

Unité élémentaire : Application Réseau Mobile

Belhadj Walid

Master 1 SICOM

Rapport Tp 2:

Dossier de simulation : Vienna_LTEA_SLS_v2.0_Q3_2018.

Plateforme : Matlab, v 2015.

Partie I :

Type des canaux radio dans le simulateur et localisez leur répertoire :

Canal montant - UpLinkChannelModel

Canal descendant : DownLinkChannelModel

Répertoire : channel_models

Paramètres associés à chaque type de canal radio :

DownLinkChannelModel :

shadow_fading_model : quel model utilise (fast fading ...)

macroscopic_pathloss_model

fast_fading_model

eNodeBs : nombre de stations de bases

attached_UE : utilisateurs attachés.

attached_eNodeB stations de base attachées.

eNodeB_id identifiant de la station de base

pos position

pathloss

shadow_map_id

interfering_eNodeB_ids

the_RB_grid

UpLinkChannelModel :

% Feedback buffer. Implements the delay as a circular buffer of length N

feedback_buffer

feedback_delay

attached_UE utilisateurs attachés.

Définition de shadowing

Le shadowing est l'effet que la puissance du signal reçu fluctue en raison d'objets obstruant

le chemin de propagation entre l'émetteur et le récepteur. Ces fluctuations sont ressenties sur les puissances moyennes locales, c'est-à-dire les moyennes à court terme pour éliminer les fluctuations dans le **multipath**.

Calcul de shadowing :

Calcul du shadowing local par secteur d'une station de base.

```
function [pathloss, is_set] = macroscopic_pathloss(obj)
    % Get eNodeB id
    attached_eNodeB = obj.attached_UE.attached_eNodeB;
    eNodeB_id      = attached_eNodeB.eNodeB_id;
    pos            = obj.attached_UE.pos;

    % Add RRHs (if applicable)
    if ~isempty(attached_eNodeB.RRHs)
        eNodeB_id = [eNodeB_id [attached_eNodeB.RRHs.id]];
    end

    % Now get the pathloss
    pathloss =
reshape(obj.macroscopic_pathloss_model.get_pathloss_eNodeB(pos,eNodeB_id),1,[]);
    is_set = true;
end

% Returns the macroscopic pathloss in dB between the given user's
% position and a given eNodeB.
function [pathloss, is_set] = interfering_macroscopic_pathloss(obj,eNodeB_id)
    % Get eNodeB id
    attached_eNodeB = obj.eNodeBs(eNodeB_id);
    eNodeB_id      = attached_eNodeB.eNodeB_id;
    pos            = obj.attached_UE.pos;

    % Add RRHs (if applicable)
    if ~isempty(attached_eNodeB.RRHs)
        eNodeB_id = [eNodeB_id [attached_eNodeB.RRHs.id]];
    end

    % Now get the pathloss
    pathloss =
reshape(obj.macroscopic_pathloss_model.get_pathloss_eNodeB(pos,eNodeB_id),1,[]);
    is_set = true;
end
```

Calcul du shadowing d'une toute station de base par rapport aux positions, la variable `is_set` est mis à vrai :

Définition de fast-fading

Il se produit principalement en raison des réflexions des surfaces et du mouvement de l'émetteur ou du récepteur. Un « doppler spread » élevé est observé dans les le fast fading rapides avec une Bandwidth Doppler comparable ou supérieure à la bande passante du signal et les variations de canal sont aussi rapides ou plus rapides que les variations du signal. Il provoque des distorsions linéaires dans la forme du signal en bande de base et crée des interférences entre symboles. Une façon de supprimer ISI est l'égalisation adaptative.

Calcule de Fast Fading :

```
% Set a shadow fading model
function set_fast_fading_model(obj,fast_fading_model)
    obj.fast_fading_model    = fast_fading_model;
    %obj.shadow_fading_model_is_set = true;
end
```

Partie II : Evènements réalisés dans un TTI : calcul du SINR du canal

Q1 : Sur un TTI, un utilisateur peut envoyer ou recevoir des données, mais le nombre d'utilisateurs actifs peut être plus élevé puisque un utilisateur peut nécessiter des ressources à un TTI mais pas au(x) TTI(s) suivant(s).

