Optimisation de l'algorithme de Backoff du protocole MAC pour une meilleure performance du protocole TCP dans les MANET

HAMRIOUI Sofiane, BOUAMRA Siham et LALAM Mustapha
Laboratoire de Recherche en Informatique (LARI)
Université Mouloud Mammeri de Tizi Ouzou, Algérie
hamriouirayane@yahoo.fr

***Résumé*-Le nombre des terminaux qui ne cesse d'augmenter et l'expansion des réseaux sans fil témoignent de la demande croissante des utilisateurs pour un accès à des ressources telles que des données ou des services applicatifs, à tout moment et n'importe où. Dans ce contexte, les MANET (Mobile Ad hoc Networks) peuvent jouer un rôle important puisqu'ils peuvent être utilisés lorsque le déploiement d'infrastructures est coûteux ou sans objet, ou encore pour accroître la portée des réseaux à base d'infrastructure. Toutefois, des contraintes spécifiques aux MANET existent qui demeurent des freins majeurs au développement d'applications pour les MANET. L'une de ces contraintes est la dégradation de la performance du protocole TCP (Transfer Control Protocol) due essentiellement aux nœuds exposés et nœuds cachés qui agissent négativement sur le comportement du protocole MAC (Medium Acces Control) vis-à-vis du protocole TCP surtout si ces deux protocoles ne les prennent pas en compte et ne procèdent pas à l'adaptation de leurs comportements en faveurs des flux TCP. Dans le cadre de ce présent papier, nous présenterons une optimisation de l'algorithme de backoff du protocole MAC pour une meilleure performance du protocole TCP. Cette dernière se base sur l'ajustement de ses paramètres en fonction du nombre de nœuds que constitue le MANET.**

***Mots clés*-MANET, MAC, TCP, Backoff.**

I. INTRODUCTION

Les récentes percées technologiques dans le domaine des réseaux sans fil ont conduit à une forte évolution des environnements mobiles [1]. Les MANET [2] s'appuient sur ces derniers et proposent une solution alternative afin de fournir une connectivité à des terminaux fixes ou mobiles qui demandent un déploiement rapide. La performance du protocole TCP [3] est un paramètre de QoS (Quality of Service) [4] très important qui fait, de plus en plus, l'objet de plusieurs évaluations. Cette dernière se dégrade dans les MANET vu que le protocole TCP est confronté à de nouveaux défis [5] [6], l'une d'elles est les effets du protocole MAC [7] [8] sur le protocole. Notre travail vise l'optimisation du protocole TCP dans les MANET à travers lequel nous montrerons, en premier lieu, l'impact du protocole MAC sur la performance du protocole TCP, puis, nous passerons ensuite à étudier la solution proposée. L'ajustement des paramètres de l'algorithme de backoff du protocole MAC est le principe utilisé sur lequel repose notre démarche afin d'optimiser la performance du protocole TCP.

II. ANALYSE DU PROBLEME

Le protocole de transport le plus largement répandu aujourd'hui est TCP. Cependant, la performance de ce protocole se dégrade dans les MANET vu que ces derniers sont confrontés à de nouvelles contraintes [5] [6] spécifiques à ce type de réseaux. Parmi ces dernières on trouve les nœuds cachés et exposés. La figure 'Fig. 1' illustre ces phénomènes, sur cette dernière on suppose que A envoie à B, C veut envoyer à B. C ne peut pas recevoir de A et croit que le support est libre alors décide lui aussi d'envoyer pour B ce qui conduira à une collision au niveau de B. A n'entends pas la collision alors on dit que A est "caché" pour C. supposons maintenant B envoie pour A et C veut envoyer pour D, C entend B alors il se tait mais A n'entend pas C, donc C pourrait émettre. Dans ce cas, on dit que C est "exposé" à B.

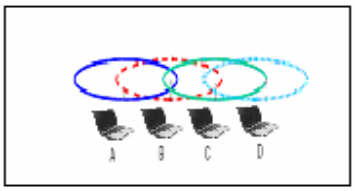

Fig. 1. Problème des nœuds caché et exposé.

