



Telecom Bretagne
655 Avenue du Technopole
29200 Plouzané

Promotion 2015 / 2018



Rapport de simulation d'un système de transmission – SIT 213

Auteurs	Marc LEMAUVIEL – Johann CORCUFF-REBECCHI – Arouna KANE – Lim Kévin CHAO – Lahoucine AHMOUCHE
E-Mails	marc.lemauviel@telecom-bretagne.eu johann.corcuff-rebecchi@telecom-bretagne.eu arouna.kane@telecom-bretagne.eu lim.chao@telecom-bretagne.eu lahoucine.ahmouche@telecom-bretagne.eu
Destinataire	Éric COUSIN, Bruno FRACASSO, Julien MALLET, François-Xavier SOCHELEAU
E-Mail	eric.cousin@telecom-bretagne.eu bruno.fracasso@telecom-bretagne.eu julien.mallet@telecom-bretagne.eu fx.socheleau@telecom-bretagne.eu
Formation suivie	Ingénieur spécialisé en Informatique, Réseaux et Télécommunications
Établissement	Télécom Bretagne
Promotion	2018

Date	Version	Modifié par	Motifs
13/09/2016	1	LEMAUVIEL Marc	Création du document
06/10/2016	2	LEMAUVIEL Marc	MAJ – Organisation travail d'équipe
17/10/2016	3	LEMAUVIEL Marc	MAJ – Etape 2

Rapport de simulation d'un système de transmission – SIT 213

Étudiants : Marc LEMAUVIEL, Johann CORCUFF-REBECCHI, Arouna KANE, Lim Kévin CHAO, Lahoucine AHMOUCHE étudiants en Formation d'Ingénieur en Partenariat

Promotion : 2018

Établissement : Télécom Bretagne

Destinataires : Éric COUSIN, enseignant chercheur au département INFO

Bruno FRACASSO, enseignant chercheur au département OPTIQUE

Julien MALLET, enseignant chercheur au département INFO

François-Xavier SOCHELEAU, enseignant chercheur au département SIGNAL & COM

TABLE DES MATIERES

TABLE DES MATIERES.....4

INTRODUCTION.....5

ORGANISATION DU TRAVAIL D'EQUIPE.....7

INTRODUCTION

L'objectif de ce projet est de nous faire étudier une chaîne de transmission (représentée par la figure 1).

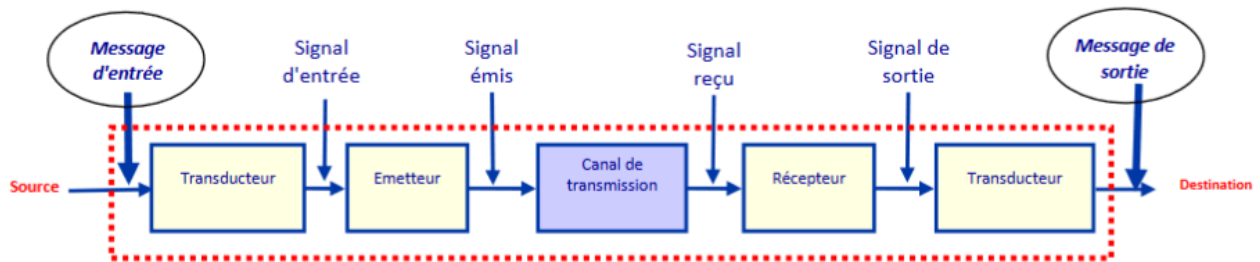


Figure 1 : Représentation d'une chaîne de transmission

Pour effectuer cette étude, nous sommes composés d'un groupe de 5 personnes. L'avancée du projet se fait au travers de 6 étapes qui permettront de simuler pas à pas les différents blocs de la chaîne de transmission. L'étape 1 s'est déroulée entre le 13/09 et le 06/10. Elle consistait à réaliser une transmission élémentaire « back-to-back ».

L'étape 2 a été effectuée entre le 07/10 et le 17/10. Son but était d'effectuer une transmission non bruitée d'un signal analogique.

Rédaction en cours, à compléter au fur et à mesure des séances.

