

Adaptive Topology-Transparent Distributed Scheduling in Wireless Networks

Qiong Sun, Victor O.K. Li, and Ka-Cheong Leung

Department of Electrical and Electronic Engineering
The University of Hong Kong, Hong Kong SAR, China
E-mail: {joansun, vli, kcleung}@eee.hku.hk

Abstract—Transmission scheduling is a key design problem in wireless multi-hop networks. Many transmission scheduling algorithms have been proposed to maximize the spatial reuse and minimize the time division multiple access (TDMA) frame length. Most of the scheduling algorithms are topology-dependent. They are generally graph-based and depend on the exact network topology information. Thus, they cannot adapt well to the dynamic wireless environment. In contrast, topology-transparent TDMA scheduling algorithms do not need detailed topology information. However, these algorithms offer very low minimum throughput. The objective of this work is to propose an adaptive topology-transparent scheduling algorithm to offer better throughput performance. With our algorithm, each node finds a transmission schedule so as to reduce the transmission conflicts and adapt better to the changing network environment. The simulation results show that the performance of our algorithm is better than the existing topology-transparent algorithms.

Index Terms—Adaptive algorithms, distributed scheduling, TDMA, topology-transparent, wireless networks.

I. INTRODUCTION

A packet radio network consists of a number of geographically dispersed mobile radio nodes which can communicate wirelessly with each other. The network topology is not fixed because of node movements and limited transmission power. Hence, designing an effective distributed transmission scheduling algorithm in multi-hop mobile ad hoc networks is a challenging research issue.

To achieve efficiency and robustness in the wireless environment, topology-transparent algorithms have been proposed [1], [2], [3], [4], [5]. In these algorithms, collisions may occur but no topology information is used, and hence no topology updates are required. In [2], the authors employed the Galois field coding theory [6] to produce a unique transmission schedule for each node so that a node has at least one conflict-free transmission slot in each frame. In [4], the authors extended the idea proposed in [2], so that the scheduling algorithm also maximizes the minimum guaranteed throughput for point-to-point communication. In [5], a topology-transparent algorithm is combined with a medium access (MAC) control scheme to improve the throughput. In [3], an adaptive elimination algorithm is proposed. Each node is assigned a permanent codeword that is unique within its two-hop communication range. However, nodes need to exchange their codewords and stop transmitting in some assigned time slots to avoid collisions.

The objective of this work is to propose an adaptive topology-transparent scheduling algorithm to offer a better average throughput performance in addition to the minimum guaranteed throughput. Each node starts to transmit based on the initial allocated time slots. The collision rate of each scheduled transmission time slot is then measured. For each node, if the collision rates of some selected time slots are greater than a given threshold, such time slots will be replaced by employing a new transmission sequence obtained from another available code. Thus, each node will find the best transmission sequence to adapt to the network environment. Hence, the proposed scheduling algorithm infers some topology-dependent information for performance improvement. In order to distinguish our proposed algorithm from the previous topology-transparent algorithm in [1], we call our algorithm the adaptive topology-transparent algorithm (ATT) and the existing pure topology-transparent algorithm (PTT). It is also different from our previous work [7], in which a pure topology-transparent scheduling scheme is proposed to improve the minimum average throughput for multicasting and broadcasting. The simulation results show that our proposed ATT substantially outperforms PTT.

The rest of the paper is organized as follows. The system model is introduced in Section II. Section III presents our proposed ATT in details. We evaluate the performance of ATT by simulation in Section IV and conclude in Section V.

II. SYSTEM MODEL

In this paper, we assume that the transmission channel is error-free and a reception failure is due to a packet collision. A node cannot transmit and receive simultaneously because there is only one transceiver per node. All network nodes are assumed to be homogeneous. To simplify the calculation, we assume the interference range of a node equals its transmission range. A wireless multi-hop network can be modeled as a bidirectional graph $G_v = (V, E)$, where V is the set of nodes and E is the set of edges to indicate which pair of nodes is connected. The number of nodes $|V|$ is denoted by N . The degree of a node v is defined as the number of its neighbors, which is always less than or equal to the maximum node degree D_{max} , which is assumed to be a constant [8].