L'utilisation des messages RTS (Ready To Send) et CTS (Clear To Send) est essentiellement la solution proposée pour régler le problème du nœud caché et du nœud exposé ('Fig. 2').

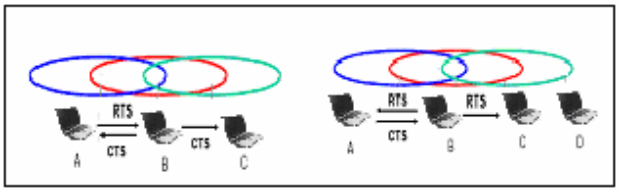

Fig. 2. Solutions des nœuds caché et exposé.

Plusieurs solutions ont été proposées dans la littérature pour améliorer la performance de TCP dans le contexte des MANET [9] [10] [11]. Ces dernières sont peu adaptées au protocole MAC employé et sont dans la plupart limitées à un protocole sur une couche sans aucune interaction. Dans la section qui suit, nous montrerons l'impact du protocole MAC sur la performance du protocole TCP.

III. Simulation de scenarios sous NS

Pour montrer l'impact du protocole MAC sur la performance du protocole TCP, nous avons utilisé l'outil NS (Network Simulator) [12]. La topologie considérée est une chaîne formée de n nœuds dans laquelle chaque nœud peut seulement communiquer avec ses voisins adjacents et seulement une seule connexion TCP est établie. On suppose que c'est le premier nœud de la chaîne qui commence de transmettre pour le dernier nœud. La disposition des nœuds dans l'espace est donnée par les figures 'Fig. 4' et 'Fig. 5' Dans nos simulations, le modèle de propagation utilisé est le modèle two-ray ground, pour le protocole MAC on a opté pour le 802.11 IEEE et un modèle d'antenne omnidirectionnelle de l'outil NS. Comme protocole de routage nous avons utilisé AODV, une taille de file d'interface de 50 dans chaque nœud. Nous avons testé la version NewReno de TCP. La taille des paquets de données TCP est de 1040 bytes. Dans les scripts, l'option *macTrace* est activée dans le but d'avoir les détails sur les trames du protocole MAC, cela nous a permis d'analyser la raison de chaque perte de paquet TCP qui s'est produite. Nous avons simulé plusieurs scénarios avec un nombre de nœuds différent. Nous donnerons ci-après les résultats de simulations des scénarios avec n = 3, 7, 9 et 12.

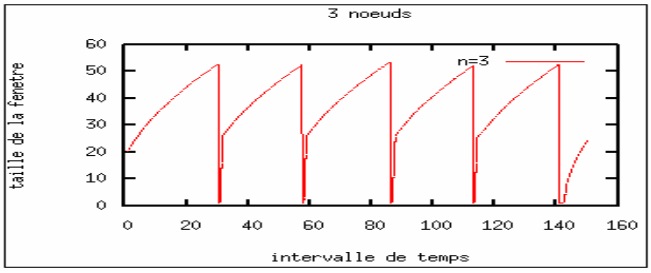

Fig. 3. L'évolution de la taille de la fenêtre de TCP dans le cas n=9 nœuds.

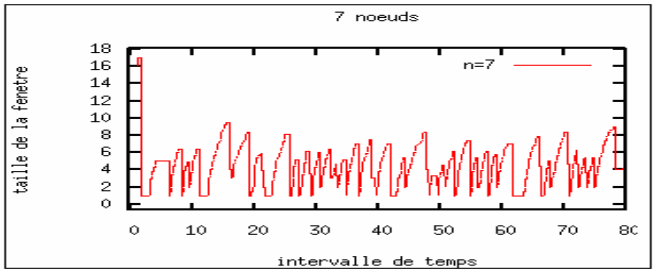

Fig. 4. L'évolution de la taille de la fenêtre de TCP dans le cas n=9 nœuds.

