

Résumé

Ces dernières décennies, les réseaux sans fil ont été déployés à un rythme remarquable dans différents contextes : les bureaux d'entreprises, les universités, les hôtels, etc. La prolifération d'équipements sans fil légers ont fait de ces réseaux la première solution pour assurer un accès internet aux utilisateurs à tout moment et en tout lieu. L'une des premières applications des communications sans fil fut les réseaux cellulaires. Même si le cœur du réseau est doté d'une infrastructure permettant une communication filaire, la transmission entre les points d'accès et les équipements des utilisateurs reste sans fil.

Installer une infrastructure est couteux et parfois impossible tel que dans un contexte post séisme. Par conséquent, une nouvelle catégorie de réseau apparue, les réseaux sans fil multi-sauts où les nœuds s'auto-configurent et construisent un réseau par eux même. Un nœud source envoie l'information directement au nœud destination s'il est situé dans sa zone de réception, sinon, l'information est transmise par d'autres nœuds, saut par saut, jusqu'à atteindre le nœud destination. En général, il y a différents types réseaux sans fil multi-sauts:

Le réseau MANET (*Mobile Ad hoc NETWORK*) configure les nœuds et s'adapte à leur mobilité pour garantir une connexion continue. Un exemple d'application de ce genre de réseaux est l'opération de sauvetage après une catastrophe naturelle. Cependant les réseaux MANET peuvent aussi être utilisés pour connecter deux ou plusieurs ordinateurs.

Le réseau véhiculaire (VANET : *Vehiculair Ad hoc NETWORK*) permet la communication entre les véhicules ayant une mobilité dynamique. L'objectif est d'échanger les informations parmi les utilisateurs et de récupérer les informations nécessaires pour gérer le trafic.

Le réseau de capteurs (WSN : *Wireless Sensor Network*) est utilisé pour collecter des mesures telles que la température et la vitesse. Les nœuds

collaborent afin d'acheminer l'information à un nœud particulier appelé puits (*Sensor gateway*). Ce type de réseau est utilisé dans de nombreuses applications telles que le suivi de l'environnement, le contrôle de climat en milieu fermé, la surveillance, la vérification de menaces et les alarmes intelligentes.

Le réseau mesh (WMN : *Wireless Mesh Network*) étend la couverture réseau à une zone donnée. Il est principalement utilisé quand l'extension de l'infrastructure d'un réseau filaire n'est pas réalisable pour des raisons financières ou géographiques. Des nœuds sans fil particuliers collaborent pour établir un réseau dans lequel l'accès à un réseau externe comme Internet, se fait via une passerelle (nœud jouant le rôle d'un point d'accès pour Internet). Puis, les utilisateurs utilisent un réseau Mesh comme cœur de réseau pour s'y connecter.

Le canal de transmission sans fil présente un taux de perte plus élevé qu'en filaire, de plus, les équipements sans fil ont une autonomie limitée. Les réseaux sans fil multi-sauts sont par conséquent confrontés à divers problèmes: un débit réduit, un délai plus long (latence), une sécurité moins accrue et une puissance d'énergie réduite. Par ailleurs, les applications déployées sur les équipements sans fil sont de plus en plus nombreuses: Voix sur IP (VoIP), vidéo à la demande, partage de fichiers, etc. Ces applications requièrent une haute qualité de voix, images, vidéos et communications multimedia. C'est pourquoi, il est nécessaire d'améliorer les caractéristiques de ces réseaux.

Assurer une communication efficace et fiable tout en garantissant une Qualité de Service (QoS) est un réel défi au vu de la nature des transmissions sans fil. La QoS est la possibilité de garantir un certain niveau de performance aux utilisateurs des applications. En réseau, cela consiste à satisfaire les exigences de l'application sur certains aspects de la connexion tels que le débit, le taux de perte et le délai. Beaucoup d'efforts ont été faits pour améliorer les performances des réseaux sans fil en termes de bande passante, consommation d'énergie, compression de données et sécurité.

Le canal de transmission partagé réduit la qualité de réception et la quantité des informations correctement reçues. Si, par exemple, deux nœuds a et b transmettent des données au nœud c en même temps, le nœud c ne peut pas déchiffrer et comprendre l'information envoyée. Ce phénomène est appelé interférence. Dans ce cas, la bande passante des liens $a \rightarrow c$ et $b \rightarrow c$, c'est-à-dire, la quantité de trafic qui peut être envoyée en un laps de temps via ces liens, est utilisée de façon non productive puisqu'une retransmission séquentielle est nécessaire. Certaines solutions essaient de prévenir les interférences en (i) choisissant un ordonnancement approprié parmi les liens, (ii) sélectionnant des routes qui réduisent les interférences ou (iii) adaptant la modulation du signal d'un nœud et de nœuds à proximité (ses voisins) comme c'est le cas pour l'OFDMA.