ETAPE 1: TRANSMISSION « BACK-TO-BACK »

L'application réalisée pour l'étape 1 correspond à la figure 2 :

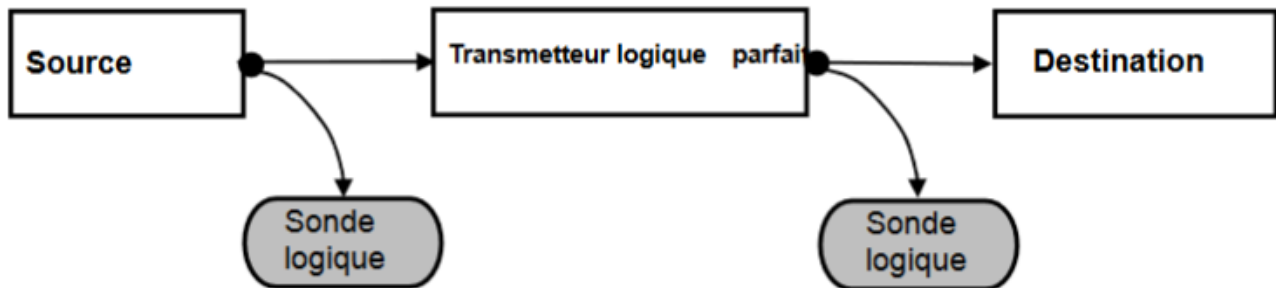


Figure 2 : Modélisation de la chaîne de transmission à l'étape 1

La source émet une séquence booléenne fixée ou aléatoire. Le transmetteur logique parfait se contente, à la réception d'un signal, de l'émettre tel quel vers les destinations qui lui sont connectées. La destination se contente de recevoir le signal du composant sur lequel elle est connectée. Des sondes logiques permettent de visualiser les signaux émis par la source et le transmetteur parfait. L'application principale calcule le taux d'erreur binaire (TEB) du système.

Le résultat de l'étape 1 est disponible sous forme du livrable 1 dans lequel figure les éléments suivants :

- Un fichier readme.txt qui donne la marche à suivre pour exécuter les différents programmes,
- 3 fichiers de scripts bash permettant d'exécuter les fichiers de code java,
- Les fichiers de code java réalisant les fonctionnalités du cahier des charges.

Afin de vérifier le fonctionnement de l'étape 1, nous avons réalisés des tests. Ces derniers nous ont permis de mettre en évidence la conformité du travail effectué par rapport au cahier des charges. Ainsi, pour l'ensemble des transmissions simulée, nous obtenons un TEB de 0 (le transmetteur logique étant parfait et comme il n'y a aucune source d'erreur, comme du bruit par exemple, le résultat est cohérent).

Cette étape a été réalisé séparément par les 2 groupes (Johann et Marc d'un côté, Lim Kevin, Arouna et Lahoucine de l'autre). C'est pour cette raison que nous ne détaillerons pas plus la réalisation de cette étape puisque le livrable a déjà été rendu par chacun des 2 groupes.

A la suite de cette étape, les 2 groupes ont été fusionnés et une organisation de travail a été mise en place pour la suite du projet. Cette organisation est décrite dans la partie suivante.

ORGANISATION DU TRAVAIL D'EQUIPE

L'organisation est un point essentiel dans ce projet. Nous sommes un groupe de 5 et nous devons donc nous répartir les rôles pour que chacun soit efficace. Pour respecter au mieux les délais et les tâches nous avons réparti des responsabilités pour ce projet :

Chef de projet - Responsable livrables : Marc Lemauiel

Responsable développement : Johann Corcuff

Responsable télécom : Lahoucine Ahmouche

Responsable validation : Arouna Kane

Responsable qualité : Lim Kévin Chao

L'objectif principal est que chacun d'entre nous acquière des compétences dans les différents domaines du projet (Info et Telecom). Pour cela nous ferons régulièrement des séances de travail en commun pour que chacun puisse comprendre les différentes parties du projet. Nous réaliserons aussi des documentations (notamment de la javadoc) tout au long du projet pour les différentes étapes du projet.