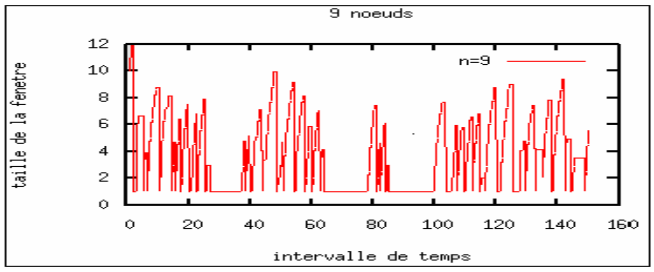

Fig. 5. L'évolution de la taille de la fenêtre de TCP dans le cas n=9 nœuds.

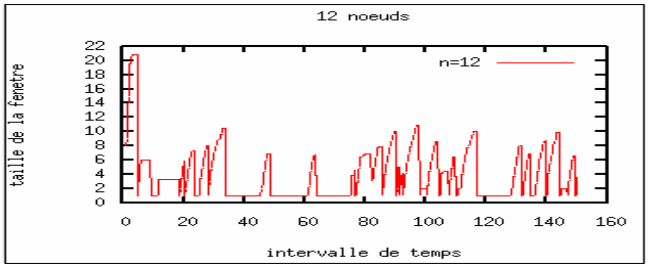

Fig. 6. L'évolution de la taille de la fenêtre de TCP dans le cas n=9 nœuds

Le graphe de la figure 'Fig. 3' reflète le fonctionnement du protocole TCP, nous distinguons facilement une même sous courbe répétée plusieurs fois. A son tour, cette courbe est divisée en deux étapes séparées par un seuil égal à 25 sur le graphe. Ces deux étapes correspondent respectivement au départ lent et évitement de congestion dont le passage d'une étape à l'autre dépend seulement de la congestion du médium. Nous pouvons dire aussi, et le graphe le confirme, qu'il y a pas de perte de paquet TCP pour n'importe quelle valeur de la fenêtre actuelle qui dépasse la valeur 50, cela est lié à l'absence des effets des phénomènes du nœud caché et exposé (car le mécanisme RTS/CTS utilisé par le protocole MAC 802.11 est idéale pour une chaîne de 3 nœuds). Apres avoir dépassé la valeur 52, la graphe subit une chute vers le 1, cela est du à la saturation du réseau. Pour les graphes des autres figures, nous constatons que, plus le nombre de nœuds participants dans la chaîne augmente, plus les diminutions de la taille de la fenêtre actuelle sont fréquentes. Pour le cas de la figure 'Fig. 5' par exemple, la fenêtre TCP se stabilise à plusieurs reprises à la valeur 1 et cela entre l'instant 25 ms et 40 ms, entre 62 ms et 80 ms et entre 85 ms et 100 ms. Le maintien de la valeur de la fenêtre à 1 par TCP durant ces intervalle de temps est du à la présence de pertes de paquets. On analysant les fichiers trace obtenus, nous avons constaté que ces pertes de paquets TCP sont dues aux pertes des trames de contrôle utilisé par le protocole MAC 802.11 lors de l'initialisation des échanges de trames de données (voir les figures 'Fig. 7' et 'Fig. 8'), ce qui empêchera la transmission des paquets correspondants. Après expiration du timeout, TCP procédera à la régularisation de son débit qui réduira la taille de la fenêtre ainsi que le nombre de paquets échangés. La source de ce phénomène, comme expliqué dans la section précédente, sont les nœuds caché et exposé qui apparaissent plus en augmentant le nombre de

nœuds dans les scénarios. Les figures 'Fig. 7' et 'Fig. 8' expliquent l'impact du nœud caché et celui du nœud exposé au cours de nos simulations.

