Vehicule intelligent et communicant

TP sur l'étude de performance des communications véhiculaires

Maxime Ferreira da Costa

Lucas Tramonte
Vitor Opsfelder Estanislau
Max Lee

Octobre 2023

Table des matières

1 Introduction

1.1 Partie 1 : Établissement du Bilan de Liaison

Nous disposons des valeurs suivantes, comme indiqué dans le document :

TABLE 1 – Informations données

$Parametre$	$Valeur$	$Significat$
W	$10MHz$	Largeur de bande des canaux
P_t	$30dBm$	Puissance transmise
A	41	Parametre du modele Winner II : $A + B * log_{10}(d)$
B	22.7	Parametre du modele Winner II : $A + B * log_{10}(d)$
s	$5dB$	Marge de shadowing
G	$0dB$	Largeur de bande des canaux
$SINR_{\min}$	$30dB$	rapport signal sur bruit minimun

Ces valeurs sont essentielles pour notre étude et serviront de paramètres dans nos calculs et nos analyses. Nous pouvons donc retrouver d, la distance maximale entre un point d'accès et un véhicule, en établissant un bilan de liaison :

$$N = -174 + 10log_{10}(W) = -104dBm \tag{1}$$

$$P_r = P_t + G - A - Blog_{10}(d) - s \tag{2}$$

$$SNR = P_r - N > SINR_{min} \tag{3}$$

$$d < 10^{P_t+G-A-s-N-SINR_{min}/b} = 358.97m \tag{4}$$

De cette manière, la distance D maximale entre les points d'accès pour que tous les véhicules soient couverts peut être déterminée :

$$D = 2 \cdot d = 717,94m \tag{5}$$

La raison pour laquelle il y a un facteur 2 dans le calcul est géométrique. Car à la fin d'une zone de détection d'un point d'accès, l'autre sera déjà en train de capter des signaux, puisque la voiture sera dans sa zone

1.2 Partie 2 : Étude de performance des méthodes d'accès ALOHA et CSMA/CA

D'abord, en utilisant le méthode ALOHA pour calculer le taux de perte de paquets en fonction de la densité n de véhicules, on doit calculer la probabilité de succès de n'avoir aucune arrivée de paquets durant une fenêtre de $2T_f$:

$$P_{\text{success}} = \exp(-2 \cdot (N-1) \cdot \lambda \cdot T_f) \tag{6}$$

Par rapport le méthode CSMA/CA, on a besoin de calculer la probalilité Π_{in}, pour qu'on puisse calculer la probabilité de chevauchement. On a utilisé un méthode iteratif pour trouver la probablité :

$$\Pi_{\text{in}} = \left(1 + q \cdot \frac{1 - p_c^m}{1 - p_c} \cdot (1 + \frac{W_0 - 1}{2 \cdot (1 - p_c)})\right)^{-1} \tag{7}$$

$$p_c = 1 - (1 - q \cdot \Pi_{\text{in}} \cdot \frac{1 - p_c^m}{1 - p_c})^{N-1} \tag{8}$$

Le critère de arrêt était défini par :

$$\xi = |p_{c,n+1} - p_{c,n}| < 0.001 \tag{9}$$

Le nombre 'm' de rétransmissions adoptés était m = 2,3 et 5 pour étudier leur perfomance.. De cette façon, on peut tracer la probabilité de chevauchement (entre 2 et 50 voitures) par nombre de vehicules, comme le montre la Figure 1 :

