# Évaluation Pseudo–subjective de la Qualité des Flux VoIP: une Approche par Réseaux de Neurones Aléatoires

Martín Varela[1]

Projet ARMOR, Irisa - INRIA/Rennes
Campus universitaire de Beaulieu
35042 RENNES CEDEX.
mvarela@irisa.fr
http://www.irisa/fr/armor/

**Résumé** :
Dans ce papier nous présentons une nouvelle approche que nous avons développée récemment pour l'évaluation *pseudo–subjective* de la qualité des flux multimédia transmis sur un réseau IP tel qu'Internet. Nous nous intéressons particulièrement aux flux de voix sur IP (VoIP), et nous présentons quelques uns des résultats que nous avons obtenus avec cette technique.
Notre approche est passive, peut se réaliser en temps–réel et donne des résultats qui ont une très bonne corrélation avec ceux obtenus par les méthodes subjectives. Elle est basée sur l'utilisation d'un Réseau de Neurones Aléatoire (RNN), entraîné avec des résultats d'évaluations subjectives pour estimer la qualité d'un flux tel que le ferait un "utilisateur moyen".

## 1 Introduction

Au cours de ces dernières années, l'explosion d'Internet a donnée lieu à une nouvelle génération d'applications multimédia qui l'utilisent comme moyen de diffusion. De telles applications, par exemple la transmission de la vidéo en temps-réel et sur demande, la téléphonie, ou la téléconférence, sont devenues courantes et elles constituent aujourd'hui un domaine de recherche très actif ainsi qu'un nouveau secteur du marché des télécommunications.

Ces applications ont certains besoins vis-à-vis des services du réseau, dues à des contraintes temporelles qui leur sont propres. Or, l'architecture d'Internet n'a pas été conçue pour supporter le temps-réel, ce qui entraîne plusieurs problèmes à résoudre lorsque l'on veut implémenter des applications ayant ces types de contraintes. De ce fait, la qualité de ces applications est fortement dépendante de la capacité, la charge et la topologie du réseau. Plusieurs résultats (Bolot *et al.*, 1999; Mohamed *et al.*, 2004; Hands & Wilkins, 1999; Choi & Constantinides, 1989; Claypool & Tanner, 1999) montrent que pour la VoIP, certains paramètres du réseau tels que le taux de perte, la distribution des pertes, le délai et la gigue influencent la qualité perçue par les utilisateurs. Cependant, la façon dont ces paramètres affectent la qualité reste difficile d'apréhender.

Dans la littérature, on trouve deux types de métriques de qualité : les *évaluations subjectives* (e.g. celle définie dans (ITU-T Recommendation P.800, 1996)), qui sont réalisées par un groupe de personnes dans un environnement contrôlé, et les *évaluations objectives*, en général basées sur des formules et des algorithmes et qui permettent d'obtenir une mesure de la qualité à partir d'un échantillon de voix (ou bien d'audio au sens large).

Étant donnée que la qualité d'un flux de voix est un concept qui reste très subjectif, il n'est pas étonnant que les évaluations subjectives soient celles qui donnent les "vraies valeurs" de la qualité perçue. Le problème de ce type d'approche est que ces évaluations sont très chères à réaliser et aussi très encombrantes, car elles ont besoin d'une logistique assez importante. De plus, elles ne sont pas, par définition, adaptées à des applications qu'ont besoin d'une évaluation automatique de la qualité (par exemple, le contrôle dynamique de la qualité dans un logiciel de téléphonie sur IP). Il est donc intéressant de développer des

techniques d'évaluation objective de la qualité permettant d'éviter les difficultés associées aux méthodes subjectives. Ces méthodes présentent d'autres problèmes, surtout concernant leur performances en ce qui concerne la corrélation des résultats avec ceux des méthodes subjectives lorsque on les utilise dans un contexte de réseau.

La méthode que nous proposons est un compromis entre la qualité des méthodes subjectives et le caractère pratique et à faible coût des méthodes objectives.

Le reste de l'article est organisé de la façon suivante. Dans la Section 2, nous présentons plus en détail les méthodes d'évaluation de qualité que l'on trouve dans la littérature. La Section 3 présente notre approche et ses bases mathématiques. Dans la Section 4 nous comparons la performance de notre approche avec celle des autres méthodes objectives. Enfin, nous présentons nos conclusions dans la Section 5.

# 2 Sur les Méthodes d'Évaluation de la Qualité de la VoIP

## 2.1 Évaluation Subjective

Les mécanismes d'évaluation subjective de la qualité d'un flux multimédia consistent principalement en des expériences de laboratoire, dans lesquelles un certain nombre de sujets sont exposés à des échantillons d'audio ou de vidéo qui ont été dégradés et doivent donner des points à chacun. Après un filtrage statistique, on prend la moyenne des opinions (*mean opinion score - MOS*) comme mesure de la qualité de l'échantillon. Ces mécanismes ont été standardisés (ITU-T Recommendation P.800, 1996) et sont très répandus (Hands & Wilkins, 1999).