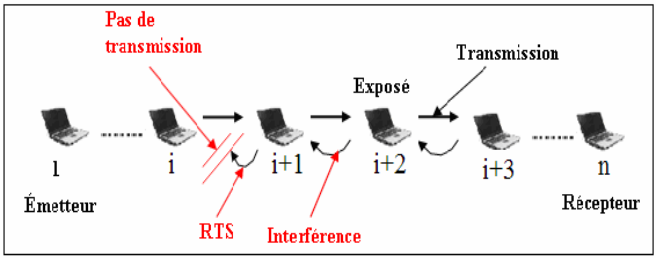

Fig. 7. Impact du nœud exposé sur TCP

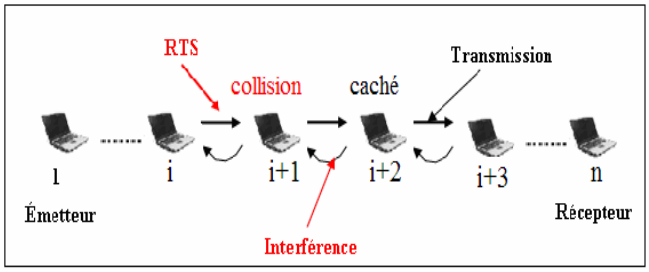

Fig. 8. Impact du nœud caché sur TCP

IV. L'OPTIMISATION PROPOSEE

Le protocole MAC repose sur l'algorithme de backoff qui lui permet de déterminer qui va accéder au médium sans fil tout en évitant les collisions. Dans le cas d'une station trouvant un canal occupé, l'émission est différée conformément à une procédure de Backoff dont le principe est le suivant : tant que le canal n'est pas libre pour un temps DIFS (après une réception réussie) ou pour un temps EIFS (après une réception échouée), le temps Backoff est décrémenté. Ce temps est calculé de la façon suivante:

$$BackoffTime = BackoffCounter * aSlotTime \qquad (1)$$

Où:

aSlotTime : est une constante de temps

BackoffCounter : est un entier de distribution uniforme dans l'intervalle [0, CW] et CW est la fenêtre de contention dont les limites minimales et maximales sont (*CWmin*, *CWmax*) sont définies par le standard suivant la couche PHY utilisée. La valeur CW est augmentée aussi en cas de non disponibilité du canal selon la formule suivante :

Avec

$$\left.\begin{array}{r} m \leftarrow m + 1 \\ CW(m) = (CWmin + 1)*2^{m} - 1 \\ CWmin <= CW(m) <= CWmax \\ m : \text{le nombre de collisions survenues} \end{array}\right\} \qquad (2)$$

Nous proposons une optimisation de cet algorithme de backoff. Le principe de cette optimisation est d'essayer de déterminer des valeurs pivots (optimales) pour *CWmin* et *CWmax* qui réduiront les collisions entre nœuds ainsi réduire de leur impact sur le protocole TCP.

En premier lieu, nous avons réalisé plusieurs simulations sur des scénarios différents dans lesquels nous avons fait varier le nombre de nœuds ainsi que les valeurs de *CWmin* et *CWmax*, nous avons constaté que les valeurs de *CWmin* et *CWmax* influence sur la performance de TCP. Les résultats nous ont permis de dégager des valeurs optimales de ces dernières pour chaque scénario en fonction du nombre de nœuds employés. On a constaté que les valeurs de *CWmin* et de *CWmax* doivent être augmenté avec l'augmentation du nombre de nœuds participants dans le scénario. Les figures 'Fig. 9', 'Fig. 10', 'Fig. 11' et 'Fig 6.12' donnent les résultas obtenus avec 9 nœuds et 12 nœuds.

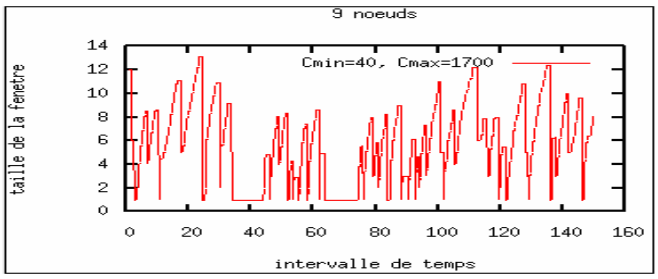

Fig. 9. L'évolution de la taille de fenêtre TCP avec *CWmin*=40 et *CWmax*=1700 dans le cas n=9 nœuds.

