

Codage réseau fiable

De nos jours, la théorie du codage réseau est de plus en plus présente dans des domaines très variés de l'informatique contemporaine. Bien que la vision que porte la communauté sur les réseaux de communication ait changé grâce à cette théorie, ses implications sur la façon dont nous étudions les architectures en *code-and-forward* est toujours sous-estimée. Plus précisément, nous constatons que même si beaucoup de chercheurs dans les réseaux informatiques sont maintenant familiers avec les avantages des codes opportunistes tels que COPE [47], leurs approches pour valider leurs travaux dans ce domaine restent très proches des méthodologies propres aux réseaux *store-and-forward*. Cela est dû à l'absence dans la littérature d'un cadre théorique dédié qui permette l'évaluation et la validation des systèmes de codage réseau.

De fait, il est indéniable que dans un environnement idéal le codage réseau opportuniste permet d'améliorer l'utilisation de la bande passante et de réduire la latence des réseaux maillés et hiérarchiques. Pourtant, il subit les mêmes défaillances que d'autres systèmes de codage basés sur des méthodes d'acquiescement et de retransmissions venues du monde du *store-and-forward*. Ces défaillances existent parce que les codes proposés dans la littérature, qu'ils soient linéaires ou opportunistes, ne sont conçus que pour garantir une fiabilité dite « classique » des communications au lieu d'adopter des méthodes spécialement conçues pour le codage réseau.

Aussi, avant d'entamer une quelconque étude de fiabilité du codage inter-machines, nous devons dans un premier temps savoir si les méthodes d'appréciation de la fiabilité utilisées dans le *store-and-forward* sont suffisantes pour accomplir cette tâche. Concrètement, dans un réseau non-codé, garantir la fiabilité des communications consiste à maximiser le taux d'acheminement à un saut et de bout-en-bout tout en gardant des délais raisonnables et une certaine équité. Cela peut s'évaluer simplement en mesurant les délais et les taux de pertes de paquets de proche-en-proche et de bout-en-bout. Cependant, dans un système codé, l'acheminement réussi d'un paquet ne rime pas forcément avec succès de la communication, car un paquet codé qui arrive à son destinataire n'a de valeur que s'il peut être décodé.

Afin de remédier à ces insuffisances nous définissons dans ce chapitre la notion de *codage fiable*. Nous affirmons que ce concept constitue un élément important dans l'évaluation et la conception de nouveaux mécanismes de codage réseau. Étonnement, il n'a jamais (ou pas explicitement) été pris en compte dans la littérature actuelle, même s'il peut avoir un impact négatif sur l'efficacité des algorithmes de codage conçus pour les réseaux sans fil, où le médium est par nature très peu fiable. L'élément clé du codage fiable est de fournir un premier ensemble d'outils formels qui contribuent à combler le fossé entre l'implémentation du codage réseau et sa fiabilisation, et cela en répondant à des questions cruciales. Notamment :

1. Est-ce que les mécanismes de codage réseau subissent la même dégradation de performances que ceux en *store-and-forward* en présence de pertes de paquets ?
2. Est-ce que les métriques classiques de performances (débit, délai, taux d'acheminement) suffisent à évaluer les réseaux codés ?
3. Et finalement, peut-on évaluer la fiabilité des algorithmes de codage de la même manière que celle des systèmes en *store-and-forward* ?

Dans un second temps, nous proposons d'illustrer comment un codage *non fiable* peut réduire les performances des réseaux de capteurs sans fil et comment il est possible d'y remédier. Comme exemple empirique, nous montrons la façon dont les interférences et les collisions peuvent influencer sur le comportement de CoZi, le mécanisme de codage opportuniste proposé dans le chapitre précédent. Puis nous présentons ReCoZi une solution simple, et pourtant efficace, qui se base sur un mécanisme d'acquiescement par écho [91] et sur l'inférence de topologie locale pour améliorer la robustesse du codage réseau lors de présence de pertes non-négligeables. Enfin, nous montrons que ReCoZi surpasse nettement le routage en *store-and-forward* ZigBee en termes de taux d'acheminement avec cependant une légère baisse en termes de latence dans certains cas.

Ce chapitre est organisé comme suit. Dans la section 4.1 nous abordons le problème du codage fiable dans un réseau de capteurs sans fil. Nous y étudions le comportement des algorithmes de codage opportunistes face aux pertes de paquets et définissons la notion de codage fiable. Nous montrons que ce dernier permet d'offrir des communications plus robustes que ce soit de proche-en-proche ou de bout-en-bout. Nous développons ensuite dans la section 4.2 l'évaluation de ReCoZi un mécanisme de retransmission spécialement conçu pour assurer un codage réseau fiable quelques soient les conditions du médium dans un réseau de capteurs sans fil. Nous y montrons les performances et discutons son apport pour le codage opportuniste inter-machines. Enfin, la section 4.3 conclut le chapitre.

4.1. Codage Fiable

Il est important de comprendre que ce que nous définissons comme codage fiable diffère de la notion classique de fiabilité dans les réseaux de communication par paquets. En effet, la fiabilité dans les réseaux sans fil par exemple fait référence à une maximisation du taux d'acheminement des données en utilisant des mécanismes de retransmission pour permettre aux paquets de traverser le réseau et d'arriver sans erreurs à leur destination [93].

Un codage réseau fiable concerne plutôt la capacité d'un nœud à encoder des paquets de manière suffisamment efficace pour qu'un nombre maximum de ses prochains sauts puissent en extraire de l'information utile, et cela quelques soient les conditions du réseau (interférences, collisions, défaillances de nœuds, etc.). Ainsi, nous proposons les métriques suivantes qui définissent les principaux critères qu'un algorithme de codage doit satisfaire pour garantir des communications fiables et efficaces.

4.1.1. Inférence de topologie

Cette métrique représente l'un des paramètres clés dans la fiabilisation des algorithmes de codage réseau, car une prédiction erronée de la disponibilité des paquets dans le voisinage d'un nœud peut conduire au choix d'un code sous-optimal voir à une combinaison de paquets non-decodable. La Figure 24 (a) montre un exemple dans lequel un nœud intermédiaire i fait une prédiction exacte de ce que ses voisins à un saut ont déjà reçu. Dans ce cas de figure, le nœud effectue une opération de codage optimale qui permet à l'intégralité des nœuds voisins de décoder le paquet. Inversement, les Figure 24 (b) et (c) illustrent des situations où i encode des paquets en se basant sur une inférence de topologie locale incorrecte causée par la perte de paquets de *feedback*, ce qui conduit respectivement, à un décodage partiel dans (b) et impossible dans (c).