Bien que l'évaluation subjective est le mécanisme le plus précis pour déterminer la qualité des flux audio transmis sur un réseau de paquets, elle présente plusieurs inconvénients, à savoir :

- très coûteuse, tant en temps qu'en ressources,
- pas adaptée aux applications qui ont besoin d'une évaluation automatique de la qualité,
- encombrante à réaliser, et
- inadaptée aux applications temps-réel (et donc à la tarification ou au contrôle des flux).

## 2.2 Évaluation objective

Les difficultés associées à l'évaluation subjective de la qualité ont donnée lieu au développement d'autres métriques de qualité qui, bien que moins "performantes", sont beaucoup plus pratiques et moins coûteuses. Parmi les méthodes objectives les plus connues on trouve le Signal-to-Noise Ratio (SNR), le Segmental SNR (SNRseg), le Perceptual Speech Quality Measure (PSQM) (Beerends & Stemerdink, 1994), les Measuring Normalizing Blocks (MNB) (Voran, 1997), l'ITU E–model (ITU-T Recommendation G.107, 2003), l'Enhanced Modified Bark Spectral Distortion (EMBSD) (Yang, 1999), le Perceptual Analysis Measurement System (PAMS) (Rix, 1999) et PSQM+ (Beerends, 1997). Ces méthodes ne donnent pas toujours des bonnes corrélations avec la perception humaine et donc leur utilité comme remplacements des évaluations subjectives est limitée. De plus, à l'exception de l'E-model, toutes ces métriques proposent une façon de comparer l'échantillon reçu avec l'original et ne sont donc pas adaptées aux applications en temps–réel.

Étant donné que la qualité d'un flux VoIP est influencée par beaucoup de paramètres (Hands & Wilkins, 1999; Claypool & Tanner, 1999), il n'est pas facile de concevoir une formule synthétique qui puisse les relier et qui donne une "bonne" évaluation quantitative. Cette complexité nous a mené à développer une méthode alternative pour l'évaluation de la qualité de ces flux.

# 3 Une Approche Pseudo–subjective : l'Évaluation avec des Réseaux de Neurones Aléatoires

La méthode que nous proposons (Mohamed *et al.*, 2004; Mohamed & Rubino, 2002) est un hybride entre l'évaluation subjective et l'évaluation objective, qui peut être appliqué aussi bien à la VoIP, qu'à l'audio *hi-fi* et à la vidéo. L'idée est de faire évaluer subjectivement plusieurs échantillons dégradés et utiliser ensuite ces résultats pour entraîner un RNN, en lui apprenant la relation entre les paramètres à l'origine de la dégradation des échantillons et la qualité perçue.

Pour que cela marche, on doit considérer un ensemble de $P$ paramètres qui peuvent avoir un effet sur la qualité *a priori*. Pour la voix, par exemple, on peut choisir le codec, le taux de perte dans le réseau, le délai moyen de bout en bout, etc. On appellera cet ensemble $\mathcal{P} = \{\pi_1, \ldots, \pi_P\}$. Une fois $\mathcal{P}$ défini, on doit choisir des valeurs représentatifs pour chaque $\pi_i$, avec une valeur minimal $\pi_{\min}$ et une valeur maximal $\pi_{\max}$, selon les conditions sous lesquelles on croit que le système fonctionnera. Soit $\{p_{i1}, \cdots, p_{iH_i}\}$ l'ensemble de ces valeurs possibles du paramètre $\pi_i$, où $\pi_{\min} = p_{i1}$ and $\pi_{\max} = p_{iH_i}$. Le nombre de valeurs à choisir dépend de la taille de l'intervalle choisi et de la précision désirée. Dans ce contexte, on appellera *configuration* à un ensemble $\gamma = \{v_1, \ldots, v_P\}$ où $v_i$ est l'une des valeurs choisies pour $\pi_i$.

Comme le nombre total de configurations peut être très important, il faut choisir un sous-ensemble de configurations à utiliser pour l'évaluation subjective. Ce choix peut être aléatoire, mais il est important d'avoir des valeurs aux extrémités des intervalles considérés pour chaque paramètre. Il est aussi important d'essayer d'avoir plus de points dans la région de l'espace où se trouvent les configurations qui seront les plus courantes dans l'usage quotidien de l'application considérée. Une fois que les configurations à utiliser ont été choisies, il faut générer des échantillons dégradés par la transmission sur le réseau avec chaque configuration considérée. Pour cela, on peut utiliser un simulateur, ou une maquette de réseau.

