Algorithmes pour l'estimation des données dans les réseaux de capteurs

Alia Ghaddar[1,2] and Tahiry Razafindralambo[1] and Isabelle Simplot-Ryl[1] and David Simplot-Ryl[1] and Samar Tawbi[2]

[1] *INRIA/ CNRS / Université de Lille 1, France*
[2] *Université Libanaise, Liban*

La collecte des données est un des enjeux majeurs dans les réseaux de capteurs. En effet, les communications induites par la transmission de données réduisent considérablement la durée de vie du réseau. Une des techniques utilisées pour réduire la quantité de données transférées est l'agrégation et selon le type des données étudiées, une des possibilités est l'utilisation de série temporelle ARMA. Dans cet article, nous proposons quatre algorithmes d'agrégation de données s'appuyant sur le modèle AR permettant ainsi la diminution de la consommation d'énergie dans les réseaux de capteurs et augmentant la durée de vie de ceux-ci.

Keywords: Réseaux de capteurs, agrégation temporelles, modèle ARMA

1 Introduction

Les réseaux des capteurs constituent un des domaines de recherche les plus actifs actuellement. Ils se révèlent utiles dans de nombreuses applications pour collecter et traiter des informations provenant de l'environnement grâce à des nœuds capteurs[ASSC02] : mesure des températures, surveillance militaires, automatisation dans les bâtiments, sécurité, etc. Les capteurs utilisés dans ces réseaux ont des capacités limitées tant en mémoire qu'en termes de puissance de calcul. De plus, ils sont souvent deployés dans des environnements accidentés et, une fois deployés, y accéder physiquement est difficile, voire impossible. Remplacer la batterie du capteur est impossible, ce qui fait de cette dernière une ressource critique. Les capteurs collectent et transmettent des données d'une façon autonome à d'autres nœuds appelés puits. En raison de la taille de l'espace à surveiller ces transmissions se font souvent par plusieurs sauts en utilisant d'autres capteurs comme relais. La transmission d'une seule donnée peut ainsi impliquer plusieurs capteurs et une consommation d'énergie sur chacun d'eux. Une des techniques permettant de réduire la quantité de données transférées est l'agrégation de données [IEGH02] et [Cay03], qui permet de réduire la charge de traffic acheminé dans le réseaux [KES02]. Deux types d'agrégation existent : l'agrégation spatiale où l'on cherche à combiner des informations venant de différentes sources et l'agrégation temporelle venant de différentes périodes de temps. Dans cet article, nous nous limitons au dernier type, et proposons différents algorithmes d'agrégation temporelle de données.

La section 2 introduit le modèle ARMA et une vue rapide sur les travaux du domaine. Dans la section 3, nous proposons quatres algorithmes d'estimation de données afin de réduire le coût de transmission. La section 4 contient une étude de cas basée sur des données réelles pour comparer la qualité des algorithmes proposés. La section 5 conclut notre travail.

2 État de l'art

Dans cette section, nous donnons une définition du modèle ARMA et une revue rapide sur les travaux du domaine. Les modèles ARMA sont les principaux modèles de séries temporelles. Étant donnée une série temporelle X_t, ils sont utilisés pour comprendre et prédire, éventuellement, les valeurs futures de cette série.

Un modèle ARMA(k,q) est composé de deux parties : une partie autorégressive (AR) d'ordre *k* qui analyse l'autocorrélation dans la série temporelle et joue un rôle important dans les applications car elle peut être extrapolée linéairement à tout instant *n*, à partir des *k* valeurs précèdentes. L'autre partie de la série est la moyenne-mobile (MA) d'ordre *q*. La série (AR) se présente sous la forme :