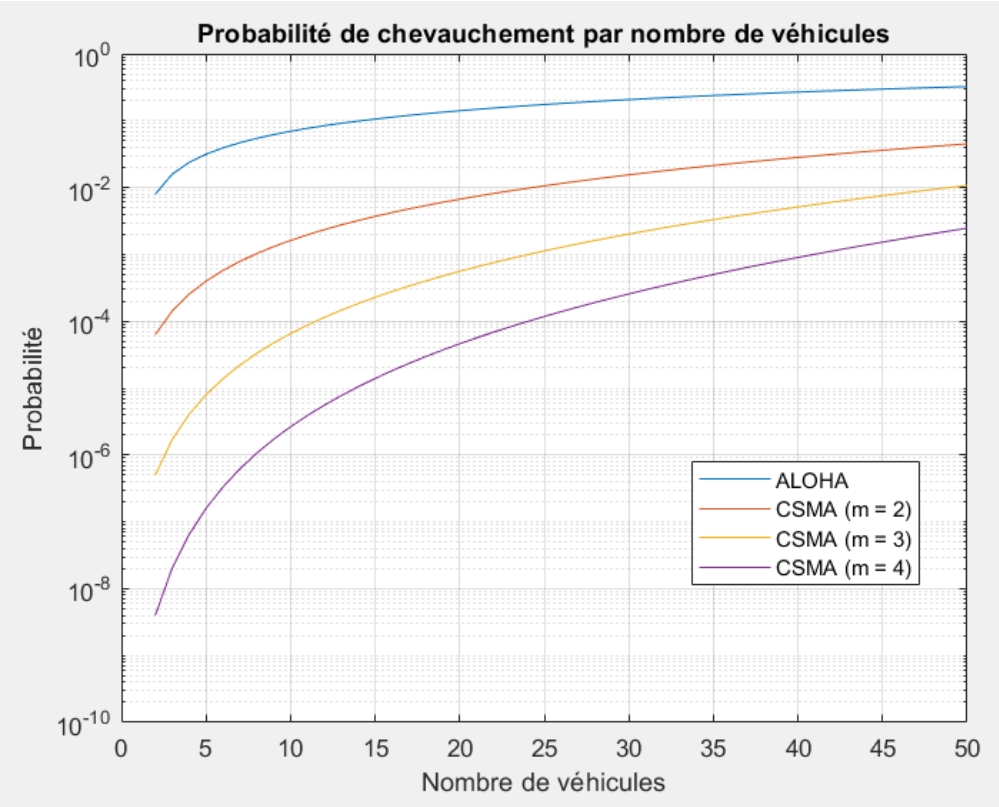

FIGURE 1 – ALOHA vs CSMA performances

Selon la figure, on voit que le méthode CSMA aura toujours un taux de perte inférieur à ALOHA. Ainsi que l'augmentation du chifre de retransmissions mène a une diminuition du taux de perte.

1.3 Partie 3 : Mise en place du scénario avec Automated Driving Toolbox et Interfaçage avec MATLAB

Dans cette partie, on a simulé en utilisant *Matlab Toolbox Automated Driving*, qui nous a permi de collecter des données en temps réels d'une route avec voitures autonomes. La vitesse des voitures était fixé par 30 km/h. La simulation a été faite avec 24 voitures distribués sur la route de façon aléatoire, et en cherchant une simulation plus réaliste, on a choisi ce nombre de voiture car notre route modelée a environ 400 m à une échelle 1 :10, alors, la route réelee aura 4 km, et les réquisites de functionnement étaient que la densité de la route était 3 véhicules/voie/km. Notre route a 2 voies, cela nous ammène à 24 voitures.

Alors, on a crée une function à *Matlab* pour déterminer les postions de chaque voiture par rapport au point d'accès. Cette distance a été calculé par la norme de coordonnées en 2D.

De plus, la function retourne une estimative du nombres de voitures détectées chaque *time-step*, et avec cette information on a pu calculer la taux de perte da la communication. L'estimation a été faite par le rapport entre le nombre de paquets détectés et ceux qui sont envoyés.

On denote :

$$D[k] = \frac{\text{nombre de paquets détectés}}{\text{nombre de paquets envoyés}}$$

Le taux de perte est donné par le complementaire :

$$P[k] = 1 - D[k]$$

Selon le protocole utilisé, on a m = 2 retransmissions, ça permet dire que :

$$P_{total} = (P[k])^2$$

Avec les données, on a tracé le taux de perte a chaque moment :