Plus formellement, on doit choisir un ensemble de $M$ échantillons $(\sigma_m)$, $m = 1, \cdots, M$ d'un type et d'une durée en accord à ce qui est spécifié par exemple dans (ITU-T Recommendation P.800, 1996). On a aussi besoin d'un ensemble de $S$ configurations $\{\gamma_1, \cdots, \gamma_S\}$, où $\gamma_s = (v_{s1}, \cdots, v_{sP})$ et $v_{sp}$ est la valeur du paramètre $\pi_p$ dans la configuration $\gamma_s$. À partir de chaque échantillon $\sigma_i$ on construit un ensemble $\{\sigma_{i1}, \cdots, \sigma_{iS}\}$ d'échantillons qui ont été transmis sur le réseau sous des conditions variées. C'est à dire, l'échantillon $\sigma_{is}$ est la séquence qui a été reçue lorsque l'émetteur à envoyé $\sigma_i$ à travers le système source–réseau où les $P$ paramètres considérés ont les valeurs correspondantes à la configuration $\gamma_s$.

Une fois que les échantillons ont été générés, il faut réaliser une évaluation subjective, pour associer une valeur de qualité à chaque échantillon dégradé (et donc à chaque configuration). Après un filtrage statistique des résultats [1], la séquence $\sigma_{is}$ reçoit une note $\mu_{is}$ (souvent cette note est un MOS). Une fois que l'on a une valeur de qualité associée à chaque configuration, on peut entraîner le RNN qui pourra par la suite donner des estimations de qualité pour des flux semblables (dans le sens où les paramètres considérés aient des valeurs dans les intervalles considérés, ou pas trop loin) à ceux que l'on a fait évaluer subjectivement. Pour entraîner le réseau, on utilise une partie des résultats obtenus avec les tests subjectifs et on garde une autre partie pour vérifier que le RNN est capable de faire des bonnes évaluations même pour des configurations qu'il n'a pas vues lors de la phase d'apprentissage (si les résultats sont acceptables, on dit que le réseau est validé).

Cette méthode permet d'avoir des bonnes estimations de qualité pour des variations importantes de tous les paramètres considérés, au coût de l'évaluation subjective de quelques échantillons.

## 3.1 RNN : Réseaux de Files d'Attente comme Outils d'Apprentissage

Dans cette Section nous présentons brièvement les principes mathématiques sous-jacentes aux RNN. Un RNN est en fait un type de réseau de files d'attente qui a été développé assez récemment (Gelenbe, 1989; Gelenbe, 1995; Gelenbe & Hussain, 2002).

1. Ce filtrage sert à éliminer les résultats des sujets qui ne sont pas fiables.

Il s'agit d'un réseau Markovien ouvert, avec des clients positifs et négatifs, aussi appelé un G-Network. On a $N$ nœuds (ou neurones) qui sont des files $./M/1$ (on notera le taux de service du nœud $i$ par $\nu_i$), inter-connectés, qui reçoivent et renvoient des clients depuis et vers l'environnement ainsi que de nœud à nœud. Le processus d'arrivée des clients positifs (négatifs) est Poisson avec un taux $\lambda_i^+$ ($\lambda_i^-$). À la sortie de la file $i$, un client abandonne le réseau avec une probabilité $d_i$, va vers la file $j$ en tant que client positif avec une probabilité $r_{ij}^+$, ou en tant que client négatif avec une probabilité $r_{ij}^-$. Lorsqu'un client négatif arrive dans une file, il tue le dernier client (s'il y en avait) et il disparaît (dans tous les cas). Les transferts entre files sont instantanés et donc on ne voit jamais un client négatif dans le réseau. À n'importe quel instant, on observe seulement des clients positifs dans le modèle. Les clients négatifs jouent un rôle de *signaux*, modifiant l'état du système.

Notons $X_t^i$ le nombre de clients dans la file $i$ à l'instant $t$. Il a été prouvé dans (Gelenbe, 1990; Gelenbe, 1995) que lorsque le processus Markovien $\vec{X}_t = (X_t^1, \cdots, X_t^N)$ est stable, sa distribution stationnaire est du type forme–produit. Donc, si l'on assume $(\vec{X}_t)$ stationnaire, on a :

$$\Pr(\vec{X}_t = (k_1, \cdots, k_N)) = \prod_{i=1}^{N} (1 - \varrho_i) \varrho_i^{k_i}.$$

Les facteurs $\varrho_1, \cdots, \varrho_N$ sont les charges des nœuds dans le réseau. Ces charges ne se calculent pas avec un système linéaire comme pour les réseaux de Jackson, mais avec le système non-linéaire suivant :

$$\varrho_i = \frac{\lambda_i^+ + \sum_{j=1}^{N} \varrho_j \nu_j r_{ji}^+}{\nu_i + \lambda_i^- + \sum_{k=1}^{N} \varrho_k \nu_k r_{ki}^-}. \tag{1}$$

On peut prouver que lorsque ce système a une solution $\varrho_1, \cdots, \varrho_N$ telle que pour chaque nœud $i$ on a $\varrho_i < 1$, alors le processus est stable et on a le résultat forme–produit (cf. (Gelenbe, 1990)).

