Guide d'utilisation de l'implémentation du serveur NISQA pour Debian

Auteurs :
Séverin CHEVALIER
Liam COURSODON
Jules DECAESTECKER
Zaccarie KANIT
Clémence MARINIER
Léo STENGEL
Elias TRANCHANT

Décembre 2022

1 Présentation

Ce projet a été réalisé par 7 étudiants d'IMT Atlantique Brest avec Vincent BARRIAC, Yvain JORIGNY et Nicolas PENNANEACH de Orange (Innovations/Networks). NISQA est un algorithme évaluant la qualité d'une communication audio développé par des chercheurs de l'Université Technique de Berlin. Ce projet visait à remplacer l'algorithme PESQ anciennement utilisé. Ce guide d'utilisation détaille l'architecture et le fonctionnement du serveur HTTP pour Debian et complète le *readme* associé à ce projet.

2 Contenu de l'archive

Dans cette archive se trouve plusieurs fichier :

- README.md : description du répertoire et consignes d'utilisation
- NISQA : dossier contenant le package debian. À packager pour créer un fichier .deb qui sera ensuite installé
- requirements.txt : fichier contenant les dépendances python à installer
- install.sh : script bash installant les dépendances et le logiciel debian
- dg105.wav : fichier audio servant aux tests
- test.sh : script bash effectuant un test du bon fonctionnement du logiciel debian en utilisant l'audio dg105.wav

3 Installation

En se plaçant dans le répertoire du projet :

- Pour construire le .deb:

```
dpkg-deb -b nisqa
```

- Pour installer le serveur sans installer les dépendances:

```
sudo dpkg -i nisqa.deb
```

- Pour télécharger et installer les librairies et le logiciel :

```
sudo pip install -r requirements.txt
```

```
sudo apt-get install ./nisqa.deb
```

4 Utilisation

4.1 Lancement du serveur

- Pour lancer le serveur

```
sudo su - nisqa -c "rwd up"
```

- Pour éteindre le serveur

```
sudo su - nisqa -c "rwd down"
```

- Pour vérifier l'état (allumé / éteint) du serveur

```
sudo su - nisqa -c "rwd status"
```

4.2 Lancement d'une mesure

- Pour utiliser avec un fichier .wav

```
curl -H "Content-type: audio/vnd.wave" --data-binary
@- http://127.0.0.1:8080/cgi-bin/nisqa.cgi  < audio.wav
```

- Test avec test.sh

```
./test.sh 127.0.0.1
```

Ce script fait le test avec le fichier dg105.wav et compare le résultat obtenu avec le résultat qu'on obtient normalement si on applique directement le modèle NISQA au fichier audio (écrit à la main dans le script). Comme l'algorithme NISQA est du machine learning, les décimales des résultats peuvent différer selon les machines. C'est pourquoi on ne compare que les 3 premiers chiffres après la virgule.

4.3 Résultats attendus

Le résultat de la commande curl précédente est de la forme :

```
[MOS, Noisiness, Coloration, Discontinuity, Loudness]
```

Chaque note est comprise entre 1 et 5. On peut avoir par exemple :

```
[1.495162,2.407247,2.001792,2.026033,2.895231]
```

5 Fonctionnement et Maintenance

5.1 Cadre d'utilisation du serveur NISQA pour Debian

Ce logiciel Debian vise à être déployé sur une machine virtuelle Debian. Le schéma ci-dessous explique l'interaction des robots sipcan (robots qui envoient les audio sur le réseau et simulent une communication VoIP) avec le serveur NISQA.