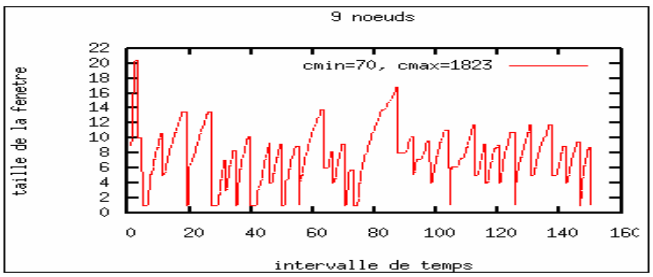

Fig. 10. L'évolution de la taille de fenêtre TCP avec *CWmin*=70 et *CWmax*=1823 dans le cas n=9 nœuds.

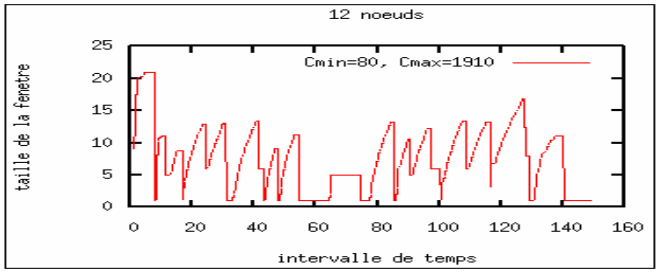

Fig. 11. L'évolution de la taille de fenêtre TCP avec *CWmin*=80 et *CWmax*= 1910 dans le cas n=12 nœuds.

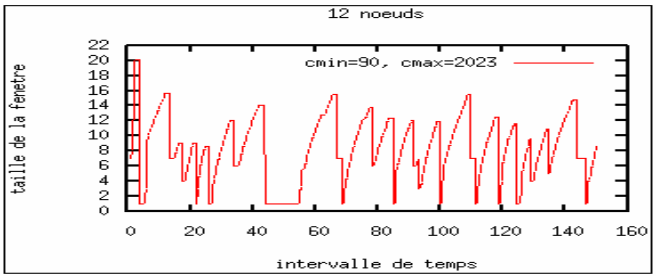

Fig. 12. L'évolution de la taille de fenêtre TCP avec *CWmin*=90 et *CWmax*=2023 dans le cas n=12 nœuds.

Les figures 'Fig. 9' et 'Fig. 10' montrent une nette amélioration de la performance de TCP par rapport à la figure 'Fig. 5' et cela sur le plan perte de paquets qui est réduite et la taille de la fenêtre qui est augmentée. Nous rappelons que les valeurs de *CWmin* et de *CWmax* utilisées dans le scénario de la figure 'Fig. 5' sont celles fixées par défaut à savoir *CWmin*=31 et *CWmax*=1023. Cette amélioration revient au fait qu'on a augmenté les valeurs des bornes de la fenêtre de contention des nœuds ce qui réduira la probabilité d'avoir les mêmes chance d'accéder au médium sans fil ainsi réduire les collisions qui conduira ainsi à diminuer leurs conséquences sur le protocole TCP ainsi améliorer sa performance. Les figures 'Fig. 11' et 'Fig. 12' montrent aussi une amélioration de la performance de TCP par rapport à la figure 'Fig. 6'. Après avoir fixé *CWmin* et *CWmax* à des valeurs différentes durant plusieurs scénarios, nous avons constaté que les valeurs 70 pour *CWmin* et 1823 pour *CWmax* donne le meilleur résultat pour le scénario à 9 noeuds. Quant au scénario à 12 nœuds, les valeurs dégagées comme optimales sont *CWmin* =90 et *CWmax* =2023. Ces valeurs sont relativement plus grandes que celle utilisées dans le scénario avec 9 nœuds à cause de l'augmentation du nombre de nœuds qui est de 12 nœuds. Nous avons également constaté durant nos simulations que des très petites valeurs ou des très grandes valeurs pour *CWmin* et *CWmax* n'améliore pas ou dégrade la performance de TCP.