In our algorithm, time is divided into equal length frames. Each frame is divided into q subframes, each of which consists of p time slots. During a subframe, each node selects a time

slot to transmit. Thus, each node has q transmissions within a time frame. The time frame structure is shown in Fig. 1. For a given node, its transmission slots are determined by its assigned polynomial as discussed in [4]. Here, a new term “codeword” is used, denoted by CD_v . It indicates which sequence of the time slots is selected to transmit for node v . Thus, as shown in Fig. 1, the codeword for node v is $CD_v = (3, 2, \dots)$. Furthermore, the number of collisions for any two nodes per frame is at most k , where k is the degree of the assigned polynomial [4].

To ensure that every node has at least the minimum guaranteed throughput, two constraints should be satisfied as discussed in [4]. Thus, we have:

$$p \geq N^{\frac{1}{k+1}} \quad (1)$$

$$q \geq kD_{max} + \alpha \quad (2)$$

where $\alpha \geq 1$.

Based on (1), a code family is defined as a set of all possible codewords calculated from the assigned polynomials. The number of elements in the code family, denoted by Q , is p^{k+1} . Each node is assumed to know the availability of each feasible codeword.

As discussed in [1], the frame length can be minimized by choosing an optimal set of values for p , q , and k . This in turn maximizes the minimum throughput. Therefore, $p = p_1$, where p_1 is the minimum prime number or prime power $\geq \lceil k_0 \rceil D_{max} + \alpha$ and $q = \lceil k_0 \rceil D_{max} + \alpha$, or $p = p_2$, and p_2 is the minimum prime number or prime power $\geq \lfloor k_0 \rfloor D_{max} + \alpha$ and $q = \lfloor k_0 \rfloor D_{max} + \alpha$.

$$kD_{max} + \alpha = N^{\frac{1}{k+1}} \quad (3)$$

where k_0 is the unique positive root of (3). The packet transmissions among different nodes in the network can be coordinated as follows:

- 1) Obtain k_0 from (3) and then determine p and q . Then, the minimum length of a frame, $L_{min} = \min(p_1 \lceil k_0 \rceil D_{max} + \alpha, p_2 \lfloor k_0 \rfloor D_{max} + \alpha)$.
- 2) Each node is randomly assigned a unique codeword.
- 3) Each node transmits its data packets at its assigned slots according to an adaptive scheduling algorithm to be described in the next section.

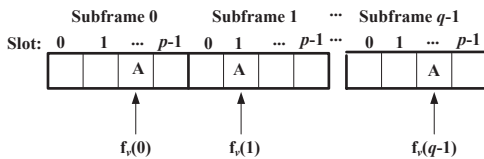


Fig. 1. The frame structure.

III. ADAPTIVE TOPOLOGY-TRANSPARENT DISTRIBUTED SCHEDULING SCHEME

A. Proposed Adaptive Scheme

The work flow of the proposed scheduling scheme, as shown in Fig. 2, is described in the following:

- 1) Initially, each node can transmit its packets based on its initial calculated codeword.
- 2) For each time slot y chosen by Node i , the collision rate is estimated based on the latest H time frames, and can be calculated as follows:

$$CR_i(y) = \frac{\sum_{x=t-H}^{t-1} 1_{\{slotid_i(x)=y \text{ and this transmission fails}\}}}{\sum_{x=t-H}^{t-1} 1_{\{slotid_i(x)=y\}}} \quad (4)$$

where $slotid_i(x)$ denotes the slot identification number in frame x , and 1_A is the indicator function of event A , which means that this value equals one when A happens and it is zero otherwise.

- 3) At the beginning of each frame, the measured collision rate of a selected time slot is compared with the given collision threshold CRT to see whether the time slot should be kept or replaced. If $CR_i(y) < CRT$, Node i will continue to use slot y to transmit. Otherwise, y will be replaced by the codeword replacement method to be described below. Thus, whether we keep or replace the selected time slot y depends on the following condition:

$$y \text{ is } \begin{cases} \text{kept, if } CR_i(y) < CRT; \\ \text{replaced, if } CR_i(y) \geq CRT. \end{cases} \quad (5)$$

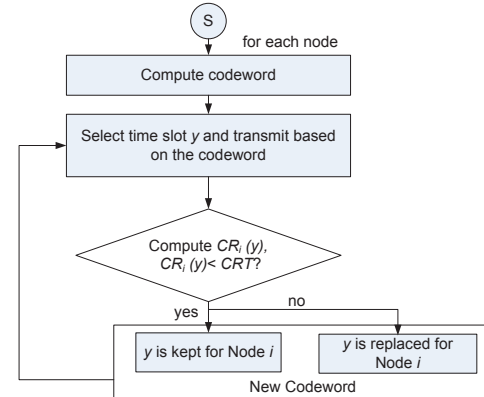


Fig. 2. The work flow of ATT.

