Projet NTR : Routage opportuniste

Rogala Jimmy, de la Ferté Apolline, Dubois Quentin, Gachelin Jérémie

Table des matières

Chapitre 1

Contexte

Ce projet concerne l'approfondissement de l'étude des protocoles de routage opportuniste dans les réseaux sans fil. Il s'agit de simuler ces nouveaux protocoles "opportunistes" adaptés au milieu sans fil, pour choisir le meilleur chemin en fonction des nombreuses contraintes de cet environnement.

Les contraintes à prendre en compte, en complément des métriques habituelles pour tous les environnements comme le nombre de saut et le coût, sont les différentes conditions radio influant grandement sur les transmissions.

L'objectif visé est d'obtenir, en autres, une augmentation du débit global du système, une économie d'énergie et une amélioration de la QoS (Qualité de Service).

Chapitre 2

Hypothèses techniques

Dans un choix de simplicité, nous avons utilisé le langage Java, nous permettant de travailler facilement sur nos différents systèmes d'exploitation. C'est de plus un langage dont nous possédions tous les bases.

Le programme simule un réseau qui est constitué d'une grille de noeuds. Il est structuré de façon à rendre la communication entre les noeuds plus aisée. Chaque noeud représente un appareil (Device) et est relié à d'autres noeuds via des liens (Link). Chaque lien possède différentes caractéristiques de débits : débit instantané, moyen, minimum et maximum, générés aléatoirement pour chaque lien et modifiables à souhait.

Les unités utilisées dans le cadre de notre simulation, tant en terme de débit que de temps, sont totalement fictives et ne se réfèrent à aucune unité de base du Système international.

Chapitre 3

Scénario

Nous avons simulé la transmission de données depuis un point A, l'expéditeur, jusqu'à un point B, le destinataire. Chaque noeud possédant une file d'attente de paquets à transmettre. La transmission se fait à chaque unité de temps, la quantité de données envoyée dépend du noeud destinataire ainsi que du débit du lien qui les relie. Le noeud destinataire est fourni à chaque noeud en fonction de l'algorithme de routage utilisé.

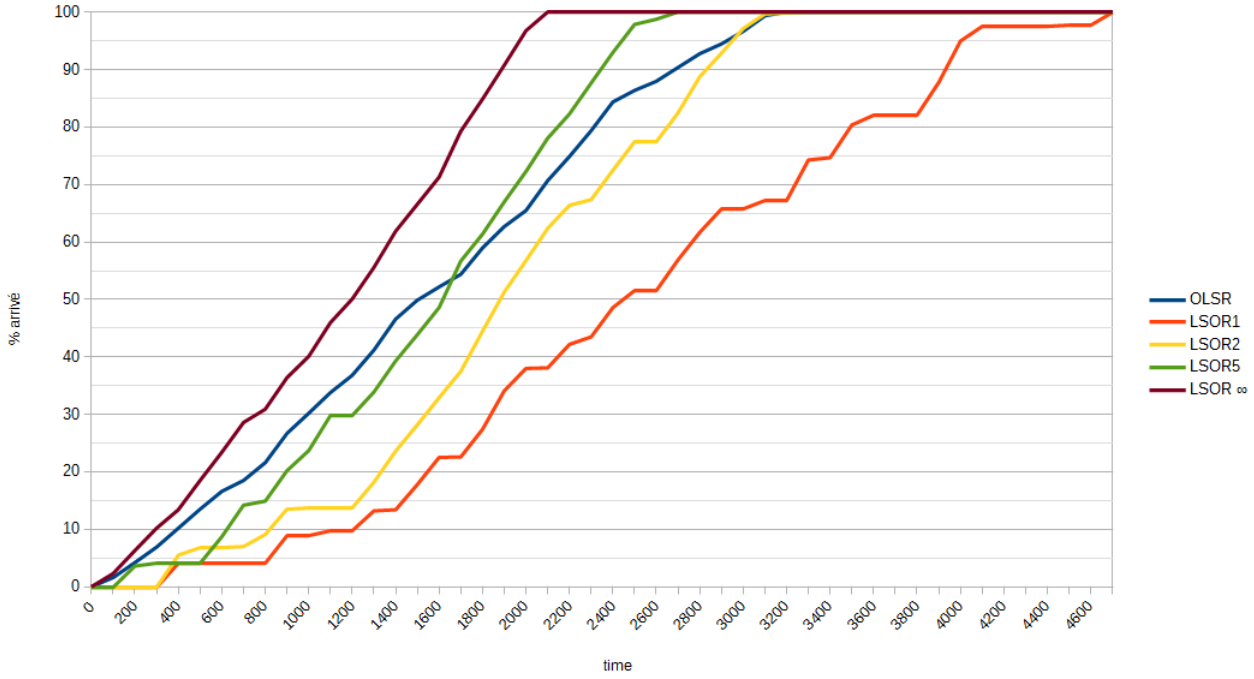
La simulation se termine lorsque le noeud d'arrivé a reçu l'ensemble des données transmises par le noeud de départ. Nous pouvons donc obtenir, pour un réseau donné et une quantité de paquets, un nombre d'unité de temps représentant la performance des différents algorithmes.