V. MODELISATION DE L'OPTIMISATION

A. *Modélisation de CWmin*

Apres avoir réalisé plusieurs scénarios avec des nombres différents de nœuds communiquant, nous avons relevé pour chaque scénario les valeurs optimales *CWmin*. Ces dernières nous ont permis de dresser le graphe illustré sur la figure 'Fig. 13'.

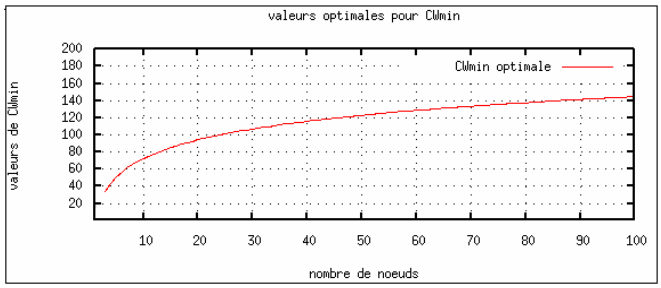

Fig. 13. Les valeurs optimales de *CWmin* pour les scénarios entre 3 et 100 nœuds.

Le graphe de la figure 'Fig. 13' correspond clairement à une courbe d'une fonction logarithmique translaté d'une constante, alors on pose la fonction de la forme la forme suivante:

$$CWmin(n) = y(n) = \beta \log(n) \qquad (3)$$

Tel que ***n*** est le nombre de nœuds employés dans le scénario.

Pour déterminer la valeur de la constante β, nous procédons comme suite :

$$\beta = y(n)/\log(n) = 31. \qquad (4)$$

31 correspond bien à la valeur initiale de *CWmin*

B. *Modélisation de CWmax*

Apres avoir réalisé plusieurs scénarios avec des nombres différents de nœuds communiquant, nous avons également relevé pour chaque scénario les valeurs optimales *CWmax*. Ces dernières nous ont permis de dresser le graphe illustré sur la figure 'Fig. 14'.

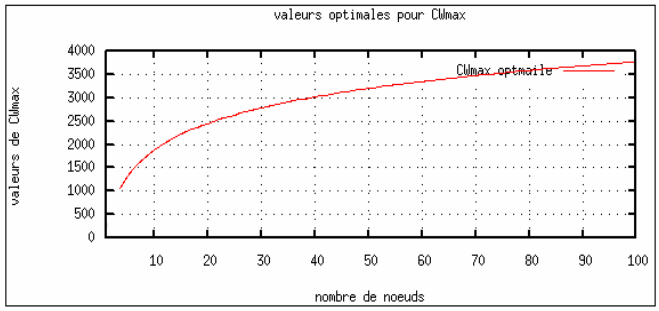

Fig. 14. Les valeurs optimales de *CWmax* pour les scénarios entre 3 et 100 nœuds.

La figure 'Fig. 14' illustre un graphe correspondant à une courbe d'une fonction logarithmique translaté d'une constante, alors on pose la fonction de la forme la forme suivante:

$$CWmax(n) = y(n) = \gamma \log(n) \qquad (5)$$

Tel que *n* est le nombre de nœuds employés dans le scénario.

Pour déterminer la valeur de la constante γ, nous procédons comme suite :

$$\gamma = y(n)/\log(n) = 820 = 1024 * 5/4 \quad (6)$$

1024 correspond bien à la valeur initiale de *CWmax*

C. Algorithme de Backoff optimal

Apres optimisation, le nouvel algorithme de Backoff devient donc :

$$\left.\begin{array}{l} m \leftarrow m + 1 \\ CW(m) = (CWmin(n) + 1)*2^m - 1 \\ \underline{\text{Tel que}} : \\ CWmin(n) <= CW(m) <= CWmax(n) \\ CWmin(n) = \beta \log(n) \\ CWmax(n) = \gamma \log(n) \\ \underline{\text{Avec}} : \\ m : \text{le nombre de collision.} \\ n : \text{est le nombre de nœuds employés} \\ \beta : \text{est une constante égale à } 31 \\ \quad = \text{CWmin initiale} \\ \gamma : \text{est une constante égale à } 1024*5/4 \\ \quad = \text{CWmax initiale}*5/4 \end{array}\right\} \quad (7)$$