Plusieurs mécanismes existent pour permettre une inférence de topologie en quasi temps-réel dans un réseau de communication par paquets. Dans les réseaux statiques, la topologie environnante d'un nœud peut être facilement prédite à l'aide de méthodes telles que [51] ou [63] qui utilisent de simples heuristiques pour « deviner » ce que les nœuds voisins ont déjà reçu ou écouté. Ces techniques peuvent rapidement devenir inefficaces chaque fois que le canal de communication subit des pertes trop importantes qui font que le nœud intermédiaire n'arrive pas à faire des prédictions précises. Dans [47], les auteurs optent pour l'utilisation de rapports de réception afin d'informer un nœud codeur de l'ensemble de paquets disponibles au niveau de ses voisins. Cette approche est efficace parce que l'inférence de topologie est réalisée en temps réel. Toutefois, elle peut impliquer une surcharge de paquets de contrôle élevée, ce qui n'est pas souhaitable pour les réseaux de capteurs où le nombre de transmissions doit être minimisé.

4.1.2. Décodabilité

Nous appelons décodabilité la probabilité $Dec(c)$ qu'un code c sélectionné par un nœud s soit décodé par toutes ses destinations. L'équation suivante décrit la valeur de Dec .

$$Dec(c) = \begin{cases} \sum_{i=0}^{N(s)} \frac{P_{dec_i}(c)}{N(s)}, & N(s) \neq 0 \\ 0, & \text{sinon} \end{cases} \quad (7)$$

Avec $P_{dec_i}(c) = P[dec_i(c) \neq \emptyset]$ la probabilité qu'un nœud i puisse extraire au moins un paquet lors du décodage de c et $dec_i(c)$ qui est l'ensemble des paquets décodés par un nœud i à partir de c . $N(s)$ est le nombre de nœuds qui sont présumés par s comme étant capables de décoder c (et non le nombre total de voisins).

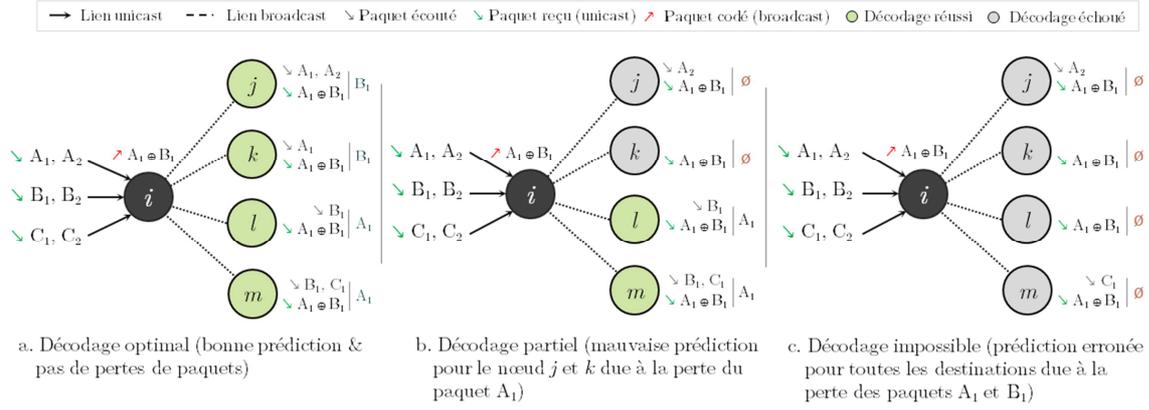


Figure 24. Illustration de l'impact du taux de pertes sur l'efficacité du codage réseau

Les travaux antérieurs dans le codage réseau ont principalement concentré leurs efforts dans l'amélioration de la décodabilité. [94] par exemple étudie l'impact de la taille de la file d'écoute des paquets sur l'efficacité du codage et [95] analyse celui de l'ordonnancement sur le rendement des codes opportunistes. Même si la décodabilité est souvent désignée dans la littérature comme l'efficacité du codage, elle ne suffit pas à affirmer que le choix d'un code est optimal. En effet, une décodabilité élevée ne signifie pas nécessairement un codage réseau plus fiable.

4.1.3. Innovativité

Par innovativité nous désignons la probabilité qu'un paquet codé c produise au moins un paquet utile chez toutes ses destinations. Cette valeur est définie dans la formule suivante :

$$Inn(c) = \begin{cases} \sum_{i=0}^{N(s)} \frac{P_{inn_i}(c)}{N(s)}, & N(s) \neq 0 \\ 0, & \text{sinon} \end{cases} \quad (8)$$

Avec $P_{inn_i}(c) = P[dec_i(c) \notin (Q_{in}(i) \cup Q_{ovh}(i))]$. $P_{inn_i}(c)$ est la probabilité qu'un paquet c contienne de l'information utile pour un nœud i , i.e., que le paquet extrait de c n'existe ni dans sa file des paquets reçus $Q_{in}(i)$, ni dans celle des paquets écoutés $Q_{ovh}(i)$.

Un paquet est considéré comme utile s'il n'a jamais été reçu ou écouté par son destinataire auparavant. Utilisée pour augmenter le rang de la matrice de décodage dans le codage linéaire, l'innovativité est souvent considérée comme triviale dans le codage opportuniste car les nœuds reçoivent des rapports de réception de leurs voisins. Pourtant, lorsque les défaillances de liens augmentent et que des rapports de réceptions sont perdus, les messages codés contiennent moins de paquets innovants ce qui dégrade les performances du processus de codage. Ainsi, avant qu'un nœud intermédiaire ne décide quel paquet et à partir de quel flux l'inclure dans le prochain code à envoyer, il doit savoir si ce paquet sera innovant après son décodage à sa (ses) destination(s). Ceci peut être garanti si ce nœud sait (ou estime) ce que ses voisins ont déjà reçu ou écouté. Notons que plus grande est la valeur de l'innovativité, plus grande doit être la probabilité pour qu'un paquet soit inclus dans le prochain code à transmettre.