Pour utiliser un tel réseau de files d'attente comme un outil d'apprentissage, on procède comme suit : les variables d'entrée du système (dans notre cas, les paramètres de réseau et codage considérés) correspondant à une configuration $\gamma_t$ sont normalisées en [0,1] et on les identifie aux taux d'arrivée des clients positifs à $P$ nœuds spécifiques, $\lambda_1^+, \cdots, \lambda_P^+$. Tous les autres taux d'arrivée de clients positifs depuis l'environnement sont mis à 0, de même que tous les taux d'arrivée de clients négatifs depuis l'environnement. La qualité associée à $\gamma_t$ lors de l'évaluation subjective est associée (après normalisation dans [0,1]) à la charge d'un nœud spécifique $o$, dit nœud *de sortie*. Le problème est alors réduit à trouver un réseau tel que lorsque $\lambda_1^+ = v_{1t}, \cdots, \lambda_P^+ = v_{Pt}$, la charge du nœud $o$ est proche de $\mu_t$, la qualité associée à $\gamma_t$, et ceci pour toutes les configurations que l'on souhaite utiliser pour l'apprentissage. Ceci est un problème d'optimisation où les variables de contrôle sont les paramètres restants du système, c'est à dire, les taux de service $\nu_i$ et les probabilités de routage $r_{ij}^+$ and $r_{ij}^-$.

Pour tous les nœuds $i$ tels que $d_i < 1$ (c'est à dire, ceux qui n'envoient pas tous leurs clients vers l'environnement), on note $w_{ij}^+ = \nu_i r_{ij}^+$ et $w_{ij}^- = \nu_i r_{ij}^-$. Ces facteurs sont appelés *poids*, comme pour les réseaux de neurones classiques et ils jouent un rôle similaire dans ce modèle. Normalement, lors de l'apprentissage, au lieu d'optimiser par rapport aux taux de service et aux probabilités de routage, on le fait par rapport aux poids et on fixe les taux de service des neurones "de sortie" (i.e., qui ont $d_i = 1$) à une valeur constante.

Pour entraîner le réseau on utilise un algorithme dit de *gradient descent*, qui peut être résumé comme suit. On dispose d'un ensemble de $K$ paires d'entrées/sorties, que l'on notera comme $\{(\vec{x}^{(k)}, \vec{y}^{(k)}),\ k = 1, \cdots, K\}$, avec $\vec{x}^{(k)} = (x_1^{(k)}, \cdots, x_N^{(k)})$ et $\vec{y}^{(k)} = (y_1^{(k)}, \cdots, y_N^{(k)})$. Le but de l'apprentissage est de faire en sort que si l'on a $\lambda_i^+ = x_i^{(k)} \forall i$ (avec $\lambda_i^- = 0$), la probabilité d'occupation des neurones de sortie à l'état stationnaire $\varrho_i$ soit proche de $y_i^{(k)}$ (dans notre cas, on a un seul neurone de sortie, et $y_i^{(k)} = \mu_k$), et ceci pour toutes les valeurs de $k$. Lors du démarrage de l'algorithme, on initialise arbitrairement les valeurs des poids, et on fait $K$ itérations qui les modifient. Soient $w_{i,j}^{+(0)}$ et $w_{i,j}^{-(0)}$ les valeurs initiales des connections entre

les neurones $i$ et $j$. Alors, pour $k = 1, \cdots, K$, l'ensemble de poids pour le pas $k$ est calculé à partir des poids du pas $k-1$, en utilisant un *schéma d'apprentissage*. Formellement, on note $\mathcal{R}^{(k-1)}$ le RNN obtenu au pas $k-1$, défini par les poids $w_{i,j}^{+\,(k-1)}$ et $w_{i,j}^{-\,(k-1)}$. Lorsque l'on affecte les taux d'entrée dans $\mathcal{R}^{(k-1)}$, l'on obtient (si on assume que le réseau est stable) $\varrho_i^{(k)}$, soit par résolution du système non-linéaire (1). Les poids au pas $k$ sont alors définis comme :

$$w_{i,j}^{+\,(k)} = w_{i,j}^{+\,(k-1)} - \eta \sum_{l=1}^{N} c_l (\varrho_l^{(k)} - y_l^{(k)}) \frac{\partial \varrho_l}{\partial w_{i,j}^{+}}, \tag{2}$$

$$w_{i,j}^{-\,(k)} = w_{i,j}^{-\,(k-1)} - \eta \sum_{l=1}^{N} c_l (\varrho_l^{(k)} - y_l^{(k)}) \frac{\partial \varrho_l}{\partial w_{i,j}^{-}}, \tag{3}$$

où les dérivées partielles $\partial \varrho_h / \partial w_{m,n}^{*}$ ('$*$' étant '+' ou '$-$') sont évaluées en $\varrho_h = \varrho_h^{(k)}$ et $w_{m,n}^{*} = w_{m,n}^{*\,(k-1)}$. Le facteur $c_l$ permet de donner des poids différents aux différents neurones de sortie. Le réel $\eta$ est le *facteur d'apprentissage*, qui permet de contrôler la vitesse de convergence et la précision de la procédure. Si pendant ce processus l'un des poids devient négatif, on le remet à 0 et on l'ignore lors des itérations suivantes. Une fois que les $K$ itérations sont finies, on recommence avec $k = 1$ jusqu'à ce que certaines conditions de convergence soient remplies. Cet algorithme minimise la fonction de coût