$$X_{i+k} = a_0 + a_1 X_{i+k-1} + \ldots + a_k X_i \quad (1)$$

où a_0,...,a_k sont les paramètres du modèle, X_t est une série temporelle. Différentes approches d'agrégation temporelle des données [TSM06] basées sur la prédiction des mesures ont produit un effet important sur la réduction du coût de communication. Le puits exploite un modèle de série temporelle pour prédire localement des données au lieu d'une communication directe avec le capteur. Certaines approches [TSM06], [CDHH06], [Cha04] utilisent des modèles AR/ARMA (autoRegression/AutoRegression Moving Average). Selon [Val07], chaque nœud, en plus de son activité de surveillance de l'environnement, calcule un modèle ARMA basé sur la série de W échantillons consécutifs afin d'exploiter la corrélation temporelle des données. Les étapes d'estimation des paramètres du modèle sont comme suit : dès qu'un nœud a collecté *W* échantillons, il calcule localement le modèle ARMA (k,q) : $k+1$ paramètres sont calculés pour la partie AR et sont transmis ainsi que les *k* premiers échantillons. Régulièrement, dès qu'un capteur collecte *S* nouveaux échantillons, il mesure l'erreur *e* entre ces nouveaux échantillons et les *S* valeurs prédites à partir du modèle précédent. Si l'erreur est inférieure à un seuil, le modèle précédent est conservé et aucune transmission d'information n'est réalisée. Sinon, le nœud recalcule les coefficients ARMA à partir des *W* échantillons les plus récents et les transmet au puits. Ainsi le modèle ARMA joue un rôle important dans la consommation d'énergie, par exemple si un nœud transmet 40 coefficients ARMA complexes et 39 échantillons au lieu de 5000 données brutes, ça permet de réduire plus de 90% (79/5000=0.015) la consommation d'énergie. Cependant, il a été démontré que la transmission d'un bit est équivalente, en termes d'énergie à l'exécution d'environ 1000 instructions. Dans cet article, nous essayons de réduire la consommation d'énergie par la modélisation de séries temporelles linéaires en utilisant le processus AR du modèle ARMA, dans le but de chercher une corrélation des données qui permet de réduire la communication tout en préservant la précision des données estimées. L'utilisation de la série AR et non la série ARMA complète permet aussi de réduire la complexité de calcul sur chaque capteur.

3 Algorithmes proposés

Nous allons ignorer dans cette article, la partie MA qui capture le comportement aléatoire et qui est indépendante du processus AR et nous n'utiliserons que la partie AR qui convient aux cas où les mesures ne varient pas rapidement comme la mesure des températures pour des raisons de simplicité.
Basiquement, les capteurs remontent les informations de l'environnement surveillé vers le puits. En utilisant une série AR, ce sont les paramètres de la série qui sont transmis au puits (si nécessaire) pour reproduire les données perçues par le capteur. Toutefois, chaque remontée entraîne une charge de communication et un coût en énergie non négligeable. Pour réduire la charge du trafic acheminé, nous proposons différents algorithmes ayant des propriétés différentes (nous prenons : *S*=1, *W*=*k*).

3.1 Algorithme 1

Cet algorithme se base sur l'idée d'envoyer k valeurs au puits au lieu des 2k+1 valeurs envoyées par le modèle décrit dans la section précédente. Ainsi, un capteur n'a pas besoin d'envoyer k mesures et k+1 paramètres. Si une estimation dépasse le seuil, seulement k nouvelles mesures sont envoyées au puits pour recalculer la série. Ainsi, le puits et le capteur, chacun à part, calculent localement les paramètres de la série prédictive. L'algorithme consiste à construire et résoudre un système d'équations pour recalculer les paramètres de la série prédictive. *Exemple* : Prenons k=3. Si l'estimation X_t dépasse le seuil, des nouvelles données réelles N_t, N_{t+1}, N_{t+2} sont envoyées au puits. Mais pour construire le système d'équations, on a encore besoin de k+1 valeurs précédentes à N_t. On note par (X_i), la série de k+1 valeurs approximées des valeurs réelles précédentes à N_t (plus ou moins une certain seuil) avec i $\in \{t-k-1,..,t-1\}$ et $X_i \in [N_t - seuil, N_t + seuil]$. Le système d'équations sera construit à partir des informations suivantes : (X_i), N_t, N_{t+1} et N_{t+2}. Les équations sont : N_{t+2}=a_0+a_1 N_{t+1}+a_2 N_t+a_3 X_{t-1}, N_{t+1}=a_0+a_1 N_t+a_2 X_{t-1}+a_3 X_{t-2},

N_t=a_0+a_1 X_{t-1}+a_2 X_{t-2}+a_3 X_{t-3}, X_{t-1}=a_0+a_1 X_{t-2}+a_2 X_{t-3}+a_3 X_{t-4}.
Ce premier algorihtme est une approche très simple de l'agrégation de données.