Une manière surprenante pour éviter l'interférence est de la considérer comme un phénomène positif et de surcroît, en profiter pour réduire la consommation de bande passante. En effet, prenons l'exemple du réseau "Alice et Bob". Dans ce scénario, il y a un flux d'Alice vers Bob et un autre de Bob vers Alice. Comme Alice n'est pas dans la zone de réception de Bob, un nœud intermédiaire (relai) est utilisé. Traditionnellement quand le relai reçoit le paquet de données P_1 d'Alice et P_2 de Bob, il transmet simplement P_1 à Bob et P_2 à Alice. Ce scénario nécessite quatre transmissions. Une nouvelle approche permet au relai de combiner de façon intelligente les paquets en utilisant par exemple un OU exclusif (XOR) et de diffuser en broadcast le

paquet résultant $P_1 \oplus P_2$. Ensuite, Alice décode le paquet combiné en

effectuant un XOR avec le paquet qu'elle avait préalablement envoyé, c'est-à-

dire, $P_1 \oplus (P_1 \oplus P_2)$ et récupère ainsi le paquet original P_2 . De façon similaire,

Bob décode $P_2 \oplus (P_1 \oplus P_2)$ et récupère P_1 . Ainsi, seulement trois transmissions

ont été nécessaires. Cette approche permettant aux nœuds intermédiaires de combiner les paquets au lieu de les transférer sans traitement est appelée « codage réseau ».

Nous avons expliqué ici un scénario très simple du codage réseau. Un scénario plus sophistiqué est de combiner plus de deux paquets. Pour l'opération de codage, quelques coefficients sont nécessaires. Ces derniers sont prélevés à partir d'un espace fini. Par exemple, en combinant k paquets P_1, P_2, \dots, P_k en utilisant k coefficients $C_1, C_2 \dots C_k$, le paquet combiné est alors $C_1.P_1 + C_2.P_2 + \dots + C_k.P_k$ et sa taille est égale à la taille d'un paquet traditionnel, des k identifiants de paquets et des k coefficients. L'overhead généré est considéré comme négligeable comparé aux avantages du codage réseau, il a été estimé à environ 3% dans la littérature. Nous rappelons que combiner des paquets peut se faire en codage spatial (à travers les liens) ou temporel (coder un paquet avec des paquets précédents).

Nous reprenons (Chapitre 2) l'état de l'art sur le codage réseau. Nous décrivons d'abord ses aspects théoriques et nous introduisons, étape par étape, de nouvelles notions qui ont permis la mise en pratique du codage réseau. Quelques exemples d'implémentations sur différents types de réseaux (mesh, WSN, P2P, cellulaire, etc.) sont donnés pour illustrer les applications du codage réseau. Après avoir étudié les différents schémas du codage réseau, nous choisissons les plus appropriés aux applications ciblées (unicast, multicast, multi-sources multi-destinations, sensibles au délai, sensibles à la bande passante).

En introduisant le nouveau paradigme de combiner les données à travers les nœuds, le codage réseau présente de nombreux domaines d'application. En effet, il augmente le débit dans les réseaux mesh, réduit la

consommation d'énergie dans les réseaux de capteurs, augmente la capacité des réseaux de diffusion multi-sources multi-destinations, améliore la disponibilité des données pour le transfert de fichiers dans les réseaux P2P (pair à pair) et CDN (Content Digital Networks) et renforce la robustesse des réseaux en termes de sécurité.

Dans cette thèse, nous nous intéressons à l'application du codage réseau pour garantir la QoS dans les réseaux sans fil multi-sauts. Pour atteindre cet objectif, nous nous concentrons sur trois points principaux : (i) la gestion des interférences pour optimiser la consommation de la bande passante, (ii) l'adaptation des paramètres du codage réseau pour améliorer les performances du réseau telles que le débit et le délai et (iii) l'augmentation de la capacité en utilisant un nouveau schéma de réseau. Ces trois idées sont détaillées dans ce qui suit.

Optimisation de la bande passante

L'objectif initial du codage réseau était de réduire le nombre transmissions et donc la consommation de la bande passante dans le réseau. Cependant, au lieu d'appliquer le codage après avoir routé (choisi les chemins) des flux, quelques approches le considèrent durant la phase de routage afin d'optimiser la bande passante. D'autres algorithmes de routage atteignent cet objectif en minimisant les interférences. Ces dernières ont impact néfaste sur les performances du réseau.