Chapitre 4

Résultats analysés

4.1 Performance en pourcentages complétés

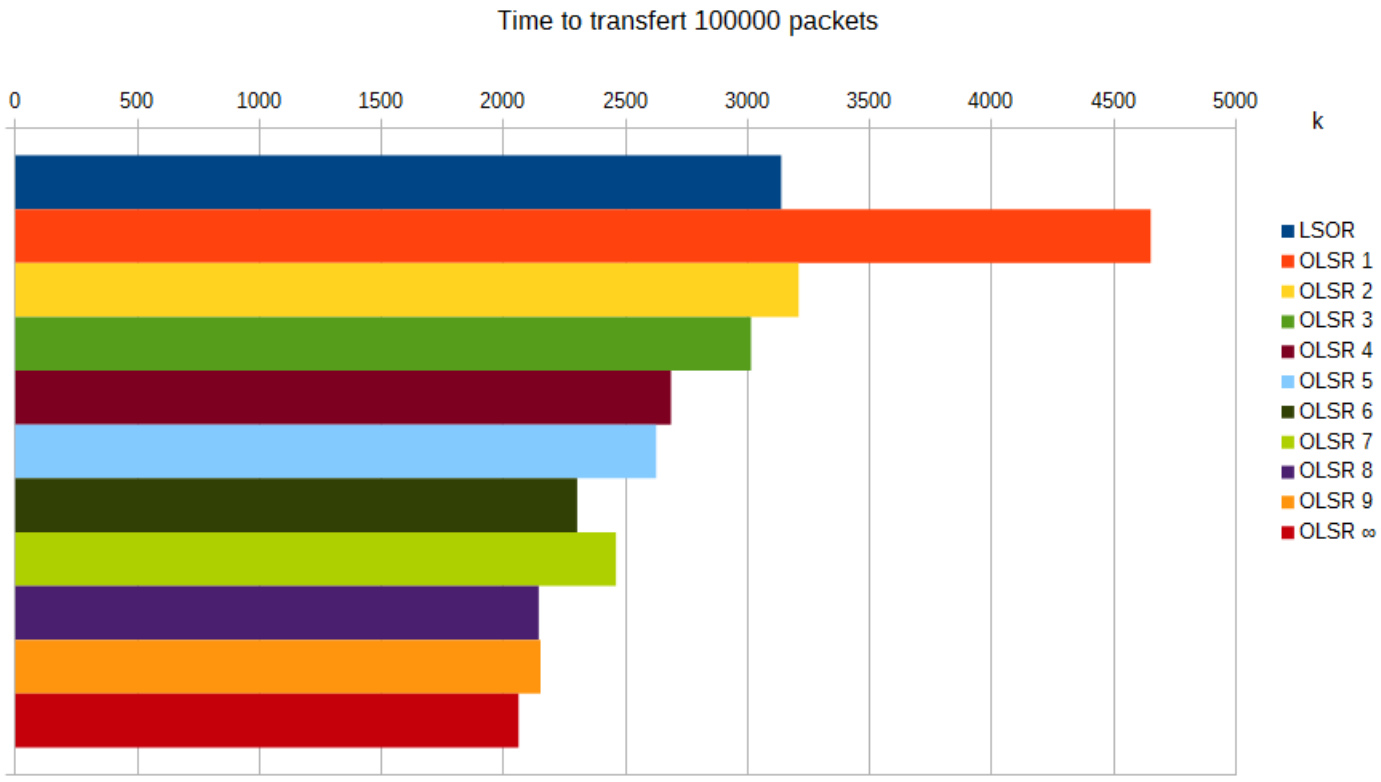
FIGURE 4.1 – Performance des algorithmes à travers le temps