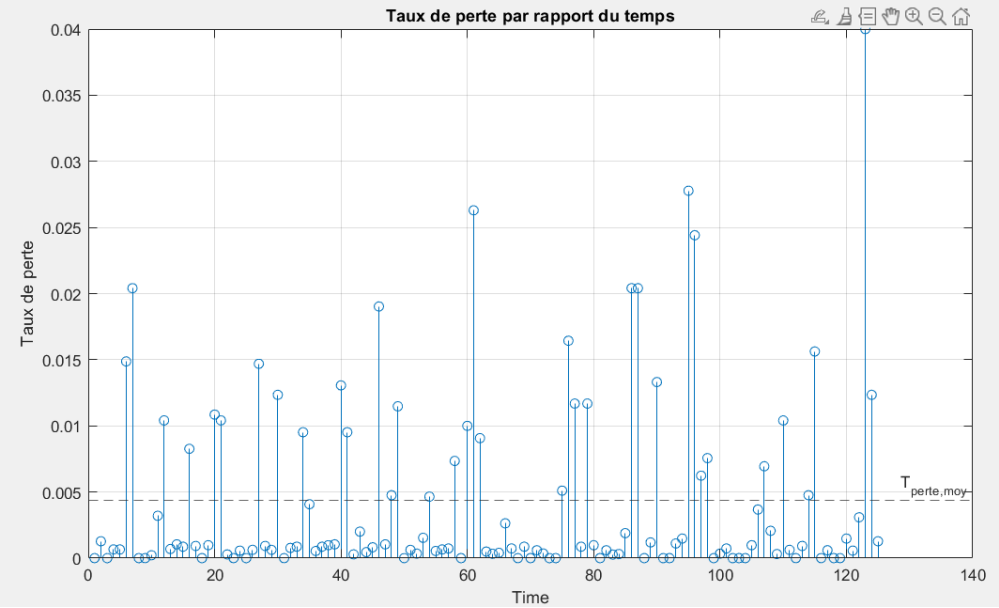

FIGURE 2 – Taux de perte vs temps

Selon le modèle pour le CSMA-CA avec 2 rétransmissions, le taux de perte attendu est 0.001, et avec les données experimentales, on a un taux moyenne de 0.0044, mais la plupart des taux de perte observés sont inférieures à 0.001, donc le résultat est cohérent avec l'ordre de grandeur attendu, et le modèle peut être validé

En suivant, les codes utilisés pour calculer le paramètres :

Deuxième Partie :

```
% TP - Communications
% Code filled by the following students:
% Lucas TRAMONTE
% Vitor OPSFELDER ESTANISLAU
% Max LEE

% TP - Communications
% On a les informations suivantes:

Pt = 30;
W = 10*10^6;
s = 5;
SNR_min = 30;
A = 41; B = 22.7;
```

```
% Le bruit peut  tre  calcul  en utilisant l'expression:

N = -174 + 10*log10(W)

% La distance maximale est donn e par:

d = 10^((Pt-A-s-N-SNR_min)/B)

% Pour  viter  une d connexion, il faut que le v hicule soit entre 2 point
% d'acc s, donc, la distance maximale D sera 2*d.

D = 2*d;


%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% ALOHA:

% On suppose que la transmission ALOHA peut  tre  model e comme un processus
% de Poisson. La transmission est faite pour 1 paquet   chaque 100 ms,
% alors, le param tre <<lambda>> sera:

lambda = 1/(100*10^-3);

% Il faut calculer le param tre T_f, c'est   ffectivement   le temps de
% transmission de chaque paquet. D'abord, on sait que le paquet a une
% taille de 500 octets (500*8 = 4000 bits), et on a une   fficacit   spectrale
% de 1 bit/s/Hz, avec une bande de transmission de 10 MHz, alors, le temps
% de transmission sera: 4000 bits    tre  transmis, dans 1s, 10x10^6 bits
% sont transmis. Par proportion:

Tf = (500*8)/(10^7);

% La probabilit  que les paquets ne se chevauche pas est donn e par:
% P_success = exp(-2*(N-1)*lambda*Tf)
% Alors, la probabilit  de chevauchement sera le compl ment:

% La grille de v hicules sera de 2 jusqu'   50 v hicules pour faire une
% graphique:

N = 2:1:50;
P = 1 - exp(-2*(N-1)*lambda*Tf);
semilogy(N,P)
title('Probabilit  de chevauchement par nombre de v hicules (ALOHA)')
xlabel('Nombre de v hicules')
```

```
ylabel('Probabilit ')
grid on;

% variables:
% CSMA

W = 5;
q = 1 - exp(-lambda*Tf);

tol = 1e-4; % critere de arret du methode iteratif
dif = 10;
pc = 0;
it = 0;
pc_calc1 = compute_CSMA(2,tol,q,W);
pc_calc2 = compute_CSMA(3,tol,q,W);
pc_calc3 = compute_CSMA(4,tol,q,W)

% Comparaison entre ALOHA et CSMA
figure()
semilogy(N,P,N,pc_calc1.^2, N,pc_calc2.^3,N,pc_calc3.^4)
legend('ALOHA','CSMA (m = 2)', 'CSMA (m = 3)','CSMA (m = 4)')
title('Probabilit de chevauchement par nombre de v hicules')
xlabel('Nombre de v hicules')
ylabel('Probabilit ')
grid on;

function[pc_calc] = compute_CSMA(m,tol,q,W)
pc_calc = [];
pc = 0;
it = 0;
for k = 2:1:50
dif = 10;
while (abs(dif) > tol)
    it;
    pc_ant = pc;
    pc = 1-(1- q*(1-pc_ant^m)/(1-pc_ant)*(1/(1+q*(1-pc_ant^m)/
    (1-pc_ant)*(1+ (W-1)/(2*(1-pc_ant))))))^(k);
    dif = pc - pc_ant;
    it = it +1;
end
pc_calc = [pc_calc pc];
end
end
```