$$\frac{1}{2} \sum_{l=1}^{N} c_l (\varrho_l^{(k)} - y_l^{(k)})^2.$$

Le choix de ce type de réseau de neurones a été fait car les autres méthodes d'apprentissage que l'on a essayé n'ont pas été aussi performantes. Avec les RNN nous n'avons pas trouvé des problèmes de surentraînement, ce qui était le cas avec des ANN pour une application similaire (Mohamed & Rubino, 2002). Nous avons aussi essayé d'utiliser un filtre bayesien pour quantifier la qualité des flux de voix, mais les résultats obtenus n'ont pas été aussi précis que ceux obtenus avec les RNN. Finalement, les RNN ont d'autres avantages ; par exemple, il est très facile de calculer les dérivées par rapport aux paramètres d'entrée, ce qui permet de savoir comment la qualité varie avec chaque paramètre considéré.

# 4 Comparaison de la Performance de Notre Approche et d'Autres Méthodes Objectives

## 4.1 Performances des Méthodes Objectives

Pour déterminer la valeur d'une méthode d'évaluation de qualité dans un contexte donné, il faut savoir comment elle se comporte par rapport aux méthodes subjectives, qui sont la référence.

Dans cette section nous présentons des données de performance de quelques unes des méthodes d'évaluation objective les plus répandues qui sont disponibles dans la littérature (e.g. (Hall, 2001; Yang, 1999)). Nous présentons aussi des mesures de performance de notre approche, tirées de quelques expériences que nous avons menées. Il faut remarquer que nous ne pouvons pas comparer les résultats directement, étant donnée que les données proviennent d'expériences indépendantes. Néanmoins, la comparaison permet d'avoir une idée des performances relatives des différentes approches.

La plupart des méthodes objectives que l'on trouve actuellement dans la littérature ont été conçues pour déterminer les pertes de qualité introduits par exemple par les techniques de compression, etc. Généralement ces métriques ne sont donc pas prévues pour prendre en compte les dégradations que peut subir un flux en traversant le réseau. En plus, la plupart de ces métriques proposent une façon d'estimer la qualité essentiellement en comparant le signal émis et le signal reçu et donc elles ne sont pas adaptées à une utilisation en

temps réel (au moins comme métriques passives).

Dans la suite, nous présentons quelques mesures de performance pour les métriques suivantes : SNR, SNRseg, BSD, MBSD, EMBSD, PSQM, PSQM+, MNB(1,2), E-model, et PAMS. Ces mesures prennent en compte l'utilisation avec des dégradations dues au codage et aussi à la combinaison de codage et de la transmission à travers un réseau IP.

Pour déterminer la performance d'une métrique, on mesure le coefficient de corrélation des valeurs qu'elle produit avec les valeurs d'une évaluation subjective (MOS dans les cas que nous présentons). La Table 1 (sources : (Yang, 1999), (Voran, 1997), (Rix & Hollier, 2000) et (Rix, 1999)) montre les coefficients de corrélation pour plusieurs métriques quand on ne considère que les dégradations dues au codage. On peut observer que les méthodes les plus simples (SNR et SNRseg) ont une performance assez pauvre mais aussi qu'il y a des méthodes, telles que PSQM et PSQM+ qui donnent de très bons résultats. Lorsqu'il y a plusieurs valeurs pour le coefficient de corrélation, cela veut dire que la métrique à été évaluée avec des échantillons qui ont subi différents niveaux de dégradation.

| Métrique | Corrélation avec MOS |
|---|---|
| SNR | 0.22-0.52 |
| SNRseg | 0.22-0.52 |
| BSD | 0.36-0.91 |
| PSQM | 0.83-0.98 |
| PSQM+ | 0.87-0.98 |
| MNB2 | 0.74-0.98 |
| PAMS | 0.64-0.89 |
| MBSD | 0.76 |
| EMBSD | 0.87 |

TAB. 1 – *Coefficients de corrélation de quelques méthodes objectives d'évaluation de qualité avec le MOS. Ces résultats ont été pris dans littérature et seules les dégradations dues au codage sont considérées.*

Lorsque nous considérons aussi quelques paramètres de réseau (cf. Table 2), la performance de ces métriques exhibe une baisse importante. L'étude comparative la plus complète de ces métriques objectives dont nous avons connaissance est celle de (Yang, 1999). Malheureusement, à cause de contraintes de confidentialité imposées à l'auteur lors de la publication de son travail, tous les résultats ne sont pas donnés explicitement (il y a plusieurs méthodes qui ne sont pas nommées). De toute façon ceci ne gêne pas notre étude, car nous voulons avoir une idée de la performance de notre méthode par rapport aux autres méthodes objectives considérées. Les conditions de réseau considérées correspondent plutôt à un réseau de type GSM ou CDMA (erreurs à niveau des bits reçus, décalage temporel, *front clipping*, perte de paquets et variations de niveau), mais cela sert à montrer que lorsque l'on introduit d'autres facteurs de dégradation, les métriques ont du mal à fonctionner correctement. Il faut aussi noter que les coefficients de cette table ont été calculés à partir des courbes de régression et non pas des vraies estimations, ce qui peut les faire monter légèrement.