Nous fixons comme objectif de nos proches travaux de réutiliser cet algorithme avec des topologies différentes et de déterminer le lien entre les constantes β et γ et la topologie choisit ainsi que entre ces dernières et les facteurs de l'environnement de communication.

VI. CONCLUSION ET PERSPCTIVES

Dans le cadre de ce papier, nous avons commencé par montrer les effets des problèmes des nœuds cachés et des nœuds exposés sur la performance du protocole TCP. Ensuite, on a proposé une optimisation de l'algorithme de backoff du protocole MAC pour laquelle on a réalisé des simulations afin de monter sa contribution. Les résultats obtenus étaient satisfaisant et montrent une nette amélioration de la performance de TCP. Notre travail ne constitue qu'une partie de ce qui doit se faire dans le contexte de l'amélioration de la performance de TCP dans les MANET en intervenant sur les paramètres du protocole MAC. Nous fixons comme objectif de nos proches travaux de réutiliser cet algorithme avec des topologies différentes et de déterminer le lien entre les constantes β et γ et la topologie choisit ainsi que entre ces dernières et les facteurs de l'environnement de communication.

REFERENCES

[1] A. Ahmad, "Wireless and Mobile Data Networks,'' *Wiley-Interscience Published by John Wiley & Sons, Inc*., Hoboken, New Jersey. Published simultaneously in Canada, 2005.

[2] S. Basagni, M. Conti, S. Giordano and I. Stojmenovic, "Mobile Ad hoc Networking,'' *Ieee Press, Wiley-Interscience A John Wiley & Sons, Inc*., Publication, 2004.

[3] M.W. Murhammer, O. Atakan, S. Bretz, L.R. Pugh, K. Suzuki, and D.H. Wood, "TCP/IP Tutorial and Technical Overview,'' International Technical Support Organization, 1998.

[4] J. Li, "Quality of Service (QoS) Provisioning in Multihop Ad Hoc Networks,'' Doctorate of Philosophy, Compuer Science in the Office of Graduate Studies, University of California, 2006.

[5] A.A. Hanbali, E. Altman, and P. Nain, "A Survey of TCP over Ad Hoc Networks,'' INRIA, June 2005.

[6] V. Anantharaman, S.J. Park, K. Sundaresan and R. Sivakumar, "TCP performance over mobile ad-hoc networks: a quantitative study,'' *Wirel. Commun. Mob. Comput*, 2003; 3:1–20 (DOI: 10.1002 /wcm.172).

[7] P. Mohapatra, and S.V. Kkishnamurthy, Ad hoc Networks Technologies and Protocols, Springer Science + Business Media, Inc, 2005.

[8] H. Zhai, J. Wang, X. Chen, and Y. Fang, "Medium access control in mobile ad hoc networks: challenges and solutions,'' *Wirel. Commun. Mob. Comput*, 2006; 6:151–170.

[9] R.D. Oliveira, and T. Braun, "TCP in Wireless Mobile Ad Hoc Networks,'' NCCR-MICS project, 2002.

[10] S. Papanastasiou, S. Ould-Khaoua, and L.M. Mackenzie, "On the evaluation of TCP in MANET,'' *Proc. of International Workshop on Wireless Ad-hoc Networks, London, UK*, 23 - 26 May 2005.

[11] S. Xu, T. Saadawi, "Performance evaluation of TCP algorithms in multi-hop wireless packet networks,'' Wireless Communications and Mobile Computing 2 (1) (2002) 85–100.

[12] http://www.isi.edu/nsnam/ns/ns-documentation.