H and CRT should be configured carefully to match the dynamics of the network. We will discuss how to find the appropriate values for H and CRT in Section IV.

Besides, if the new codeword selected by Node i is the same as the codeword being used by its neighbor j , Node i and its neighbor j will collide with each other at all allocated slots. When Node i is aware of this situation, it will select another available codeword for the following transmission time frame. However, the probability of codeword collision is still very small as discussed in Section III-B.

B. Performance Analysis of ATT

The probability of collision with other nodes is analyzed when Node i adopts the codeword replacement method. The worst situation occurs when a new codeword selected by Node i is the same as one of its neighbors. In a network, the probability that a codeword is already occupied, denoted as P_c , is $\frac{N}{Q}$, since there are N nodes and Q codewords. With our codeword replacement method, the probability that the new codeword selected by Node i is already occupied, $P_c(i)$, can be estimated as $P_c(i) \approx \frac{N}{Q}$. In the worst case, Nodes i and j will collide with each other at all their allocated slots with probability $P_c(i)$. In reality, such collision probability is much smaller than $P_c(i)$, because these two nodes sharing the same codeword may not be neighbors. Since the maximum degree is D_{max} , the probability that Node j is a neighbor of Node i is $P_d \leq \frac{D_{max}}{N}$. Hence, the probability that Node i shares the same codeword with its neighbor j is $P(i, j) = P_c(i) \cdot P_d \leq \frac{N}{Q} \cdot \frac{D_{max}}{N} = \frac{D_{max}}{Q}$. Finally, we can get:

$$P(i, j) \leq \frac{D_{max}}{Q} = \frac{D_{max}}{p^{k+1}} \leq \frac{D_{max}}{N} \quad (6)$$

Hence, the probability $P(i, j)$ is indeed very small. For example, when N is 1000 and D_{max} is 10, $P(i, j)$ is no larger than 0.01. It means that Node i have a chance of up to 0.01 to collide with one of its neighbors when they both transmit simultaneously.

IV. SIMULATION RESULTS

In this section, we compare the performance of our proposed algorithm (ATT) with PTT in which the codewords are unchanged.

A. Simulation Setup

In our simulation, we use the average system throughput G_{sys} as the performance metric. G_{sys} is defined as the ratio of the average total number of successful transmissions for all nodes in the system within a frame, T_{sys} , to the frame length, L , i.e., $G_{sys} = \frac{T_{sys}}{L}$, where $T_{sys} = \sum_{i=1}^N T_i$, and $L = p \cdot q$, where p and q are determined by (1) and (2). The value of k is one or two based on (3). T_i is the mean number of successful transmissions within a frame for Node i . The performance of ATT is compared with PTT under three different types of network mobility, namely, highly dynamic, dynamic, and static. Here, we adopt the ‘‘Gauss-Markov’’ mobility model [9], which has been shown to be more realistic than the popular random waypoint model. In each simulation scenario, all nodes are initially located randomly inside the simulation region of 1000 m \times 1000 m and allowed to move. The speed of any node in a static network is zero since all nodes are fixed. For dynamic and highly dynamic networks, the speed of a node is chosen from a Gaussian distribution with a standard deviation of 0.325 ms⁻¹. The direction of movement is also chosen from a Gaussian distribution with a mean of $\frac{\pi}{2}$ and a standard deviation of $\frac{\pi}{8}$. A tuning parameter α ($0 \leq \alpha \leq 1$) is used to vary the randomness. A scenario with totally random node movements is obtained by setting $\alpha = 0$, and a scenario

with nodes of a constant speed and no directional change is obtained by setting $\alpha = 1$. Here, α is set to 0.5. Based on the minimum and maximum speed values in [10], the average speed is set to 0.9 ms⁻¹ in a dynamic network and 1.8 ms⁻¹ in a highly dynamic network.