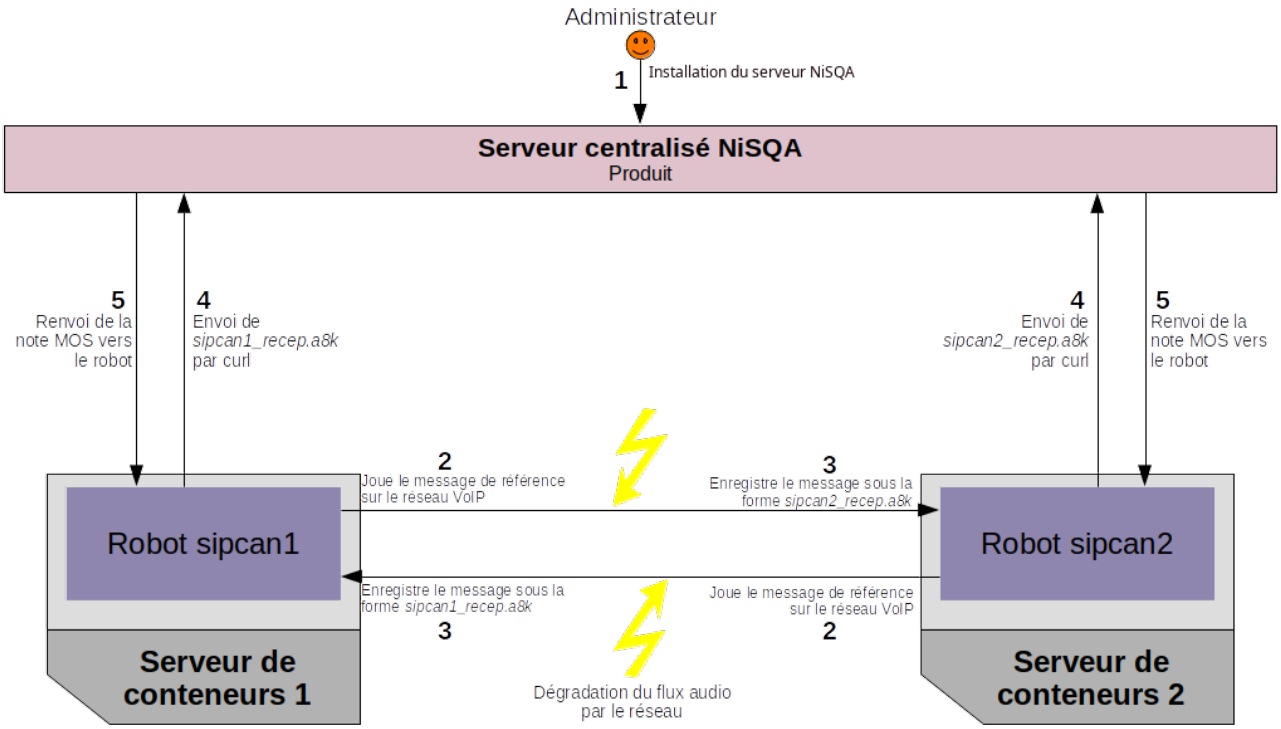

Figure 1: Périmètre de notre produit dans le cas d'un serveur centralisé

5.2 Algorithme NISQA

Le modèle d'évaluation de la qualité audio que nous utilisons est nommé NISQA pour Non-intrusive Speech Quality Assessment. C'est un réseau de neurones convolutif basé donc sur des méthodes de deep-learning codé en python. Il a déjà été entraîné sur différents ensembles de données et est maintenant utilisable tel quel, sans qu'il soit nécessaire de le réentrainer.
Pour l'instant, l'algorithme fonctionne sans fichier de référence. Cela signifie que le fichier audio dégradé après transmission n'est pas comparé avec celui enregistré avant la transmission. Les sources Python sont hébergées sur le dépôt Github de G. Mittag :
https://github.com/gabrielmittag/NISQA

5.3 Architecture du logiciel

Cette partie décrit les différents composants du logiciel Debian pour que par la suite, si une nouvelle version du logiciel NISQA est publié ou si Orange décide de changer d'algorithme d'évaluation, la modification puisse se faire sans problème.

5.3.1 Le logiciel Debian et son installation

Le répertoire NISQA est composé de 2 sous-répertoires :

- DEBIAN : Ce répertoire contient les informations propres à l'installation du logiciel Debian réalisé lors de "dpkg -i nisqa"
- home/nisqa : Lors de l'installation du package Debian, un utilisateur nisqa est créé. Le contenu du répertoire home/nisqa sera copié dans /home/nisqa

Le dossier DEBIAN contient 4 fichiers :

- control : Description des dépendances du logiciel, de son nom, des auteurs, ...
- postint : Script lancé après le dépaquetage du logiciel. C'est lui qui crée l'utilisateur nisqa et lance le serveur
- preinst : Script lancé avant le dépaquetage. Si le logiciel était déjà installé et le serveur allumé, cela éteint le serveur.
- prerm : Si le logiciel est supprimé, ce script est exécuté juste avant. Cela éteint le serveur.

5.3.2 Le serveur

L'architecture du serveur utilisé ici est la reprise de celle de PESQ. La gestion du serveur est géré par plusieurs fichier contenu dans home/nisqa/bin dont rwd (commande pour lancer/éteindre le serveur), rwdog et ezhttpd.
Le fichier home/nisqa/www/cgi-bin/nisqa.cgi est celui qui est appelé lors de la commande curl à "http://127.0.0.1:8080/cgi-bin/nisqa.cgi". C'est ce fichier qui va s'occuper des arguments, appeler le script python de NISQA et envoyer la réponse.

5.3.3 L'évaluation de la qualité audio : NISQA

Comme dit précédemment, l'appel du script python est effectué dans home/nisqa/www/cgi-bin/nisqa.cgi :

```
x="$(python3 bin/nisqa/run_predict.py --mode predict_file --pretrained_model bin/
    nisqa/weights/nisqa.tar --deg $tmp.wav 2>&1)"
```

Dans le dossier home/nisqa/bin se trouve un dossier nisqa qui est le clone du répertoire du github de Gabriel Mittag. C'est ce dossier qu'il faut mettre à jour si l'algorithme vient à évoluer. Comme vu dans la commande ci-dessus, le script run_predict.py est appelé et utilise les poids contenu dans weights/nisqa.tar .
Les erreurs sont redirigées sur la sortie standard de sorte que l'utilisateur du logiciel puisse connaître l'origine de l'erreur.