Afin de garantir un bon niveau de qualité de notre travail et notamment pour notre code, nous appliquerons des tests pour nos itérations. La réalisation de ces derniers se fera au travers de JUnit qui est un outil simple à maîtriser et qui nous permettra d'avancer rapidement sur nos tests. Nous mettrons également un point d'honneur sur la structure et la forme du code afin d'en faciliter la lecture autant que possible.

La rédaction du rapport sera la responsabilité d'une seule personne (le chef de projet) afin de garder la même ligne de conduite.

Le travail étant réalisé en groupe, la mise en commun du code est essentielle. Nous avons choisi d'utiliser un gestionnaire github pour mettre en commun nos différents travaux. Cela nous permettra de travailler en simultané sur des parties de code différentes.

BE – TEST ET VALIDATION

Les tests sont réalisés pour valider le contrat public des différentes classes et s'assurer du bon fonctionnement du programme au regard du cahier des charges conçu en amont du projet.

Les tests ont différents objectifs :

- Valider les fonctionnalités définies dans le cahier des charges établi par le client,
- Détecter de potentiels bugs et les corriger,
- S'assurer de la qualité du programme développé,

Quelles parties peut-on vraiment tester avec un programme de test ?

Les parties que l'on peut vraiment tester avec un programme de tests sont :

- Les contrats publics que sont censés remplir les différentes classes (tests unitaires),
- Le bon fonctionnement global du programme (tests d'intégrations).

Comment améliorer la testabilité ?

L'idéal est de coder les tests avant l'implémentation des différentes classes. Cela nécessite d'avoir une architecture où les contrats publics des différentes classes sont déjà définis. Dans la réalité, il se peut que l'architecture évolue ce qui peut amener à modifier les tests précédemment réalisés.

Comment faire du test de non régression à chaque nouvelle itération du projet ?

Pour cela il faut tester les parties du code les plus stables ainsi que les tests sur le fonctionnement général du programme qui ne sont pas censés changer au cours des itérations. Si du code mutable est mélangé avec du code stable il est bon de mettre une couche d'abstraction entre les deux. Pour cela on peut utiliser une interface, une classe abstraite ou des design patterns comme le pattern delegation.

Quelle confiance accorder aux résultats de simulation, et comment accroître cette confiance ?

La confiance accordée aux résultats de simulation dépend de la manière dont ils ont été réalisés. Si c'est le même développeur qui code à la fois les tests et la fonctionnalité correspondante, on pourra alors lui accorder une confiance moindre que si ces deux parties avaient été réalisées par deux personnes différentes (regards croisés). C'est pour cela que les entreprises possèdent des personnes spécialisées dans les tests et validation des programmes qui sont indépendantes des équipes de développement afin d'accroître cette confiance dans les résultats.

Comment comparer des simulations lorsqu'elles reposent sur des comportements aléatoires (message, bruit, ...) ?

Pour tester les simulations malgré des comportements aléatoires, on peut utiliser des outils statistiques permettant de mesurer un degré de corrélation suffisant entre une entrée et la sortie attendue (calcul du Taux d'Erreur Binaire, TEB). La prédétermination théorique du TEB (en utilisant MATLAB par exemple) et sa comparaison avec le TEB calculé dans le programme est un bon moyen de vérifier la véracité du code réalisé.

ETAPE 2 : TRANSMISSION ANALOGIQUE (NON BRUITEE)

La seconde étape du projet reprend le travail effectué lors de l'étape 1. Cependant nous y ajoutons un émetteur, un récepteur et changeons le canal de transmission (analogique) comme indiqué dans la figure 3 :