B. Simulation Results

In the simulation, we not only compare the performance of ATT with PTT, but also determine the appropriate values of CRT and H for each of the three types of networks with different levels of mobility. The following simulations are shown based on $N = 500$ and $D_{max} = 7$, so that $k = 2$, $p = 11$, and $q = 15$ based on (1), (2), and (3). Although other configurations of N and D_{max} are also evaluated, i.e. $(N, D_{max}) = (100, 5)$ and $(N, D_{max}) = (1000, 10)$, their performance trends are similar and thus they are not included due to space limitations. The number of runs for each experiment is ten. In addition to N and D_{max} , there are two other design parameters, namely, CRT and H . They are configured with $(CRT, H) = (20\%, 2)$, $(50\%, 4)$, and $(70\%, 6)$.

1) *Static network*: Table I shows the throughput improvement achieved by ATT over PTT with nine different configurations of CRT and H . Obviously, the smaller the values of H and CRT , the higher the throughput because smaller values of H and CRT would imply that nodes have more chances to acquire the recent knowledge about the changes in the network topology, and they may change their codewords accordingly. Figs. 3-4 show how the performance of ATT varies with the increase in the number of rounds of codeword change under various configurations of CRT and H . As shown in these two figures, the average system throughput of PTT remains unchanged, since its codeword assignment is static. However, the average system throughput of ATT increases at first and becomes stable when the number of rounds increases. The average system throughput of ATT is always higher than that of PTT, because each node can find the better matched codeword with respect to the network topology.

TABLE I
COMPARISON OF THROUGHPUT IMPROVEMENT UNDER A STATIC NETWORK

	$H = 2$	$H = 4$	$H = 6$
$CRT = 20\%$	23.76%	21.80%	15.80%
$CRT = 50\%$	21.85%	18.55%	13.79%
$CRT = 70\%$	21.48%	18.58%	12.53%

2) *Dynamic network*: Similar to the static network, ATT outperforms PTT in a dynamic network, as shown in Table II. Furthermore, when $H = 2$ and $CRT = 50\%$, ATT yields the largest performance improvement. In the dynamic network, nodes are more efficient at detecting the changes in the network topology by choosing a smaller H . A larger H leads a node to use inaccurate information and hence make a wrong decision. Yet, a larger CRT renders the nodes sluggish in reacting to the changes in the network topology, whereas a smaller CRT may lead nodes to over-react to the changes in

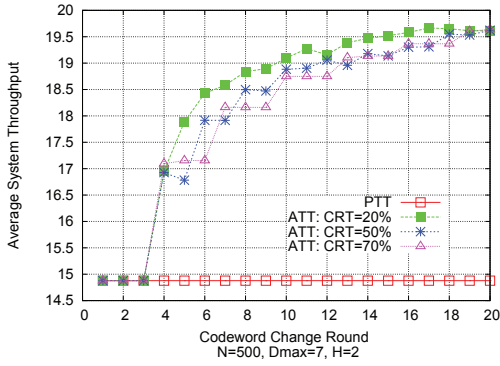


Fig. 3. Average system throughput with codeword changing under various configurations on CRT in a static network.

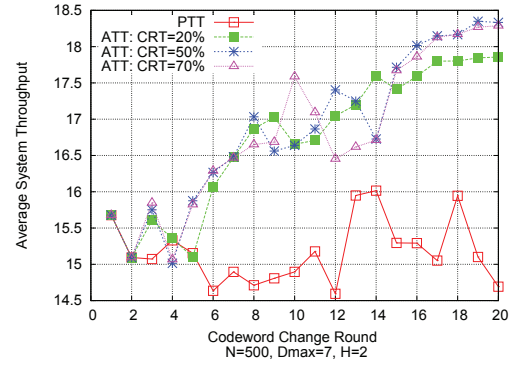


Fig. 5. Average system throughput with codeword changing under three different values of CRT in a dynamic network.

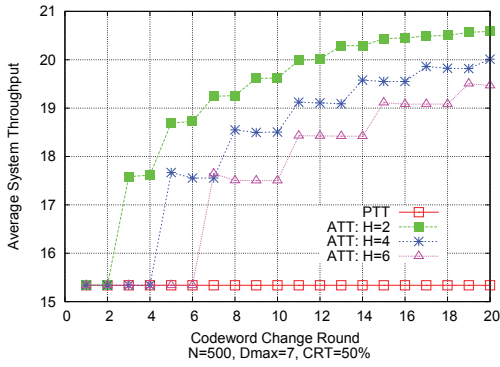


Fig. 4. Average system throughput with codeword changing under various configurations on H in a static network.