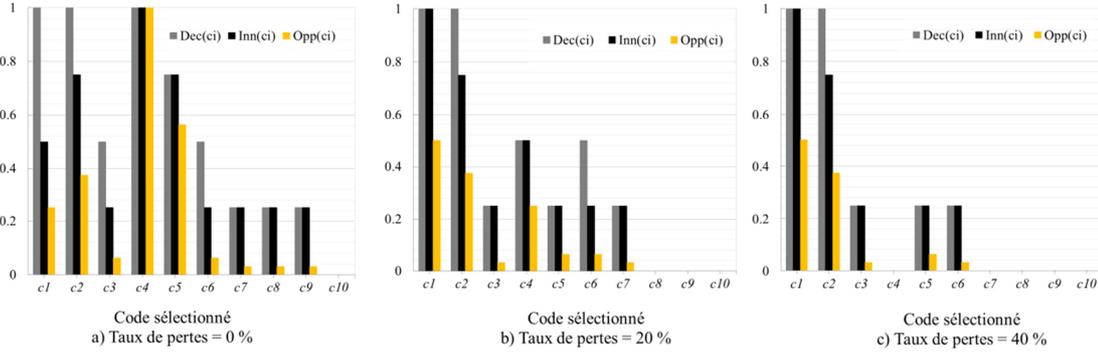


Figure 25. Illustration de l'impact du taux de pertes sur la fiabilité d'un code

L'exemple illustré dans la Figure 24 montre des situations où le même paquet codé différemment a plusieurs valeurs d'innovativité selon le taux de pertes.

Figure 24 (a) : $Inn(p_a) = 1$, lorsqu'il n'y a pas de pertes (0%).

Figure 24 (b) : $Inn(p_b) = \frac{1}{2}$, quand le paquet A_1 est perdu (20%).

Figure 24 (c) : $Inn(p_c) = 0$, quand A_1 et B_1 sont perdus (40%).

4.1.4. Optimalité

Cette valeur représente, pour un code donné, la conjonction de sa décodabilité et de son innovativité, i.e., un code c (une combinaison de paquets) est dit optimal si et seulement si ce dernier permet le décodage d'un nombre maximum de paquets et qu'il offre des valeurs maximales de décodabilité et d'innovativité. Un paquet codé peut en effet être décodable par toutes ses destinations et pourtant n'apporter aucune innovativité à aucune d'entre-elles, ou au contraire, il peut contenir des paquets innovants pour ses destinations, mais non décodables par ces dernières, affectant ainsi la fiabilité du codage. L'équation suivante donne une formalisation de l'optimalité d'un code quelconque c_i .

$$Opt(c_i) = Max \left[Dec(c_i) \times Inn(c_i) \times \frac{N_{dec_i}}{N_{pos_i}} \right] \quad (9)$$

Où N_{dec_i} est le nombre de paquets distincts effectivement décodés à partir de c_i et N_{pos_i} le nombre de paquets potentiellement décodables à partir de c_i . Améliorer l'optimalité d'un code consiste donc en la maximisation d' $Opt(c_i)$, c'est à dire permettre à l'ensemble de paquets sélectionnés d'être décodé par un maximum de destinataires, et que ces derniers puissent en extraire de l'information utile.

Par souci de clarté, nous énumérons dans le tableau 4 différentes valeurs de décodabilité et d'innovativité pour $s = \{c_i\}$, un ensemble de 10 stratégies de codage établies par un nœud intermédiaire quelconque. Ces valeurs correspondent à la topologie présentée dans l'exemple canonique de la Figure 24, avec L_0 , L_1 et L_2 qui représentent les taux de pertes des configurations illustrées dans les Figure 24 (a), (b) et (c). La Figure 25 révèle un constat important, à savoir, l'impact du taux de pertes sur la fiabilité du codage réseau. En effet, on note une baisse moyenne de plus de 50% de la valeur de Opt lorsque le taux de pertes atteint 20%. Ceci suggère que l'application de décisions de codage qui ne prennent

pas en compte les conditions du médium peut s'avérer inefficace car la meilleure stratégie de codage n'est pas forcément de transmettre le code avec l'optimalité *idéale* sans pertes.

Tableau 4. Taux de pertes et optimalité du codage

Possibilité de codage	$Dec(c_i)$			$Inn(c_i)$		
	L_0	L_1	L_2	L_0	L_1	L_2
$c_1 = A_1$	1	1	1	$\frac{1}{2}$	1	1
$c_2 = A_2$	1	1	1	$\frac{3}{4}$	$\frac{3}{4}$	$\frac{3}{4}$
$c_3 = A_1 \oplus A_2$	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$
$c_4 = A_1 \oplus B_1$	1	$\frac{1}{2}$	0	1	$\frac{1}{2}$	0
$c_5 = A_1 \oplus C_1$	$\frac{3}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{3}{4}$	$\frac{1}{4}$	$\frac{1}{4}$
$c_6 = B_1 \oplus C_1$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$	$\frac{1}{4}$
$c_7 = A_1 \oplus B_1 \oplus C_1$	$\frac{1}{4}$	$\frac{1}{4}$	0	$\frac{1}{4}$	$\frac{1}{4}$	0
$c_8 = A_1 \oplus A_2 \oplus B_1$	$\frac{1}{4}$	0	0	$\frac{1}{4}$	0	0
$c_9 = A_1 \oplus A_2 \oplus C_1$	$\frac{1}{4}$	0	0	$\frac{1}{4}$	0	0
$c_{10} = A_1 \oplus B_2 \oplus C_2$	0	0	0	0	0	0

4.1.5. Équité

Alors que le codage réseau améliore sans équivoque l'utilisation de la bande passante, il peut aussi pénaliser les flux selon la façon dont l'algorithme de sélection de paquets se comporte. Par exemple, si un nœud intermédiaire se concentre uniquement sur l'optimalité d'un code, il peut en découler une détérioration de l'équité par rapport aux différents flux de données qu'il se doit d'acheminer. En fait, en sélectionnant systématiquement des paquets à partir des mêmes flux de données à chaque opportunité de codage, le nœud peut retarder la transmission d'autres flux affectant ainsi considérablement la latence de certaines communications de bout-en-bout. Par conséquent, nous définissons $Fair_i(f_j)$ la probabilité que le prochain paquet codé c à transmettre à partir du nœud i contienne au moins un paquet du flux de données f_j .

$$\forall i \in N, j \in InFlows(i),$$

$$Fair_i(f_j) = P[c \in Q_{out}(i) \mid \exists p \in f_j, p \oplus c = c - p] \quad (10)$$

Où $Q_{out}(i)$ est l'ensemble des paquets à transmettre par le nœud i , $InFlows(i)$ l'ensemble des flux entrants f_j et p un paquet du flux f_j .