La Table 3 présente quelques résultats publiés dans (Yang, 1999; Hall, 2001), qui montrent les coefficients de corrélation obtenus pour les méthodes énumérées lors de leur utilisation avec des vrais flux VoIP. Les valeurs les plus élevées correspondent à celles présentées dans (Yang, 1999), exceptées celles de l'E-model, qui n'y est pas considéré. L'auteur n'a pas explicité les conditions de réseau ; il s'est limité à dire qu'il s'agissait de flux VoIP. Dans (Hall, 2001) les paramètres de réseau considérés sont le taux de perte (0%, 1% et 5%, la distribution des pertes n'étant pas spécifiée) et la gigue (0, 50 et 100ms de variation pour un délai de bout en bout non spécifié). Nous pensons que les différences de performance entre les deux études sont données par des conditions de réseau un peu plus dures dans (Hall, 2001), ce qui peut avoir dégradé davantage les échantillons et donc affecté aussi davantage la performance des métriques.

Parmi toutes les méthodes d'évaluation objective de la qualité des flux de voix que l'on a considérées, seul l'E-model est conçu pour travailler sans accéder au signal original, ce qui, avec ses faibles besoins de puissance de calcul fait de cette approche une bonne option pour des applications en temps-réel. Cette métrique a été conçue pour aider à la conception de nouveaux réseaux, et non pas comme une métrique de

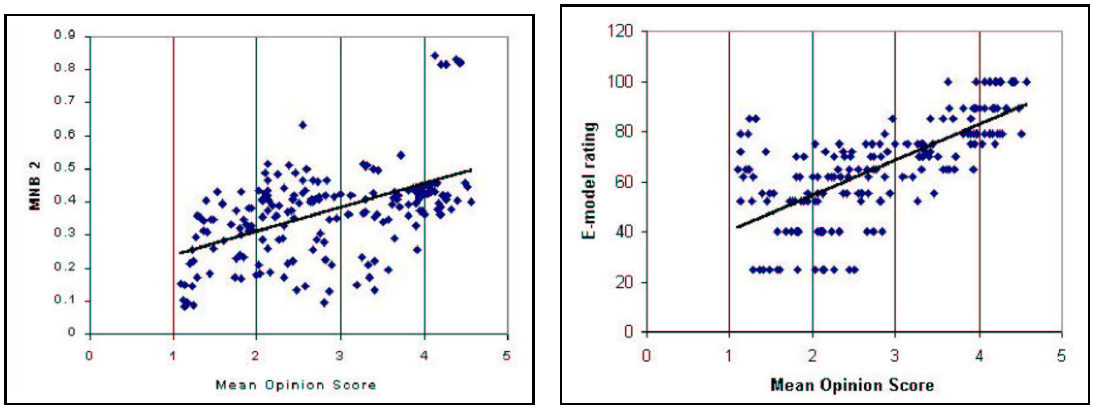
| Métrique | Corrélation avec MOS |
|---|---|
| A | 0.87 |
| B | 0.85 |
| C | 0.56 |
| D | 0.86 |
| E | 0.90 |
| F | 0.86 |
| MBSD | 0.24 |
| EMBSD | 0.54 |

TAB. 2 – *Coefficients de corrélation pour quelques méthodes d'évaluation objective avec le MOS, pour des dégradations dues au codage et à quelques paramètres de réseau que l'on peut trouver sur un réseau de type GSM ou CDMA. Noter que toutes les métriques n'ont pas été nommées explicitement.*

| Métrique | Corrélation avec MOS |
|---|---|
| EMBSD | 0.39 – 0.87 |
| MNB1 | 0.61 – 0.83 |
| MNB2 | 0.63 – 0.74 |
| E-model | 0.62 – 0.86 |

TAB. 3 – *Coefficients de corrélation pour EMBSD, MNB(1 & 2) et l'E-model avec MOS pour VoIP, pris de (Yang, 1999) et (Hall, 2001).*

qualité en soi (ITU-T Recommendation G.107, 2003). Ceci explique son manque de précision dans certaines conditions. D'ailleurs, il est encore en développement, et par exemple, c'est seulement depuis Mars 2003 qu'il y a une prévision explicite pour considérer les pertes de paquets comme un paramètre en entrée, et là encore, il n'y a rien de prévu pour les rafales de pertes.