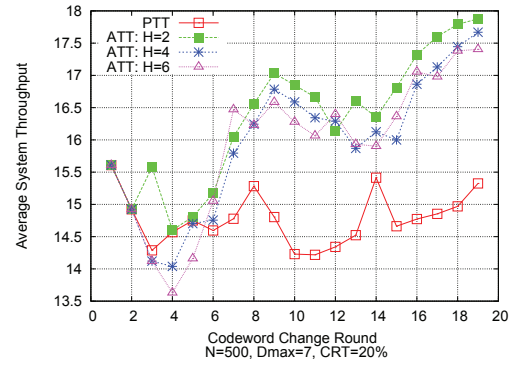


Fig. 6. Average system throughput with codeword changing under three different values of H in a dynamic network.

the network topology. Thus, an intermediate value of CRT with the smallest H gives the best performance improvement. In Figs. 5-6, the performance of ATT is evaluated with various configurations of CRT and H . Our proposed ATT algorithm outperforms PTT since codeword reassignment can reduce the number of transmission collisions. However, the performance of ATT in a dynamic network is not improved as much as that in a static network, since node movements change the network topology.

improvement among various configurations are not obvious. It is because the network topology changes are so fast that the measured collision rate becomes too stale to reflect the current state of the network topology. In Figs. 7-8, the performance of ATT is evaluated with various configurations of CRT and H . Although the performance for the highly dynamic network is not improved as much as that in the static network, ATT outperforms PTT since the codeword reassignment attempts to match the network environment better.

TABLE II
COMPARISON OF THROUGHPUT IMPROVEMENT UNDER A DYNAMIC NETWORK

	$H = 2$	$H = 4$	$H = 6$
$CRT = 20\%$	16.15%	12.96%	11.99%
$CRT = 50\%$	18.84%	14.47%	12.31%
$CRT = 70\%$	18.44%	13.40%	9.71%

3) *Highly dynamic network*: Table III shows the throughput improvement of ATT over PTT with nine different configurations of CRT and H in a highly dynamic network. As expected, the performance of ATT is better than PTT. However, the performance improvement yielded by ATT in a highly dynamic network fades with respect to those in a static network and a dynamic network. Furthermore, the differences in performance

TABLE III
COMPARISON OF THROUGHPUT IMPROVEMENT UNDER A HIGHLY DYNAMIC NETWORK

	$H = 2$	$H = 4$	$H = 6$
$CRT = 20\%$	3.15%	2.87%	2.01%
$CRT = 50\%$	3.14%	2.83%	1.94%
$CRT = 70\%$	2.76%	2.43%	1.74%

C. Simulation Summary

1) *Effect of CRT and H* : Based on the simulation results, ATT can significantly improve the performance with appropriate choices of CRT and H . For a static network, nodes with smaller CRT and H are more likely to find the more appropriate codewords to match the network topology. For a

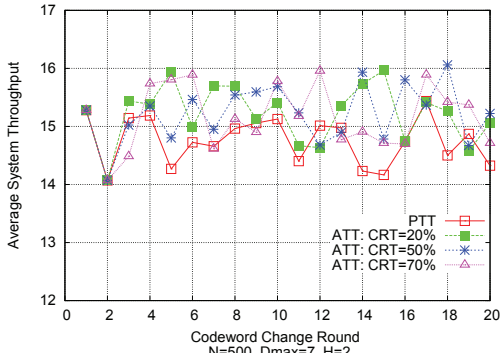


Fig. 7. Average system throughput with codeword changing under various configurations on CRT in a highly dynamic network.

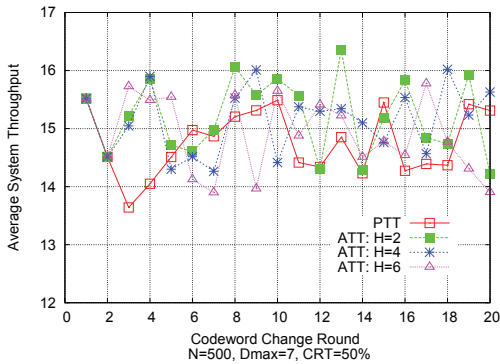


Fig. 8. Average system throughput with codeword changing under various configurations on H in a highly dynamic network.

dynamic network, a smaller H and an intermediate CRT would allow nodes to react to the changes in the network topology. Lastly, for a highly dynamic network, the performance of ATT is not sensitive to H and CRT . However, nodes with smaller H and CRT may change codewords more frequently than expected.