En regardant le fichier **LTE_sim_main**, on initialise l'horloge, on incrémente l'horloge, et si on fait bouger les utilisateurs (**keep_still = true**), on calcule le SINR et le CQI feedback , par ailleurs, les utilisateurs envoient des feedbacks vers les nœuds chaque user aura les ressources nécessaires, une évaluation et une estimation à chaque 10 ms

Q2 : feedback_channel_delay

Temps de réponse du canal pendant une communication dans le canal upLink.

La valeur par défaut : 3 pour, type de simulation « tri_sector_tilted » :

On la mise à zéro :

```
LTE_config.feedback_channel_delay = 0 ;
```

Méthode link_quality_model :

Elle nous permet de calculer le SINR du récepteur qui est la métrique afin de calculer la qualité de la liaison.

Méthode link_performance_model :

Elle permet d'évaluer si les données du Transport Block ont été bien transmises ce qui est la métrique qui évalue la bonne performance du canal.

Définition de SINR :

La quantité utilisée qui résume bien la qualité du lien entre une station de base et un terminal mobile.

C'est le rapport signal bruit interférence, et Calcul de SINR_linear

SINR_linear = RX_power ./ Interference_plus_noise_power;

SINR_dB = 10*log10(SINR_linear);

Diagramme de dépendance :

SINR_Db → SINR_linear → Puissance émise → Bruit + interférence

Rx_total :

La puissance émise par l'antenne, elle est calculée en ajoutant à la puissance (Puissance des données, puissance des signalements) et le Fading (Pathloss + Shadowing + Fast Fading).

L'allocation des puissances par le nœud aux users est **homogène**

TX_power_data = the_RB_grid.power_allocation';

TX_power_signaling = the_RB_grid.power_allocation_signaling';

```
RX_total_RB =  
(TX_power_data+TX_power_signaling)./user_macroscopic_pathloss_lin./user_shadow_fading_  
oss_lin;
```

```
RX_total = reshape([RX_total_RB; RX_total_RB],1,[])/(2);
```

Partie III :

R1 : Débogage de la politique Round robin et BestCQI :

Round robin :

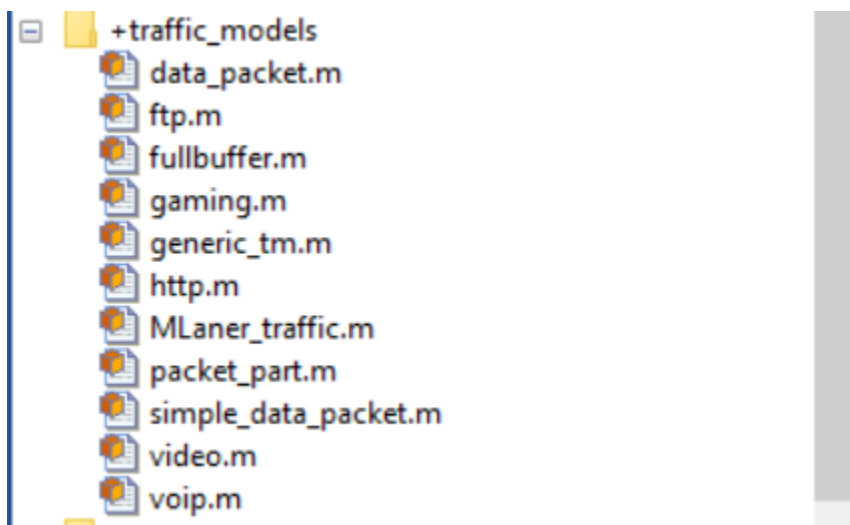
Cet algorithme attribue des **RessourcesBlock** (ressource physique) à chaque utilisateur. En déroulant,

BestCQI :

L'attribution des ressources physiques à chaque utilisateur présentant les meilleures conditions du canal d'une manière identique

R2 :

En développant le dossier traffic_models on peut voir les types de trafic utilisés par le simulateur : dans la capture ci-dessous, on trouve les différentes classes nécessaires pour simuler des comms de type ftp, voie, video.. etc.