Il est à noter que lorsqu'un algorithme de codage effectue une sélection optimale de paquets, plus la valeur de $Fair_i$ est élevée pour un flux f_j , plus grand est le débit auquel il doit s'attendre. Garantir un accès équitable au médium est usuellement effectué par de simples mécanismes de gestion de files d'attente. Pourtant, dans un réseau codé, comme l'ordonnancement des paquets dépend de la stratégie de codage, l'équité des flux de données nécessite des solutions plus sophistiquées. En effet, si un simple système FIFO est souvent suffisant pour garantir un accès équitable à la file d'attente de transmission dans les réseaux *store-and-forward*, il est impossible à mettre en œuvre pour le codage réseau. Cela est dû au fait que dans un réseau codé, les paquets ne sont pas sélectionnés dans leur ordre d'arrivée, mais plutôt en fonction des opportunités de codage optimal. A notre connaissance, le sujet de l'équité dans le codage réseau n'est pas traité dans la littérature,

bien que cette métrique puisse considérablement dégrader la latence de bout-en-bout dans les systèmes sans fil.

4.1.6. Taux d'acheminement de bout-en-bout

Cette métrique, de la même manière que l'équité, concerne les communications de bout-en-bout. Il s'agit d'un critère classique d'évaluation de la fiabilité des réseaux de communication en général. Il implique que les algorithmes de codage doivent assurer un ratio minimum d'acheminement des paquets. L'équation suivante détaille ce ratio :

$$r(i, j) = \frac{\text{Nombre de paquets reçus par } j}{\text{Nombre de paquets transmis par } i} \quad (11)$$

Où i est la source et j la destination lors d'une session de communication de bout-en-bout entre les nœuds i et j . Notons que selon l'application du réseau, les protocoles peuvent être contraints d'assurer un taux d'acheminement minimum r afin de garantir une certaine qualité de service.

4.1.7. Discussion

Même si elles peuvent comporter quelques métriques similaires comme le taux d'acheminement de bout-en-bout ou l'équité inter-flux, la notion de fiabilité classique reste assez différente du concept de codage fiable introduit dans cette première partie. Notre analyse confirme cette affirmation et met en exergue la nécessité de prévoir des mécanismes spécifiques pour la fiabilisation des réseaux codés. Puisque le concept de codage fiable n'a pas encore été traité dans la littérature, nous présentons dans ce premier travail une série de solutions prévues pour améliorer la fiabilité des réseaux de capteurs sans fil basés sur le codage opportuniste. Par ailleurs, les exigences de ces algorithmes de codage en termes de fiabilité diffèrent non seulement en fonction de l'application du réseau, mais aussi selon les composantes spécifiques à l'algorithme, telles que son mécanisme d'inférence de topologie, les stratégies de codage ou encore le système de retransmission. Ainsi, les solutions que nous proposons se doivent d'être adaptées en tenant compte de ces composantes. Un exemple de cela est donné avec ReCoZi, un mécanisme de codage fiable qui étend le codage opportuniste ZigBee (CoZi) proposé dans le chapitre précédent.

4.2. Codage ZigBee Fiable – ReCoZi

Afin d'effectuer un acheminement fiable des paquets, des mécanismes de retransmission sont souvent nécessaires dans les réseaux sans fil en général et dans les réseaux de capteurs en particulier. Ces approches ont montré une efficacité remarquable dans plusieurs protocoles de communication (802.15.4 [20], ZigBee [15], B-Mac [96], etc.). Toutefois, lorsque le codage réseau est utilisé, ces mécanismes de retransmission par acquittement peuvent devenir insuffisants en raison des caractéristiques spécifiques des algorithmes de codage. En effet, dans ces systèmes les paquets ne peuvent pas être simplement acquittés à chaque fois qu'ils sont reçus ou écoutés, parce que des opérations supplémentaires doivent être effectuées pour décider si le paquet est exploitable ou pas.

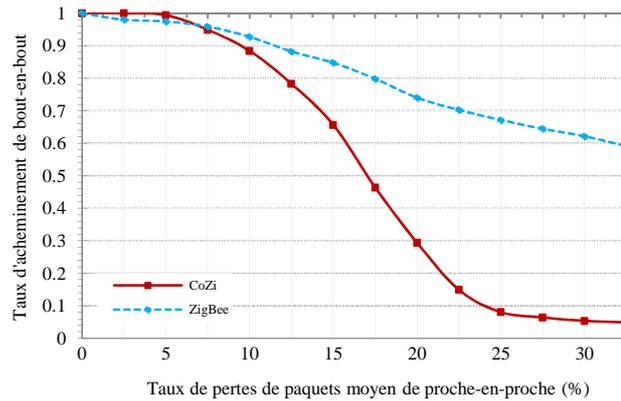


Figure 26. Taux d'acheminement vs. taux de pertes

Il est clair grâce à la Figure 26, qui évalue via simulation comment le taux de pertes impacte CoZi, que notre mécanisme de codage opportuniste souffre d'un déclin de performances bien plus rapide que ZigBee lorsque le canal radio devient erratique. Cela est dû au fait que le système d'acquiescement de CoZi est conçu pour assurer une fiabilité classique des communications et non pour garantir un codage fiable tel que défini dans la section précédente. Par exemple, un paquet codé qui a été décodé par un sous-ensemble de ses destinataires mais pas par le prochain saut sur sa route peut être considéré comme acheminé (car les paquets ACK confirment sa réception) et n'est donc jamais retransmis provoquant un chemin *sans issue* pour un flux de paquets donné. Avec l'augmentation du taux de pertes, ce phénomène devient de plus en plus fréquent menant à la prolifération de ces chemins sans issue et affectant de façon drastique le taux d'acheminement du réseau. De plus, CoZi effectue souvent des décisions de codage erronées à cause d'une mauvaise inférence de topologie causée par la perte de paquets. En fait, les nœuds codeurs ne possèdent aucun moyen pour savoir si un paquet a été perdu à cause des conditions du médium ou d'une mauvaise stratégie de codage, ce qui affecte la décodabilité et l'innovativité des codes sélectionnés à chaque transmission.