3.2 Algorithme 2

Dans cet algorithme, nous initialisons la série AR par les valeurs suivantes :

$$X_i = N_0, a_0 = 0, a_i = 1/k \text{ pour } i \leq 0 \quad (2)$$

Cet algorithme réduit le coût de transmission de l'algorithme précédent, en envoyant une seule mesure N_i au lieu de k mesures. Le ré-ajustement de la série prédictive (a_i) à partir de la valeur envoyée N_i et produisant une série ajustée ($a_i^{'}$), se fait d'une façon itérative et se base sur le fait d'augmenter la valeur de X_i de $\frac{e}{k+1}$ à chaque fois qu'on ajuste un paramètre (a_i) de la suite. Par exemple, après avoir ajusté le k^{ème} paramètre de la série ($a_k^{'}$), X_i sera égal à (N_i-e) + $\frac{e}{k+1}$ =N_i- $\frac{ke}{k+1}$. Étant donné k=3, une mesure N_i et une valeur X_i qui dépasse le seuil (e_i=N_i-X_i). Le ré-ajustement se fait de la manière suivante :
D'abord ajuster le 3ème paramètre a_3 comme suit : a_0+ a_1 X_{i-1}+a_2 X_{i-2}+$a_3^{'}$ X_{i-3}=X_i+ $\frac{e}{k-1}$puis utiliser le paramètre ajusté $a_3^{'}$ pour calculer $a_2^{'}$. X_i est alors augmenté de $\frac{2e}{k+1}$ et ainsi de suite. Pour généraliser, a$_j^{'}$ est ajusté de la manière suivante : $a_j^{'}$=$\frac{1}{X_{i-j}}$ [N_i-$\frac{je}{k+1}$-(a_0+$\sum_{t=1}^{j-1}$ a_t X_{i-t}+$\sum_{p=j+1}^{k}$ $a_p^{'}$ X_{i-p})].

3.3 Algorithme 3

La réduction du coût de transmission de cet algorithme se manifeste par la diminution du nombre de bits envoyés vers le puits. En fait, le capteur au lieu d'envoyer la valeur N_i, il envoie l'erreur e entre la valeur estimée X_i et la valeur réelle N_i. Ainsi, le puits qui a déjà estimé X_i, en recevant e peut simplement trouver N_i en calculant $N_i = X_i + e$. En suivant le même principe que dans l'algorithme précèdent, X_i est augmentée de $\frac{e}{k+1}$ pour ajuster le paramètre (a_k), et de $\frac{e(i+1)}{k+1}$ pour ajuster le paramètre (a_{k-i}). Pour généraliser, le j^{ème} paramètre de la série ajustée est calculé de la façon suivante : $a_j^{'} = \frac{1}{(X_{i-j})}[X_i + \frac{(k+1-j)e}{k+1} - (a_0 + \sum_{t=1}^{j-1} a_t X_{i-t} + \sum_{p=j+1}^{k} a_p^{'} X_{i-p})]$. La figure 1 montre qu'un nombre important d'erreurs ont une valeur plus petite que 1, alors au lieu d'avoir un traffic de communication en termes de bits provenant toujours des valeurs flottantes (Algorithmes 1 et 2), l'envoi de la partie décimale des erreurs ayant des valeurs plus petites que 1, réduit bien le traffic de communicaton en termes de bits.

3.4 Algorithme 4

Dans le but d'assurer plus de précision et de diminuer le plus possible le dépassement de seuil, un capteur n'envoie pas seulement l'erreur $e = N_i - X_i$ mais aussi les k erreurs précédentes. Ayant k estimations et k erreurs précédentes, le calcul des $a_j^{'}$ peut assurer une estimation de données plus proche des valeurs réelles que le calcul dans les algorithmes précédents (1 et 2).

4 Comparaison des différents algorithmes

La figure 1 fournit l'erreur entre les données collectées et les prédictions de chaque algorithme (L'algorithme 1 n'apparaît pas dans la figure due à sa faible précision). L'erreur est obtenue sur un jeu de données contenant 1033 échantillons de températures provenant de [Val07], un seuil de 0.05 et un modèle d'ordre k=3. Bien que l'algorithme 1 ait réduit la charge du traffic face au modèle ARMA de 2k+1 valeurs à k valeurs, sa précision est trop faible pour être retenu. D'autre part, l'algorithme 2 réduit le coût de transmission à une seule valeur face au modèle ARMA et produit les mêmes estimations que l'algorithme 3. Alors que, dans l'algorithme 3, le nombre de bits transmis est fortement réduit par l'envoi d'une valeur d'erreur (e=N_i-X_i). Le rapport des estimations acceptables sur le nombre total des données est donc égal à 668/1032=0.647, ce qui signifie que le nombre de mises à jour du modèle est donc diminué d'environ 65%. L'algorithme 4 produit un rapport de 717/1030=0.696, le nombre de mises à jour est donc diminué d'environ de 70%. Nous remarquons aussi que la precision des estimations est améliorée de 56% puisque l'algorithme 4 produit 577 estimations plus précises que l'algorithme 3. Il faut noter que dans le pire des cas, si le modèle proposé dans