2) *Effect of N and D_{max}* : The effect of the number of nodes N and the maximum node degree D_{max} on ATT compared with PTT is also evaluated in the simulation. Here, we only show the simulation results for a dynamic network since the trends for the other two types of networks are similar. H and CRT are set to two and 50%. As shown in Fig. 9(a), N is configured with five different values, namely, 32, 64, 128, 256, and 512. D_{max} is set to seven. The performance of ATT is better than PTT with increasing number of nodes N , since the codeword of each node is changed to match the network environment better when using ATT. Fig. 9(b) shows the effect of D_{max} on ATT and PTT. Here, N is 500 and D_{max} is configured with 11 different values, ranging from seven to 17. ATT also outperforms PTT. However, by increasing D_{max} , the performance drops more quickly since a larger D_{max} implies a higher chance for transmission collisions among nodes.

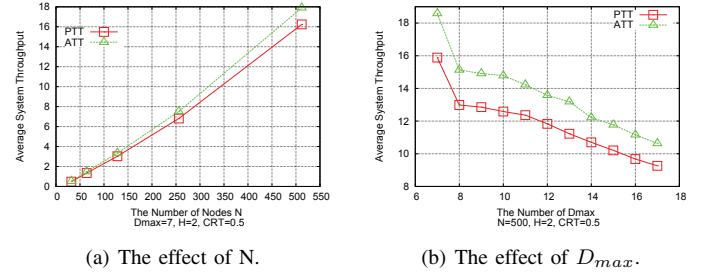


Fig. 9. The effect of N and D_{max} on the average system throughput in a dynamic network.

V. CONCLUSION

In this paper, we have proposed an adaptive topology-transparent distributed scheduling algorithm (ATT) in wireless networks. ATT does not require any information exchanges among nodes, but is able to adapt to network topology changes to produce a more efficient transmission schedule. In ATT, the collision rate of each selected time slot is compared with the given collision rate threshold CRT . If the collision rates of some selected time slots are larger than CRT , another available codeword is selected to generate a new transmission schedule so that these selected time slots are replaced. The simulation results show that ATT outperforms the existing pure topology-transparent algorithm (PTT) in all tested network scenarios. We also find that ATT is robust because it is insensitive to the total number of nodes N .

REFERENCES

- [1] Z. Cai, M. Lu, and C. Georgiades, "Topology-transparent time division multiple access broadcast scheduling in multihop packet radio networks," *IEEE Transactions on Vehicular Technology*, vol. 52, no. 4, pp. 970–984, Jul. 2003.
- [2] I. Chlamtac and A. Farago, "Making transmission schedules immune to topology changes in multi-hop packet radio networks," *IEEE/ACM Transactions on Networking*, vol. 2, no. 1, pp. 23–29, Feb. 1994.
- [3] F. Farnoud and S. Valaee, "Reliable broadcast of safety messages in vehicular ad hoc networks," in *Proc. IEEE INFOCOM'09*, April 2009, pp. 226–234.
- [4] J.-H. Ju and V. O.K. Li, "An optimal topology-transparent scheduling method in multihop packet radio networks," *IEEE/ACM Transactions on Networking*, vol. 6, no. 3, pp. 298–306, Jun. 1998.
- [5] K. Oikonomou and I. Stavrakakis, "Analysis of a probabilistic topology-unaware TDMA MAC policy for ad hoc networks," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 7, pp. 1286–1300, Sep. 2004.
- [6] R. M. Roth, *Introduction to coding theory*. Cambridge University Press, 2006.
- [7] Q. Sun, V. O.K. Li, and K.-C. Leung, "Topology-transparent distributed scheduling in multi-hop wireless networks," in *Proc. IEEE GLOBE-COM'08*, Dec. 2008.
- [8] T.-C. Hou and V. O.K. Li, "Transmission range control in multihop packet radio networks," *IEEE Transactions on Communications*, vol. 34, no. 1, pp. 38–44, Jan. 1986.
- [9] J. Ariyakhajorn, P. Wannawilai, and C. Sathitwiriyawong, "A comparative study of random waypoint and gauss-markov mobility models in the performance evaluation of manet," in *Proc. IEEE ISCIT '06*, Sep. 2006, pp. 894–899.
- [10] E. Zola and F. Barcelo-Arroyo, "Impact of mobility models on the cell residence time in WLAN networks," in *Proc. IEEE SARNOFF '09*, Apr. 2009.