Ainsi, nous considérons comme cas d'étude et durant le reste de ce chapitre, le problème du codage fiable dans les réseaux de capteurs ZigBee basés sur la couche MAC IEEE 802.15.4 et le codage opportuniste CoZi. Nous proposons ReCoZi, une série de solutions pour faire face aux défaillances de codage provoquées par l'instabilité du canal de communication. Basé sur notre définition du codage fiable, nous montrons comment grâce à un mécanisme d'acquiescement adapté et à une plus grande connaissance de l'état de liens, ReCoZi conserve à la fois le gain de performance du codage réseau et la fiabilité des communications lors de la présence de pertes.

4.2.1. Système d'acquiescement des paquets

Dans la section précédente nous avons défini une série de paramètres pour évaluer la fiabilité des algorithmes de codage réseau. En conséquence, et afin d'atténuer l'impact que peut avoir un canal de communication non fiable sur l'efficacité du codage, nous présentons un système de retransmission/recodage par acquiescement adapté aux algorithmes de codage opportuniste dans les réseaux ZigBee.

Algorithme 2. Réception d'un paquet par un routeur A

```

1: for each incoming packet  $p$  in  $Q_{in}(A)$  do
2:   if  $p$  is non-coded then
3:     if  $A$  is dest. of  $p$  and is not the final hop then
4:        $A$  adds  $p$  to its next echo-ACK packet
5:     else
6:       if  $A$  is dest. of  $p$  and is the final hop then
7:          $A$  sends an ACK to the sender
8:       else /*  $A$  overheard  $p$  */
9:          $p$  is stored in  $Q_{ovh}(A)$ 
10:      end if
11:    else /*  $p$  is coded */
12:      if  $p$  is decodable then
13:         $p_{dec} = Decode(p)$ 
14:        if  $A$  is dest. of  $p_{dec}$  and is the final hop then
15:           $A$  sends an ACK to the sender
16:        else
17:          if  $A$  is dest. of  $p_{dec}$  and is not the final hop then
18:             $A$  adds  $p_{dec}$  to its next echo-ACK packet
19:          else /*  $A$  overheard  $p$  */
20:             $p_{dec}$  is stored in  $Q_{ovh}(A)$ 
21:             $A$  adds  $p_{dec}$  to its next echo-ACK packet
22:          end if
23:        else /*  $p$  is not decodable */
24:           $p$  is dropped from  $Q_{in}(A)$ 
25:        end if
26:      end if
27:    end for

```

Au lieu d'utiliser le même système d'acquittement pour l'ensemble des paquets, nous proposons de modifier CoZi de sorte que seules les réceptions importantes soient explicitement acquittées (i.e. les paquets qui ont atteint leur destination finale) tandis que celles au niveau des nœuds intermédiaires ne sont que passivement acquittées en utilisant un système d'acquittement par écho⁴ [97]. Ainsi, afin d'éviter toute détérioration significative de la fiabilité due à la perte de paquet d'acquittement, nous ajoutons un champ *echo-feedback* dans les paquets codés sortants qui permet d'informer les nœuds voisins à la fois de tout paquet reçu ou écouté et aussi des paquets éventuellement perdus ou impossible à décoder. L'algorithme 2 décrit plus en détail ce processus de retransmission. Les lignes 4, 18 et 21 montrent que l'acquittement par écho est utilisé à chaque fois qu'un paquet est décodé par un nœud intermédiaire (qu'il soit reçu ou écouté). Cette approche passive d'acquittement consiste à valider la réception d'un paquet par sa destination à un saut en remplissant le champ ACK dans l'entête du paquet suivant à transmettre, à l'exception du dernier nœud sur la route du paquet où un acquittement explicite doit être généré (cf. lignes 7 et 15).

⁴ L'acquittement par écho est possible grâce à la propriété omnidirectionnelle du broadcast dans les liens sans fil

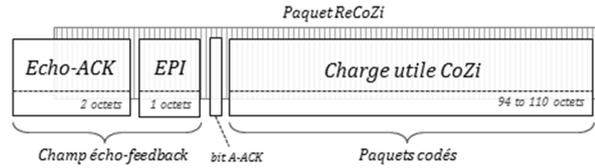


Figure 27. Champs supplémentaires nécessaires à l’acquittement par écho

Comme illustré sur la Figure 27, nous faisons usage du concept de *piggybacking* pour notifier les nœuds transmetteurs du prochain paquet attendu par leurs voisins. En effet, grâce à l’incorporation dans l’entête ReCoZi du champ EPI (Expected Packet Id) qui contient le numéro de séquence du prochain paquet attendu par le nœud, il est possible pour un nœud intermédiaire d’ajouter l’identifiant du prochain paquet qu’il s’attend à recevoir. Ainsi, ses voisins pourront effectuer des décisions de codage plus précises en vérifiant ce champ et en intégrant dans leur code le paquet manquant, s’il est disponible dans leur file d’attente. Par exemple, considérons trois nœuds intermédiaires i , j et k sur la route d’un paquet codé p . Après réception de p_i par j et avant qu’il ne le relaye à k si le prochain paquet attendu par le nœud j est p_{i+1} , le champ EPI contiendra le numéro de séquence de p_{i+1} de sorte que le nœud émetteur i soit au courant en écoutant le canal que p_{i+1} n’est pas encore reçu par le nœud j .

Cette technique permet d’effectuer une meilleure inférence de topologie locale, améliorant la décodabilité et l’innovativité des codes sélectionnés. L’utilisation du *piggybacking* permet aussi la diminution de la surcharge réseau induite par l’envoi de paquets ACK à chaque phase de décodage. En conséquence, les nœuds codeurs peuvent constamment savoir quand un paquet est perdu, et détecter quels codes n’ont pas pu être décodés par quels nœuds, ce qui permet une meilleure gestion des retransmissions et du recodage des paquets, et qui résout en partie le problème des chemins sans issue dans les communications de bout-en-bout. Par ailleurs, un vrai gain sur le nombre de transmissions peut être constaté car beaucoup moins de paquets sont dupliqués inutilement lors de la présence d’un fort taux de pertes (pertes de paquets de données ou pertes d’ACK).