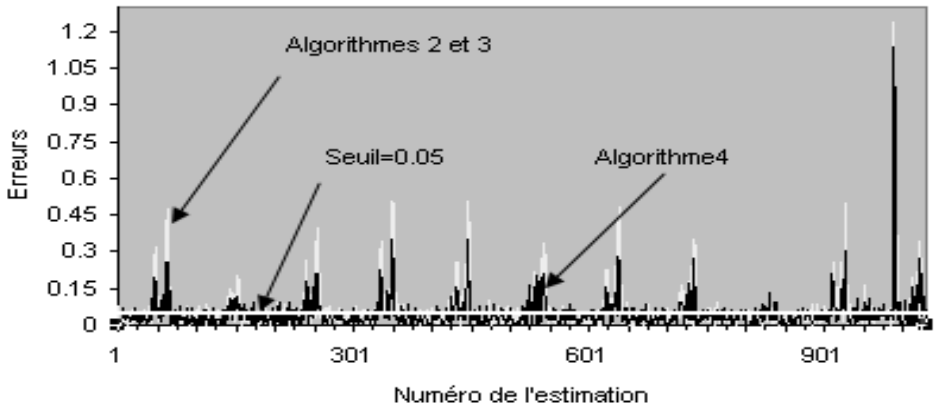

FIGURE 1: Les erreurs de température produites par les algorithmes 2, 3 et 4.

[Val07] produit des estimations qui ne dépassent pas le seuil, la figure 1 montre que les erreurs produites par l'algorithme 4 tendent vers zéro pour produire des estimations très proches de celles des séries ARMA mais avec une diminution du coût de transmission. L'algorithme 4 est dans ce sens un algorithme adapté puisqu'il est efficace en termes de consommation d'énergie et de précision.

-	algorithme 1	algorithme 2	algorithme 3	algorithme 4
# Valeurs acceptées	0	668	668	717
# Valeurs non acceptées	1030	364	364	313
Traffic/ajustement	k mesures	1 mesure	1 erreur	k erreurs
Traffic total de communication(k=3)	3090 mesures	364 mesures	364 erreurs	939 erreurs

5 Conclusion

Dans cet article, notre but était de diminuer la consommation d'énergie dans les réseaux de capteurs en réduisant la charge de communication entre les nœuds. En partant d'un modèle ARMA, nous avons proposé quatres algorithmes d'estimation de données et avons comparé l'efficacité de chacun en termes de précision des estimations et sur consommation minimale d'énergie. Les résultats montrent qu'il est possible de minimiser le nombre de ré-ajustements de la série prédictive mais aussi de diminuer le nombre d'opérations de calcul pour pouvoir prolonger la durée de vie d'un nœud.

Références

[ASSC02] Ian F. Akyildiz, Weilian Su, Yogesh Sankarasubramaniam, and Erdal Cayirci. Wireless sensor networks : a survey. *Computer Networks*, vol. 38(no. 4) :393–422, 2002.

[Cay03] E. Cayirci. Data aggregation and dilution by modulus addressing in wireless sensor networks. *IEEE Communications Letters 7*, pages 355–357, 2003.

[CDHH06] D. Chu, A. Deshpande, J. M. Hellerstein, and W. Hong. Approximate data collection in sensor networks using probabilistic models. *ICDE'06*, page 48, 2006.

[Cha04] C. Chatfield. *The Analysis of Time Series*. CRC Press, 2004.

[IEGH02] C. Intanagonwiwat, D. Estrin, R. Govindan, and J. Heidemann. Impact of network density on data aggregation in wireless sensor networks. pages 575–578. Proceedings of the 22nd International Conference on Distributed Computing Systems, July 2002.

[KES02] B. Krishanamachari, D. Estrin, and S.Wicker. The impact of data aggregation in wireless sensor networks. Vienna, Austria, July 2002. in Proceedings of the International Workshop on Distributed Event Based Systems (DEBS).

[TSM06] D. Tulone and PAQ : S. Madden. Time series forecasting for approximate query answering in sensor networks. pages 21–37. In Proc. of the 3rd Euro. Conf. in WSN, 2006.

[Val07] Fabrice Valois. *Auto-organisation de réseaux radio multi-sauts*. Hdr, INSA Lyon, 2007.