La figure 4.1 illustre la performance des différents algorithmes à travers le temps pour envoyer une quantité de paquets identique. Les algorithmes LSOR peuvent être performants comme médiocres en fonction de leur visibilité du réseau, en effet, LSOR 1 ne voit qu'à 1 noeud de distance, il fait alors des choix non pertinents car les noeuds d'après peuvent avoir un débit grandement inférieur. Nous pouvons voir que dès que l'on augmente la visibilité du réseau sur l'algorithme LSOR il gagne en performance. En théorie, LSOR infini est le meilleur de notre série car il a une vision totale des débits instantanés du réseau. Nous pouvons également constater qu'OLSR, s'appuyant sur des valeurs moyennes, n'est pas le plus mauvais. Le débit moyen est une métrique intéressante, mais il arrive fréquemment que la valeur instantanée soit inférieure à la valeur moyenne, ce qui réduit la performance d'OLSR.

4.2 Performance en quantité de paquets transmis

FIGURE 4.2 – Performance des algorithmes pour 100000 paquets



La figure 4.2 illustre la performance des différents algorithmes pour 100000 paquets. On remarque une nette contre-performance de l'algorithme LSOR 1 qui finit dernier dans ce benchmark. Nous observons une exception au niveau de LSOR 6, liée au fait que ces graphiques ne sont pas réalisés sur une moyenne de simulation mais sur une seule et unique simulation.

4.3 Performance des algorithmes

FIGURE 4.3 – Performance des algorithmes à travers le temps

	OLSR	LSOR1	LSOR2	LSOR5	LSOR ∞
0	0	0	0	0	0
100	1,6	0	0	0	2,3
200	4,2	0	0	3,6	6,3
300	6,9	0	0	4,1	10,2
400	10,2	4	5,5	4,1	13,4
500	13,5	4	6,9	4,1	18,5
600	16,6	4	6,9	8,7	23,4
700	18,5	4	7	14,2	28,6
800	21,6	4	9,1	14,9	30,9
900	26,7	8,8	13,5	20,2	36,4
1000	30,2	8,8	13,7	23,7	40,1
1100	33,8	9,7	13,7	29,7	46
1200	36,8	9,7	13,8	29,8	50,1
1300	41,2	13,2	18,2	33,9	55,6
1400	46,6	13,4	23,6	39,3	61,9
1500	49,9	17,8	28,2	43,9	66,6
1600	52,2	22,5	32,9	48,6	71,3
1700	54,4	22,6	37,5	56,7	79,3
1800	59	27,4	44,5	61,4	84,9
1900	62,7	34,1	51,3	67	90,8
2000	65,5	38	56,8	72,3	96,8
2100	70,7	38,1	62,4	78,1	100
2200	74,9	42,2	66,4	82,3	100
2300	79,4	43,5	67,4	87,7	100
2400	84,4	48,6	72,5	93	100
2500	86,4	51,6	77,4	97,9	100
2600	88	51,6	77,4	98,8	100
2700	90,4	56,9	82,5	100	100
2800	92,8	61,7	88,8	100	100
2900	94,5	65,8	92,9	100	100
3000	96,7	65,8	97,2	100	100
3100	99,4	67,3	99,7	100	100
3200	100	67,3	99,9	100	100
3300	100	74,3	100	100	100
3400	100	74,7	100	100	100
3500	100	80,4	100	100	100
3600	100	82,1	100	100	100
3700	100	82,1	100	100	100
3800	100	82,1	100	100	100
3900	100	87,8	100	100	100
4000	100	95	100	100	100
4100	100	97,5	100	100	100
4200	100	97,5	100	100	100
4300	100	97,5	100	100	100
4400	100	97,5	100	100	100
4500	100	97,7	100	100	100
4600	100	97,7	100	100	100
4653	100	100	100	100	100

La figure 4.3 illustre la performance des différents algorithmes à travers le temps. On peut voir sur cette illustration les données précises émises par la simulation.

Chapitre 5

Conclusion

Ce projet nous a permis de différencier les capacités des algorithmes OLSR et LSOR, et de comprendre leur fonctionnement.