4.2.2. Acquittement à état de lien

La préservation du taux d’acheminement des données dans les réseaux codés lors de la présence de pertes non négligeables ne peut se faire que grâce à des mécanismes supplémentaires spécialement conçu pour le codage réseau. Cependant, lorsque les performances du codage opportuniste sont satisfaisantes parce que le canal est fiable, il n’est nullement nécessaire de recourir aux mécanismes d’acquittement décrit plus haut, et cela pour optimiser l’utilisation de la bande passante disponible. Par contre, à mesure que la fiabilité du réseau se dégrade et que l’inférence de topologie devient moins précise, les mécanismes d’acheminement doivent être enrichis avec des méthodes d’acquittement qui assurent un codage réseau plus fiable dans un environnement perturbé. L’utilisation du champ RSSI⁵ dans la trame 802.15.4 combiné au taux d’erreur mesuré (BER) à la

⁵ Received Signal Strength Indication, ou indicateur de la puissance de signal reçu

réception permet à chaque nœud intermédiaire d'évaluer localement la qualité de ses liens. Grâce à ces valeurs, un nœud peut facilement déclencher l'utilisation des mécanismes d'acquittement avancés afin de faire face à une éventuelle dégradation du canal de communication. Dans le cas de ReCoZi, cela est effectué en attribuant la valeur 1 au bit A-ACK (cf. Figure 27) à chaque fois que les valeurs du *RSSI* et du *BER* descendent sous un seuil prédéfini, signifiant aux autres nœuds que les mécanismes d'acquittement avancés sont activés. Bien évidemment, afin d'éviter toute activation fortuite de ce mode, les nœuds calculent une valeur moyenne pour un nombre N de paquets reçus avant de vérifier si le seuil T a été atteint. L'équation suivante détaille ce procédé :

$$\left[c_1 \cdot \left(\frac{\sum_N RSSI_i}{N} \right) + c_2 \cdot \left(\frac{\sum_N BER_i}{N} \right) \right] < T \Rightarrow A_ACK = 1 \quad (12)$$

Où c_1 et c_2 sont des coefficients de pondération utilisés pour donner plus ou moins d'importance soit au *RSSI* ou au *BER* dans l'estimation de l'état de liens. Grâce à cette solution simple, ReCoZi fournit des estimations des conditions du canal de communication plus réactives de sorte que la fiabilité soit maintenue en utilisant les mécanismes d'acquittement aussi longtemps que le medium sans fil reste perturbé.

4.2.3. Gestion équitable des files d'attente de codage

Comme mentionné précédemment, pour être considéré comme fiable, un mécanisme de codage réseau doit fournir un taux d'acheminement élevé ainsi qu'une équité entre les différents flux de données. Nous avons pu noter lors de l'évaluation de CoZi un écart important entre les délais propres à chaque communication de bout-en-bout, et cela parce que les mécanismes de codage opportuniste ne font aucune distinction entre les flux puisque souvent l'efficacité du codage prévaut. En effet, il arrive fréquemment que l'algorithme d'ordonnancement sélectionne des paquets du même sous-ensemble de flux, en négligeant d'autres communications de bout-en-bout (surtout lors de la présence de pertes) menant à l'expiration de la durée de vie du paquet en nombre de saut (*Radius* dans ZigBee), ce qui affecte clairement le taux d'acheminement moyen du réseau. Pour faire face à ce problème, ReCoZi comprend un mécanisme de gestion de file d'attente de codage grâce auquel chaque nœud supervise sa file de transmission, vérifiant si les paquets d'un flux donné sont restés une période de temps déterminée sans jamais être intégrés dans un code et de ce fait, transmis. Dans ce cas, le nœud en question suspend toutes les opérations de codage en cours afin d'envoyer ces paquets lors de la prochaine opportunité de transmission. Cette solution évite d'appliquer des stratégies de codage inéquitables qui réduisent le taux d'acheminement de certaines communications de bout-en-bout.

4.2.4. Analyse de performance

Dans cette section nous évaluons la fiabilité du codage ReCoZi dans différentes conditions radio. Nous quantifions l'optimalité de l'algorithme et le taux d'acheminement qu'il offre. Puis nous comparons ses performances (délai et débit) avec CoZi et un routage ZigBee standard. Enfin, nous montrons la surcharge de paquets de contrôle induite par les mécanismes de retransmissions propres à ces trois systèmes.

Tableau 5. Paramètres de simulation

Durée de simulation	600 s
Topologie	Hierarchique (ZigBee) BO = 15 SO = 15
Routage	ZigBee cluster-tree routing
Protocole de couche MAC	IEEE 802.15.4
Capacité	250 kbps
Modèle de propagation	Lognormal shadowing (M = 4.5 dB)
Modèle d'atténuation	Rayleigh/ Longley-Rice
Rayon de transmission	~35 mètres
Type de trafic	CBR variable
Période de mise en tampon	500 ms
Taille des paquets	116 octets
Nombre de routeurs	80%
Nombre de nœud terminaux	19%

4.2.4.1. Environnement de simulation

Nous utilisons la plateforme de simulation Qualnet 4.5 [84] pour évaluer notre solution. Où ReCoZi, CoZi, ZigBee et la couche MAC 802.15.4 sont implémentés. Nous considérons un réseau de 100 nœuds uniformément positionnés dans une surface de 150x150m. Le coordinateur ZigBee est placé au centre du réseau et le reste des nœuds sont soit des routeurs soit des nœuds terminaux. Chaque communication à un débit qui varie de 25 à 100 *paquets/s*. Tous les paramètres de simulation sont résumés dans le tableau 5.

Afin d'évaluer la fiabilité de ReCoZi et de le comparer à CoZi et ZigBee, nous appliquons des taux de pertes différents au réseau et calculons les résultats sur la base des métriques suivantes : l'optimalité du codage, le taux d'acheminement des paquets, le délai, le débit et la surcharge du réseau.