Dans un réseau sans fil, le canal de transmission est partagé par plusieurs liens, par conséquent, un nouveau flux est sujet à des interférences inter-flux engendrées par les flux existants et des interférences intra-flux causées par le flux lui-même. Si les interférences ne sont pas prises en compte durant le routage, la bande passante réelle du flux accepté peut varier en fonction de la charge du réseau. Lorsqu'une interférence survient, une retransmission peut être nécessaire et dans ce cas, the bande passante décroît. Aussi, quand un flux exige une bande passante et qu'il y a réservation de la même quantité de bande passante sur chacun des liens du chemin, ceci

peut s'avérer insuffisant. En effet, deux liens successifs, par exemple, ne peuvent transmettre en même temps et par conséquent leurs bandes passantes respectives sont divisées par deux. En définitive, le flux obtient une bande passante inférieure à celle qu'il a exigée, la QoS de service n'est donc pas respectée.

Pour calculer les routes minimisant l'impact des interférences, une méthode communément utilisée consiste à utiliser un modèle basé sur les cliques. Elle retrouve exactement le meilleur chemin en prenant en compte les interférences et donne ainsi une borne supérieure de la capacité du réseau. Au vu des avantages de codage réseau, il est intéressant de le combiner avec la considération des interférences durant la phase de routage. Ceci prendrait en compte l'impact des interférences et considèrerait en même temps un scénario de codage impliquant plus d'un flux comme dans l'exemple d'Alice et Bob.

- Dans ce contexte, nous améliorons (Chapitre 3) un algorithme existant appelé ROCX (*Routing with Opportunistically Coded Exchanges*) en introduisant la notion d'interférence. ROCX utilise la programmation linéaire pour router les flux tout en considérant le schéma de codage d'Alice et Bob. En d'autres termes, il effectue un reroutage des flux pour les faire croiser autant que possible minimisant ainsi la consommation de la bande passante. Par ailleurs, pour la gestion des interférences, nous améliorons le modèle basé sur les cliques et l'appliquons dans notre nouvel algorithme, I2ILP (*Intra-flow Interference aware Linear Programming*), qui utilise la programmation linéaire pour trouver le chemin approprié d'une source à une destination en considérant l'impact des interférences intra- et inter-flux. En utilisant I2ILP, quand un flux est admis dans le réseau, la bande passante qui lui est allouée est maintenue stable et la QoS garantie. Cependant, le chemin alloué au flux ne change pas, c'est-à-dire que la notion de reroutage n'est pas considérée. Nous proposons alors une combinaison de ROCX et I2ILP, appelée IROCX (*Interference aware Routing with*

Opportunistically Coded Exchanges), qui introduit les contraintes d'interférences de I2ILP dans le programme linéaire de ROCX. Par conséquent, IROCX bénéficie des avantages du codage réseau, de la prise en compte des interférences et du reroutage.

En plus des approches données ci-dessus, les avancées technologiques des équipements matériels permettent aux interfaces sans fil de Contrôler leur puissance de transmission. Cette nouvelle caractéristique modifie dynamiquement la zone de réception d'un nœud et donc le nombre de ses voisins. L'impact des interférences change en conséquence. Cette technologie, connue sous le nom de contrôle de topologie, permet aux nœuds de changer la topologie (graphe) du réseau dans le but de minimiser les interférences. Combiner les trois domaines abordés ici permettrait d'augmenter davantage les améliorations de chaque approche prise séparément.

- Nous combinons (Chapitre 4) IROCX avec l'approche du contrôle de topologie. Cet algorithme suppose que les nœuds ont plusieurs niveaux de puissance de transmission. Ils sont donc capables d'adapter la topologie du réseau pour maximiser le nombre des flux admis tout en respectant leur QoS en termes de bande passante. L'impact du contrôle de topologie sur le schéma d'Alice et Bob n'est pas évident car plusieurs zones d'interférences sont impliquées. Pour vérifier que cet impact est positif, nous adaptons la manière de calculer les liens qui s'interfèrent mutuellement et proposons l'algorithme TC-IROCX (*Topology Control and Interference aware Routing with Opportunistically Coded Exchanges*). Le codage réseau réduit le nombre de transmissions, le modèle basé sur les cliques garantit le respect de la QoS, le Contrôle de topologie ajuste les zones interférences et le reroutage équilibre la charge sur les cliques. Grâce à cette synergie, près de 100% des ressources du réseau sont utilisées pour maximiser le taux

d'admission des flux tout en respectant leur QoS. Les résultats de simulation ont également montré que TC-IROCX surpassait tous les algorithmes cités précédemment.

Prise en compte du délai

La plupart des travaux sur le codage réseau ont été appliqués pour améliorer le débit. Le compromis entre débit et délais engendre une augmentation de débit qui dépend du délai. C'est pourquoi le délai, qui est un paramètre de performance crucial, devrait être bien ajusté lors de l'application du codage réseau.