à travers la variable **aPrioriPdf**. Dans la fonction **LTE_trafficmodel** permet d'attribuer une priorité à un trafic donné dont les exemples :

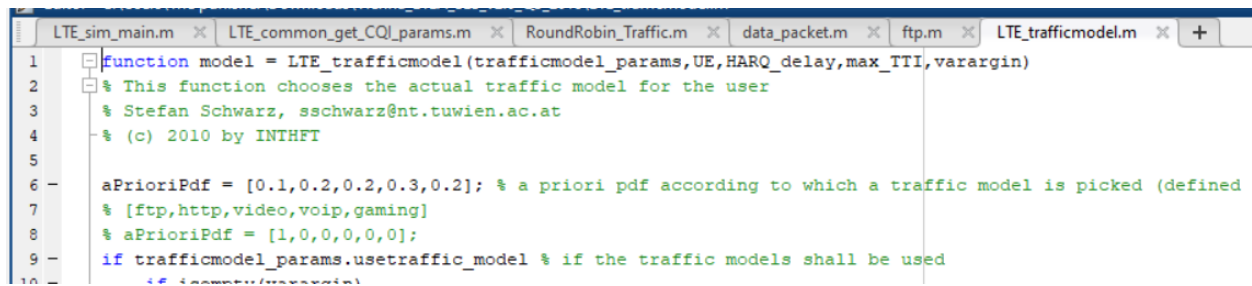
10 % des cas on va avoir du ftp

20 % des cas on va avoir du http

20 % des cas on va avoir du trafic video

30 % des cas on va avoir du trafic en voip

20 % des cas on va avoir du trafic gaming

A screenshot of a MATLAB script editor window. The title bar shows several open files: LTE_sim_main.m, LTE_common_get_CQI_params.m, RoundRobin_Traffic.m, data_packet.m, ftp.m, and LTE_trafficmodel.m. The main window displays the code for the LTE_trafficmodel function. The code includes comments in French and English, and defines the aPrioriPdf vector as [0.1, 0.2, 0.2, 0.3, 0.2] for traffic models ftp, http, video, voip, and gaming respectively. The function is designed to be called from LTE_sim_main.m.

```
1 function model = LTE_trafficmodel(trafficmodel_params,UE,HARQ_delay,max_TTI,varargin)
2 % This function chooses the actual traffic model for the user
3 % Stefan Schwarz, sschwarz@nt.tuwien.ac.at
4 % (c) 2010 by INTHTF
5
6 aPrioriPdf = [0.1,0.2,0.2,0.3,0.2]; % a priori pdf according to which a traffic model is picked (defined
7 % [ftp,http,video,voip,gaming]
8 % aPrioriPdf = [1,0,0,0,0];
9 if trafficmodel_params.usetraffic_model % if the traffic models shall be used
10 % if isempty(model)
```

R4 : Simulation qu'utilise tous les types de trafic :

Il faut voir avec le type de scheduler

```
%% Traffic models
if isfield(LTE_config,'traffic_models')
    if LTE_config.traffic_models.usetraffic_model &&
~(strcmp(LTE_config.scheduler,'constrained') ||
strcmp(LTE_config.scheduler,'alpha fair') ||
strcmp(LTE_config.scheduler,'prop fair traffic') ||
strcmp(LTE_config.scheduler,'round robin traffic') ||
strcmp(LTE_config.scheduler,'CarScheduler'))
```


Puis on manipule les variables :

```
LTE_config.traffic_models.usetraffic_model = true;
```

Et juste après le débogage on peut voir le trafic de différents utilisateurs :

Nombre d'utilisateurs de chaque type de trafic : 1 §

R5 : Comparaison des deux débits de simulation :

 LTE_GUI_show_aggregate_results