4.2.4.2. Optimalité

Dans un premier temps, nous calculons l'optimalité par rapport au taux de pertes de paquets. Ce calcul se fait en utilisant la décodabilité, l'innovativité et le nombre de paquets décodés pour un code donné à chaque nœud intermédiaire. Nous pouvons constater sur la Figure 28 que ReCoZi améliore considérablement l'optimalité des codes sélectionnés, même lorsque le canal devient très erratique. Cela est dû à l'utilisation des mécanismes de retransmission décrits plus haut, où les décisions de codage sont plus précises grâce notamment à l'activation de l'acquiescement par écho qui permet une inférence de topologie plus réaliste du nœud en question. Il est à noter que nous ne calculons pas la valeur de *Opt* pour ZigBee car aucun codage n'est effectué dans ce cas.

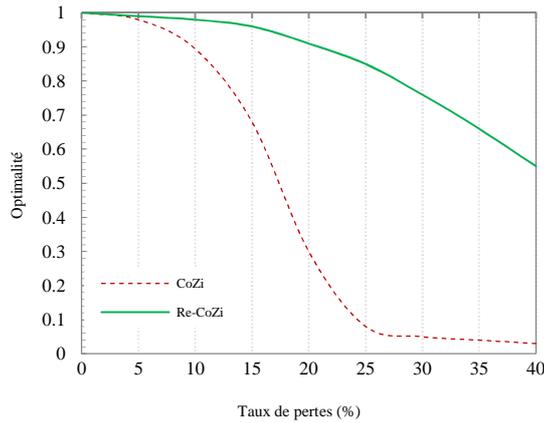


Figure 28. Valeur de l'optimalité et taux de pertes de paquets

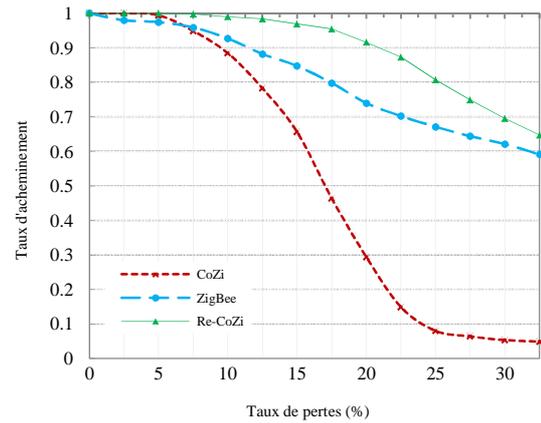


Figure 29. Taux d'acheminement et taux de pertes de paquets

4.2.4.3. Taux d'acheminement

Nous pouvons voir sur la Figure 29 que le taux d'acheminement fourni par ReCoZi surpasse à la fois ZigBee et CoZi à mesure que le canal subit plus de pertes. En fait, avec l'activation des méthodes d'acquiescement avancés, ReCoZi effectue des décisions de codage plus précises qui limitent le phénomène des chemins sans issue et permettent de retransmettre efficacement les paquets codés perdus en vérifiant le champ Echo-ACK des paquets reçus et écoutés. Notons également que ReCoZi fournit un taux d'acheminement plus élevé que celui offert par ZigBee. Ceci est une conséquence directe de l'utilisation du codage réseau au niveau des nœuds intermédiaires du chemin, puisque l'utilisation de la bande passante est optimisée avec plusieurs paquets relayés en une seule transmission. Néanmoins, lorsque le taux de pertes de paquets atteint un certain seuil et que les opportunités de codage se raréfient, les valeurs du taux d'acheminement constatées avec ReCoZi et ZigBee convergent ce qui est normal car les deux systèmes fonctionnent similairement dans ces conditions.

4.2.4.4. Délai de bout-en-bout et équité

La Figure 30 montre que la latence moyenne des communications lors de l'utilisation de ReCoZi est légèrement supérieure à celle de ZigBee mais toutefois nettement moins importante que celle offerte par CoZi et cela lorsque le réseau endure des pertes élevées de paquets. Si ReCoZi assure un délai de bout-en-bout inférieur à CoZi, c'est parce que via sa gestion différente de la file de codage, il limite l'apparition de chemins sans issue causés par les systèmes d'acquiescement inadéquats au codage réseau et qu'il assure plus d'équité que son homologue : ces deux problèmes sont les causes directes de la présence de délais supplémentaires dans les réseaux codés. En effet, ils retardent significativement certains flux de paquets affectant la latence moyenne de bout-en-bout dans le réseau. Il est aussi à noter que dans un contexte plus idéal (sans pertes), ReCoZi conserve les mêmes performances que CoZi.

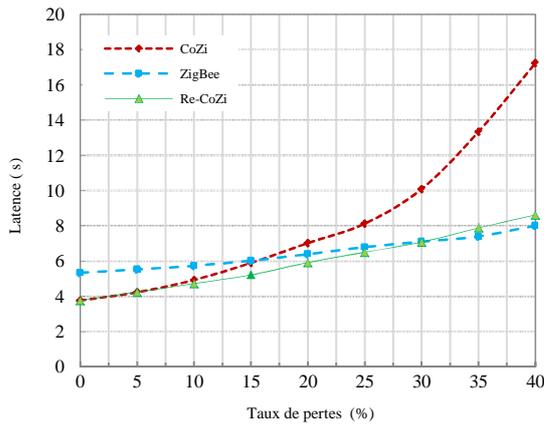


Figure 30. Latence et taux de pertes

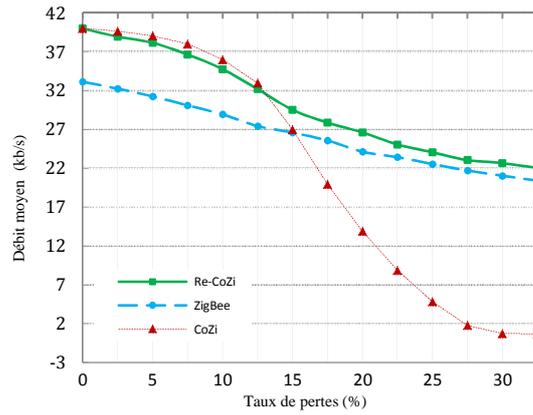


Figure 31. Débit et taux de pertes

4.2.4.5. Débit effectif

Le débit effectif⁶ global est augmenté grâce à l'amélioration de l'optimalité du codage. En fait, comme le montre la Figure 31, ReCoZi évite que le débit ne s'effondre comme c'est le cas dans CoZi, grâce à l'activation de l'acquittement à état de lien qui permet d'éviter les mauvaises décisions de codage. De plus, nous remarquons que le débit global obtenu par ReCoZi est presque identique à celui de ZigBee lorsque le taux de perte dépasse 30%. Cela est dû au fait que l'optimalité des codes sélectionnés diminue drastiquement avec l'augmentation des pertes de paquets, ce qui réduit l'ensemble des opportunités de codage possibles et donc ne permet d'envoyer, dans la plupart des cas, que des paquets non codés.