(a) Performance de MNB2

(b) Performance de l'E-model

FIG. 1 – Scatter plot *des résultats de MNB2 et l'E-model contre des MOS pour un ensemble de échantillons dégradés par le codage et la transmission sur le réseau (pris de (Hall, 2001)).*

Comme un exemple des problèmes que l'on peut trouver aujourd'hui avec les méthodes objectives disponibles dans la littérature, nous présentons dans la Figure 1 deux *scatter plots*, l'un pour MNB2 et l'autre pour l'E-model. Ces figures montrent clairement qu'il y a beaucoup de valeurs inconsistantes entre ces métriques et les évaluations subjectives (points avec le même valeur pour l'évaluation objective et des écarts très importants pour le MOS, et vice-versa).

### 4.2 Performance de l'Approche Pseudo–subjective

Nous présentons ici des mesures de performance de notre approche sur des flux VoIP. Nous avons utilisé le Robust Audio Tool (RAT) (London, 2002) sur une maquette de réseau où l'on a généré des pertes selon un modèle de Gilbert (Gilbert, 1960) simplifié, qui permet d'obtenir des processus de pertes très semblables à ceux observables sur Internet (Yajnik *et al.*, 1999; Bolot & Vega Garcia, 1996). Nous avons aussi utilisé un mécanisme de correction d'erreurs à l'avance (*Forward Error Correction* – FEC) et nous avons tenu compte de ses paramètres dans notre évaluation. La Table 4 montre les paramètres que nous avons considéré dans notre expérience (il faut noter que notre approche permet de considérer n'importe quel ensemble de paramètres qui soient d'intérêt).

Nous avons gardé environ 115 configurations parmi toutes les configurations possibles. À partir de 12 échantillons originaux nous avons généré 115 groupes de 4 séquences et nous avons procédé à réaliser une évaluation subjective avec un groupe de 17 sujets. Après le filtrage statistique des résultats, nous avons éliminé ceux de l'un des sujets. Avec ces résultats, nous avons procédé à entraîner plusieurs RNN, avec des différentes architectures et avec plusieurs variations dans les tailles des ensembles d'entraînement et de validation. Les résultats obtenus ne présentent pas beaucoup de variation et on a trouvé que même avec des architectures très simples pour le RNN, on peut avoir de très bonnes estimations. Les coefficients de corrélation (entre la sortie du RNN et les MOS) obtenus vont de 0.73 à 0.93 (les résultats plus faibles correspondent aux tests avec une faible taille de l'ensemble d'entraînement). La Figure 2 montre un *scatter plot* pour un ensemble de validation et un RNN *feed forward* de 3 couches. Dans la Figure 3 nous montrons les MOS réels et estimés par le RNN pour le même ensemble de valeurs de validation. Nous avons constaté que l'erreur absolue la plus importante est de 0.58 points dans l'échelle de cinq points utilisée. Une différence de cette magnitude est à peine perceptible par un utilisateur moyen.

| **Parameter** | **Values** |
|---|---|
| Loss rate | 0% . . . 15% |
| Mean loss burst size | 1 . . . 2.5 |
| Codec | PCM Linear 16 bits, GSM |
| FEC | ON(GSM)/OFF |
| FEC offset | 1 . . . 3 |
| Packetization interval | 20, 40, and 80ms |

TAB. 4 – *Paramètres de codage et de réseau utilisés pour notre expérience.*

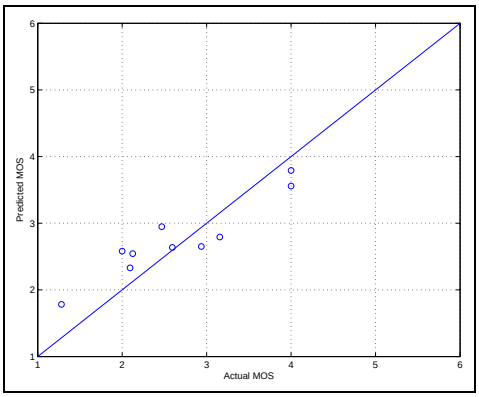



FIG. 2 – *Scatter plot pour un ensemble de validation (pas connu du RNN). Coefficient de corrélation = 0.93*

## 5 Conclusions

Dans cet article nous avons présenté une méthode pour l'évaluation de qualité des flux VoIP récemment développée, et nous avons étudié sa performance par rapport aux méthodes les plus répandues que l'on

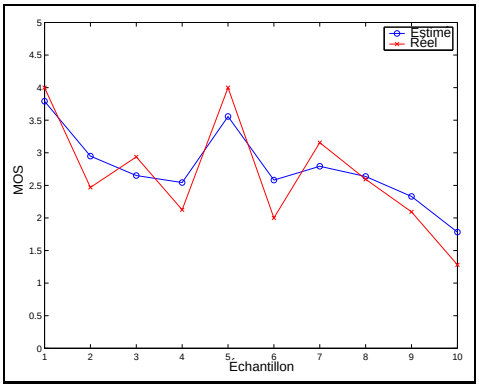



FIG. 3 – *MOS réel et estimé par le RNN pour l'ensemble de validation. Erreur absolue maximale : 0.58*

trouve aujourd'hui dans la littérature.