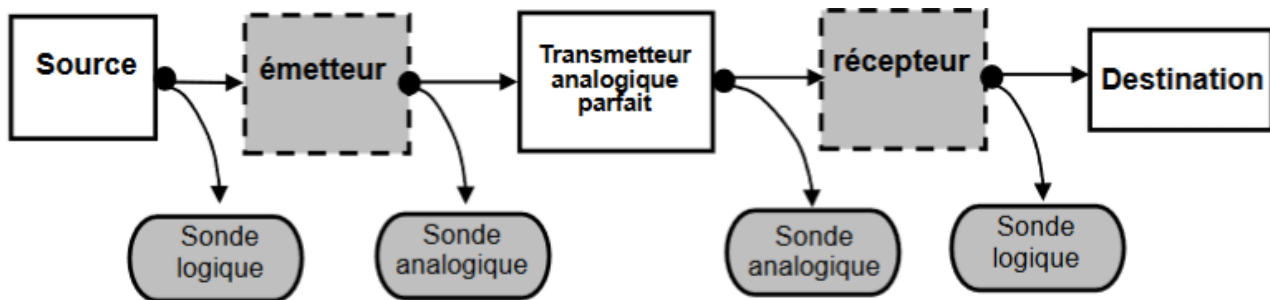


Figure 3 : Modélisation de la chaîne de transmission à l'étape 2

L'ajout de l'émetteur et du récepteur permet de réaliser la conversion « logique => analogique » puis « analogique => logique ». Grâce à cela, le signal peut être émis au travers d'un canal analogique (dans notre cas, c'est le transmetteur analogique parfait).

Cela correspond à ce qu'il se passe dans la réalité. Nous disposons d'un signal numérique (composé de 0 et de 1 logiques) que nous souhaitons transmettre au travers une chaîne de transmission. Pour ce faire, nous convertissons ce message logique en un signal analogique grâce à l'émetteur. L'intérêt de cette pratique est de pouvoir adapter le signal au canal de transmission (hertzien, guidé optique, guidé électrique, ...). Au bout du canal de transmission, le récepteur reçoit le signal analogique puis le convertit en signal numérique (le but étant que le message en sortie soit identique à celui en entrée).

La transmission de l'information se fait au travers de l'utilisation de l'un de ces 3 signaux dont les caractéristiques sont décrites ci-après :

- NRZ : forme d'onde rectangulaire
- NRZT : forme d'onde trapézoïdale (temps de montée ou de descente à 1/3 du temps bit)
- RZ : forme d'onde impulsionnelle (amplitude min sur le premier et dernier tiers du temps bit, impulsionnelle sur le tiers central avec un max au milieu du temps bit égal à l'amplitude max)

La réalisation de l'étape 2 s'est déroulée lors de sessions de partages sur notre temps de travail personnel. Afin d'optimiser notre gestion des ressources, nous avons mis en place un dépôt Github. Grâce à cela, chacun de nous pouvait travailler sur le projet indépendamment des autres personnes. Toutefois, nous avons dû gérer l'encodage des fichiers java sous les différents OS (Windows et Linux) ainsi que la présence de fichiers liés aux configurations individuelles de nos IDE.

D'un point de vue implémentation, nous sommes tout d'abord partis sur une première architecture où nous utilisons l'héritage afin de spécialiser les émetteurs et les récepteurs selon le type de signal transmis (NRZ, NRZT ou RZ). Cela nous a amené à avoir un grand nombre de classes et une relative redondance du code.

C'est pourquoi nous avons opté pour une seconde architecture utilisant le pattern delegation. Nous avons ainsi créé le package « codeur » qui contient des encodeurs et des décodeurs pour chacun des types de signaux. Il suffisait ensuite de passer en argument l'encodeur (respectivement le décodeur) désiré à l'émetteur (au récepteur). Ainsi nous n'avions plus qu'une seule classe émetteur (et récepteur) ce qui nous a permis de factoriser le code. Néanmoins, nous avons dû modifier nos tests unitaires afin de les adapter à cette nouvelle architecture.

D'un point de vue validation, nous avons voulu tester si les signaux produits par les encodeurs à partir d'une suite binaire connue correspondaient bien aux exigences du cahier des charges. Nous avons donc analysé si le signal en sortie de l'encodeur puis du décodeur correspondait bien aux attentes.

De plus, nous avons modifié le script bash demo.sh afin de présenter les nouveaux modes de transmissions analogiques du signal conçus lors de cette étape.

Enfin, tout comme pour l'étape 1, nous obtenons un TEB de 0 ce qui correspond à une transmission parfaite. En effet, la chaîne de transmission utilise un canal parfait pour le moment. Lors de l'étape 3, nous modifierons le canal de transmission pour ajouter un bruit blanc additif gaussien ce qui entraînera des variations du TEB.