Au vu des explications ci-dessus, nous pouvons remarquer que le codage réseau a divers paramètres: la manière de combiner les paquets, la taille de la base de l'espace vectoriel des coefficients, le nombre de paquets à combiner, etc. L'objectif ici est d'utiliser certains d'entre eux pour gérer le délai de réception et de décodage des paquets. En effet, quand un ensemble de k paquets (appelés génération) sont combinés ensemble, la destination, dans notre système, ne peut les exploiter qu'après réception de k combinaisons indépendantes. La couche application utilise des paquets natifs (non codés) et a, par conséquent, à attendre jusqu'à la fin de la phase de décodage. Cependant, réduire le nombre de paquets à combiner décroît les gains du codage réseau en termes de robustesse et débit, l'augmenter engendre un long délai pour la couche application. Un travail intéressant est donc de gérer en même temps le débit et le délai afin de profiter des avantages du codage réseau tout en considérant la QoS en termes de délai pour les utilisateurs finaux.

- Nous étudions (Chapitre 5) le problème du délai pour le trafic multicast en proposant une manière d'adapter dynamiquement la taille de génération aux variations de l'état du réseau (délai de bout en bout, congestion, pertes de paquets). L'approche résultante, appelée DYGES (*DY*namique *GE*neration *Size*), conserve les avantages du codage réseau en termes de débit

et robustesse contre les pertes tout en respectant un seuil de délai de décodage imposé par l'application. Ainsi, DYGES prend en compte les contextes de réseau et de contenu et est dit « *Context-aware* ». Nous montrons, sur un réseau sans perte, que DYGES assure un délai de décodage régulier.

- Ensuite, nous étudions (Chapitre 6) le comportement de DYGES dans un réseau avec pertes de paquets. Contrairement aux pertes d'ACK, celles des paquets de données sont contrecarrées par l'utilisation du codage réseau et de la redondance. Nous montrons que la perte des ACKs a un impact considérable sur les performances de DYGES. Pour pallier ce problème, nous tirons profit du mode d'écoute opportuniste déjà utilisé sur les nœuds pour les paquets de données. Ce mécanisme rend la transmission des ACKs plus fiable et maintient un délai de décodage régulier même quand le taux de pertes sur les liens varie, apportant ainsi plus de robustesse contre les pertes. L'algorithme amélioré est appelé, RDYGES (*Reliable DYnamic GEneration Size*).
- Grâce à ses avantages, DYGES s'avère être une solution intéressante pour l'acheminement de contenu en améliorant la QoS et la QoE (Qualité d'Expérience) dans un réseau de domiciles. Ce réseau hétérogène comprend une *Internet-box* transmettant un trafic de diffusion vidéo à des IPTVs. L'architecture que nous proposons (Chapitre 7) tire profit du fait d'avoir plusieurs transmissions d'une même information. Dans le scénario considéré, DYGES permet de compenser les pertes de paquets et augmente de façon significative le débit. Nous déduisons par nos simulations que pour des applications tolérant un délai de décodage en début de diffusion, notre solution apporte une meilleure qualité de vidéo à l'utilisateur final sans nécessiter un équipement additionnel.

Augmentation de la capacité

L'exemple classique du codage réseau est le célèbre réseau Butterfly dans lequel un flux multicast est réalisé avec moins de transmissions grâce au codage réseau. Cependant, ce scénario est utilisé avec un codage déterministe, ce qui veut dire que le schéma de codage/décodage doit être connu à l'avance sur l'ensemble de la topologie du réseau. Le codage réseau aléatoire est l'une des évolutions clés qui a permis le passage à l'échelle du codage réseau et son application dans des systèmes distribués. Convaincus des avantages du codage réseau aléatoire, nous pensons que la conception d'un schéma de diffusion multi-sources multi-destinations basé sur des sous-réseaux Butterfly peut améliorer la capacité du réseau et par conséquent améliorer le débit et le délai.

- Nous présentons (Chapitre 7) le réseau GBFLY (*Generalized ButterFLY*) qui est appliqué pour des flux multi-sources multi-destinations. En augmentant le nombre de liens partagés, il surpasse le réseau classique Butterfly, en d'autres termes, il fournit des gains plus élevés en termes de nombre de transmissions et de capacité flow-max coupe-min comparés à ceux obtenus avec un simple schéma Butterfly. De plus, nous caractérisons la taille moyenne du buffer des nœuds sources en utilisant la théorie des files d'attente et confirmons nos résultats par simulation

Cette dernière partie représente une première étape pour caractériser la taille des buffers des nœuds. Une suite logique est d'établir une caractérisation sur les nœuds intermédiaires et destination. D'autres perspectives possibles résultant de cette thèse seraient de prendre en compte les contraintes d'énergie et d'étendre TC-IROCX à un schéma de codage plus sophistiqué que ce lui d'Alice et Bob. Aussi, DYGES peut être amélioré en utilisant des NACK en substitution des ACKs afin de permettre son application dans des réseaux plus volumineux.