4.2.4.6. Surcharge réseau

La Figure 32 illustre la croissance du nombre de paquets de contrôle nécessaires pour ZigBee, CoZi et ReCoZi. Cette surcharge réseau inclut les messages d'acquittement explicites et les paquets de feedback nécessaires à l'inférence de topologie. Elle représente la fraction de la bande passante utilisée par ces paquets de contrôle et les en-têtes 802.15.4. Puisque ReCoZi n'utilise que peu de paquets d'acquittement explicites, il requiert beaucoup moins de paquets de contrôle que CoZi et ZigBee. Cependant, comme il se base sur des en-têtes spéciaux au niveau de la trame 802.15.4, la charge utile des paquets ReCoZi est plus petite que dans CoZi. Ceci explique le fait que le débit de ReCoZi est légèrement en deçà de celui de CoZi dans la Figure 31 (dans les cas où le taux de pertes est inférieur à 12.5%).

Il est aussi à noter que la réduction de la surcharge réseau est liée à la réduction du nombre global de transmissions grâce au codage réseau, ce qui est en soi une amélioration importante dans la mesure où dans les réseaux de capteurs sans fil l'utilisation du module radio doit être minimisée.

⁶ Le débit effectif est calculé au niveau des nœuds destinataires, il n'inclut pas les retransmissions, les paquets ACK et les paquets redondants.

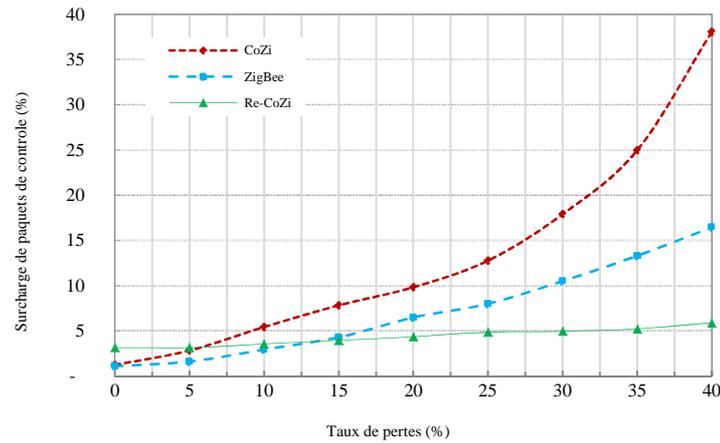


Figure 32. Surcharge réseau et taux de pertes

4.3. Conclusion

Dans ce chapitre, nous montrons que les méthodes de fiabilisation inspirées des systèmes *store-and-forward* ne sont pas adaptées au codage réseau en raison de ses spécificités (ordonnancement, écoute de paquets, décodabilité, etc.). Afin de résoudre ce problème, nous introduisons la notion de codage fiable : une série de métriques qui permettent la conception de systèmes de codage réseau plus robustes, i.e. qui garantissent un traitement équitable des flux de données, un taux d'acheminement optimisé et un ordonnancement efficace des paquets quel que soit l'état du médium sans fil. Afin notamment de démontrer l'intérêt de ces métriques, nous proposons ReCoZi, un mécanisme de codage fiable pour les réseaux de capteurs ZigBee. Ce code opportuniste utilise une méthode de retransmission basée sur l'état des liens afin de garantir des communications codées plus tolérantes aux pertes. Enfin, nous évaluons les performances et la robustesse de ReCoZi et montrons qu'il pallie à la problématique de vulnérabilité aux pertes inhérente aux protocoles de codage réseau sans fil en général.

Les résultats de notre étude de performance et de fiabilité montrent que ReCoZi répond à l'objectif fixé qui est de garder la valeur ajoutée du codage réseau même lorsque le canal de communication devient erratique. Grâce à de simples mécanismes de retransmissions adaptés aux communications codées, ReCoZi permet d'effectuer un codage robuste plus tolérant aux pertes, avec une faible surcharge réseau et un accès équitable au médium. En gardant la valeur ajoutée du codage réseau, il permet aussi de réduire le nombre de transmissions, de mieux exploiter la bande passante disponible et ainsi de réduire les délais de bout-en-bout.

Cet effort dans l'amélioration de la conception, de la validation et de l'évaluation des techniques de codage fiables s'inscrit comme une première formalisation qui permet aux concepteurs de systèmes de codage réseau de mieux appréhender l'étude de performance et de fiabilité de leur solution. Grâce à cette formalisation, il devient plus facile de quantifier à la fois les apports et les défaillances des protocoles de communication basés sur le codage réseau sans fil.

Toutefois, la fiabilisation des mécanismes de codage réseau n'est pas le seul problème auquel les réseaux de capteurs sans fil sont confrontés. L'adaptabilité des protocoles de communication dans ces architectures, qu'ils soient codés ou pas, reste l'élément clé qui permet d'établir leur viabilité dans un environnement aussi contraint que celui des capteurs sans fil. En effet, un protocole de communication pour les réseaux inter-machines n'est efficace que s'il est fiable, qu'il produit des performances élevées et qu'il peut fonctionner en consommant le moins de ressources possible au niveau des nœuds qui composent le réseau.

Ainsi, le chapitre suivant continue dans cet effort d'adaptation du codage réseau aux contraintes des réseaux de capteurs sans fil. Nous y proposons un système opportuniste de codage par index pour des réseaux très contraints et étudions son rendement par deux analyses, la première théorique qui prouve l'efficacité de son algorithme de codage, et la deuxième empirique, qui montre son adaptabilité dans un vrai réseau de capteurs sans fil.