Troisième Partie :

```
% Code filled by the following students:
% Lucas TRAMONTE
% Vitor OPSFELDER ESTANISLAU
% Max LEE

% TP - Communication V2X


function [counter,relative_position_AP] = vehicle_position(strData)
% Input : 'exported_file.mat' file (struct generated by Driving Scenario
% Designer after simulation -> Export -> Export Sensor Data)
% Output : counter = counter array the number of vehicles in the range of
% the AP indexed by time
% relative_position_AP = relative distance from each vehicle from the AP
% indexed by time

%initialization of scenario
data_in = load(strData); %loads the scenario file into the variable data
fild_name=fieldnames(data_in);
simulation = data_in.(fild_name{1}); %simulation variables indexed by time

%paramters initialization
K = length(simulation); %number of simulation points
position_RSU = simulation(1).ActorPoses(1).Position; %take the fixed
                                                     % position of the AP

relative_position_AP = {};
counter = [];
SNR_permis = 30;
com_success = [];
for i = 1 : K

    %Compute the relative distance from each vehicle from the AP
    distances = [];

 for n = 2:length(simulation(1).ActorPoses) % the first column is the
                                            % data of the AP, so our loop
                                            % starts for the 2nd column

        % We calculate the norm in order to have the distance of each car
        % to the AP (in 2D - plane)

        dist = norm(position_RSU-simulation(i).ActorPoses(n).Position,2);
        distances = [distances dist];
```

```
 end
 SNR = [];
 for p = 1:length(simulation(i).ObjectDetections)
 SNR_mesure = simulation(i).ObjectDetections{p,1}.ObjectAttributes{1,1}.SNR;
 SNR = [SNR SNR_mesure];
 end

  relative_position_AP = [relative_position_AP {distances}]; % adding the
                                                             %calculated
                                                             %distances
  dist_AP = 70; % We define the range to be 70 m (same of the sensor range)
                                                        %in the toolbox

  %Compute the number of vehicles in the range of the AP
  % using the method find to count the numbers of cars inside a range
  % in a step-time
  cars_inside = length(find(distances < dist_AP));
  counter = [counter cars_inside];

  % Compute the communications that were not well received, filtering by
  % the SNR minimum of 30 dB.
  comm = length(find(SNR > SNR_permis));
  com_success = [com_success comm];
  % Create a vector with the ratio D[k]
  com_success(i) = com_success(i)/(length(simulation(i).ObjectDetections))


end

% plot the graph to show the loss rate each step-time
stem(1:1:K,(1-com_success).^2)
title('Taux de perte par rapport du temps')
xlabel('Time')
ylabel('Taux de perte')
% compute the average loss rate
taux_perte = mean((1.-com_success).^2)
yline(taux_perte,'--',"T_{perte}_{,moy}")
grid on
end
```