenario (the different values of n). The average time spent in interference state is the sum, from all n's, of the probability of each scenario times the time spent in each scenario, which is (with the times from exercise c)) t3 (the time to leave the interference state) plus the time to go from state 3 to state 4 and back from state 4 to state 3 (t3 + t4) multiplied n times.

The result is 10.5 minutes (10:30,[mm:ss]).

The result was produced with the following Matlab code:

```
%2.f
p3_2 = 20/(20+5); %p probability of going from state 3 to state 2 (exiting interference state)
p3_4 = 5/(20+5); %p of goinf from state 3 to state 4 (not exiting interference state)
p4_3 = 2/2; %p of going from state 4 to state 3 (also not exiting interference state)
avgTime=0;
for i=0:9
    avgTime=avgTime + (p3_4*p4_3)^i * p3_2*(t3+(t3+t4)*i);
end
```

The average time is not multiplied by 60, because it already comes in minutes, since the times used (t3 and t4) are already in minutes from before, in exercise c).