Notre méthode a des performances comparables à celles des méthodes objectives les plus performantes, et même meilleures dans les cas où l'on rencontre des conditions de réseau semblables à celles d'Internet. De plus, nôtre méthode permet d'évaluer la qualité d'un flux reçu sans accéder à la version original (c'est à dire, telle qu'émisse par la source), ce qui, ajouté à son faible coût en calcule, la fait adaptée à l'évaluation en temps-réel. Ceci permet de nombreuses applications, par exemple pour le contrôle dynamique de la qualité, la vérification des SLA (*Service Level Agreements*) pour la tarification, etc.

# Références

BEERENDS J. (1997). Improvement of the p.861 perceptual speech quality measure. ITU-T SG12 COM-34E.

BEERENDS J. & STEMERDINK J. (1994). A perceptual speech quality measure based on a psychoacoustic sound representation. *Journal of Audio Eng. Soc.*, **42**, 115–123.

BOLOT J.-C., FOSSE-PARISIS S. & TOWSLEY D. (1999). Adaptive FEC–based error control for Internet telephony. In *Proceedings of INFOCOM '99*, p. 1453–1460, New York, NY, USA.

BOLOT J.-C. & VEGA GARCIA A. (1996). The case for FEC-based error control for packet audio in the Internet. In *ACM Multimedia Systems*.

CHOI A. & CONSTANTINIDES A. (1989). Effect of packet loss on 3 toll quality speech coders. In *Second IEE National Conference on Telecommunications*, p. 380–385, York, UK.

CLAYPOOL M. & TANNER J. (1999). The effects of jitter on the perceptual quality of video. In *Proceedings of ACM Multimedia Conference*.

GELENBE E. (1989). Random neural networks with negative and positive signals and product form solution. *Neural Computation*, **1**(4), 502–511.

GELENBE E. (1990). Stability of the random neural network model. In *Proc. of Neural Computation Workshop*, p. 56–68, Berlin, West Germany.

GELENBE E. (1995). G-networks: new queueing models with additional control capabilities. In *Proceedings of the 1995 ACM SIGMETRICS joint international conference on Measurement and modeling of computer systems*, p. 58–59, Ottawa, Ontario, Canada.

GELENBE E. & HUSSAIN K. (2002). Learning in the multiple class random neural network. *IEEE Trans. on Neural Networks*, **13**(6), 1257–1267.

GILBERT E. (1960). Capacity of a burst–loss channel. *Bell Systems Technical Journal*, **5**(39).

HALL T. A. (2001). Objective speech quality measures for Internet telephony. In *Voice over IP (VoIP) Technology, Proceedings of SPIE*, volume 4522, p. 128–136, Denver, CO, USA.

HANDS D. & WILKINS M. (1999). A study of the impact of network loss and burst size on video streaming quality and acceptability. In *Interactive Distributed Multimedia Systems and Telecommunication Services Workshop*.

ITU-T RECOMMENDATION G.107 (2003). The E-model, a computational model for use in transmission planning.

ITU-T RECOMMENDATION P.800 (1996). Methods for subjective determination of transmission quality.

LONDON U. C. (2002). Robust Audio Tool website. http://www-mice.cs.ucl.ac.uk/multimedia/software/rat/index.html.

MOHAMED S. & RUBINO G. (2002). A study of real–time packet video quality using random neural networks. *IEEE Transactions On Circuits and Systems for Video Technology*, **12**(12), 1071 –1083.

MOHAMED S., RUBINO G. & VARELA M. (2004). Performance evaluation of real-time speech through a packet network: a random neural networks-based approach. *Performance Evaluation*, **57**(2), 141–162.

HODSON, O. (2002). Packet Reflector.

RIX A. (1999). Advances in objective quality assessment of speech over analogue and packet-based networks. In *the IEEE Data Compression Colloquium*, London, UK.

RIX A. & HOLLIER M. (2000). The perceptual analysis measurement system for robust end-to-end speech assessment. In *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing – ICASSP*, p. 1515–1518, Istanbul, Turkey.

VORAN S. (1997). Estimation of perceived speech quality using measuring normalizing blocks. In *IEEE Workshop on Speech Coding For Telecommunications Proceeding*, p. 83–84, Pocono Manor, PA, USA.

YAJNIK M., MOON S., KUROSE J. & TOWSLEY D. (1999). Measurement and modeling of the temporal dependence in packet loss. In *Proccedings of IEEE INFOCOM '99*, p. 345–352.

YANG W. (1999). *Enhanced Modified Bark Spectral Distortion (EMBSD): an Objective Speech Quality Measrure Based on Audible Distortion and Cognition Model*. PhD thesis, Temple University Graduate Board.