

# A Framework for Opportunistic Allocation of Wireless Resources\*

Yun Hou and Kin K. Leung  
Electrical and Electronic Engineering Dept.  
Imperial College  
London SW7 2BT, United Kingdom  
Email: {yun.hou, kin.leung}@imperial.ac.uk

**Abstract** – Wireless multi-hop, mesh networks are being considered as a candidate to backhaul data traffic from access networks to the wired Internet. To enhance system performance, scheduling algorithms for wireless mesh networks are desirable to take advantage of multi-user diversity resulted from time-varying channel condition and space-varying path loss. Although many existing scheduling algorithms or medium access protocols have been adopted for the wireless mesh networks, they do not perform well, given that the algorithms are devised for wireless access. In this paper, we study the computational complexity in finding the optimal schedule for a mesh network with time-division-duplexing (TDD) operations. We propose a novel heuristic distributed scheduling framework for wireless mesh networks with open definitions of utility function. Performance analysis shows that our proposed framework is of polynomial-time complexity. Simulation results compare our framework with the tree-structural approach, and reveal that our proposed framework is highly capable of selecting and scheduling links with high utility in a fully distributed manner.

## I. INTRODUCTION

Backhaul networks transfer data between access points to gateway nodes, which in turn are connected to the wired Internet. Traditionally, backhaul networks are based on wired technologies such as ADSL, T1 and optical fiber. For low cost and ease of deployment, wireless networks are being considered to provide the backhaul capability [1]. Specifically, we consider here wireless mesh networks (WMNs) for the backhaul application where data can be forwarded through multiple hops before reaching the desired gateway node. A key design challenge for the wireless backhaul networks is to provide very high network throughput while meeting the quality-of-service (QoS) requirements for various data flows. As shown in Figure 1 for example, backhaul traffic is collected from a number of sources (e.g., access points, base stations) which could be randomly located. Therefore, one of the approaches to overcoming this challenge is to use advanced scheduling algorithms to realize the throughput gain and ensure QoS.

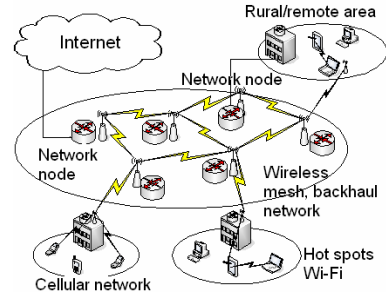


Figure 1 An example for wireless backhaul networks [1]

Scheduling for WMNs has drawn a lot of research attention recently. The distributed coordination function (DCF) with the request-to-send (RTS) and clear-to-send (CTS) mechanism proposed in IEEE 802.11 ad-hoc mode [2] has been commonly used as a candidate scheduling algorithm for wireless ad-hoc networks, because the algorithm basically selects the active transmissions in ad-hoc networks. In addition, the election-based scheduling algorithm specified in the IEEE 802.16 standard [3] is another recommended scheduling scheme for WMNs. Both 802.11 and 802.16 scheduling algorithms are fully distributed and collision free. Another scheduling technique referred as the tree-structure (TREE) method has been proposed in [4] for backhaul networks. The method maps nodes in the mesh network into a tree such that the gateway node to the Internet corresponds to the root of the tree. Nodes at the even and odd tree levels have opposite duplexing status. That is, at a given time, all nodes of the same (even or odd) level are allowed to transmit and each of them chooses one of its parent or children nodes as the destination of the data transmission. In this way, the TREE method is distributed and partially opportunistic.

These existing scheduling algorithms do not provide the optimal performance for multihop mesh networks because they do not sufficiently exploit the multi-user diversity. In WMNs, opportunistic gain could be attributed to multi-user diversities in terms of time-varying channel condition, space-varying path loss and space-time-varying interference. Unfortunately, given that 802.11 and 16 were originally devised for random radio access, neither of the access

\*This research is financially supported by the EU IST FP6 MEMBRANE Project (contract number 027310).

algorithms takes advantage of multi-user diversity in wireless environments. With regard to the TREE method, the main shortcoming is that the tree-mapping only considers part of the network links as scheduling candidates, and omits some “horizontal links” among nodes closely located physically but mapped into different branches of the tree. Omitting such candidate links in scheduling reduces the degree of multi-user diversity and hence limiting the opportunistic scheduling gain.

In this paper, we focus on the opportunistic scheduling for WMNs especially with time-division-duplexing (TDD) operations. The reason we consider TDD is that TDD is less expensive than FDD in terms of equipment cost and TDD can provide broader instant bandwidth than FDD when needed. Furthermore, nodes cannot transmit and receive at the same time in TDD systems and thus the scheduling problem for TDD mesh networks is more complicated than that for FDD networks. As a consequent, scheduling frameworks optimized for TDD systems would be readily applicable to FDD systems. In this paper, we show that the complexity of finding an optimal schedule for TDD WMNs based on comparison between maximal matchings of the network is #P-complete. Then we propose a novel heuristic distributed framework for scheduling in WMNs, and demonstrate the performance gain by simulation.

The remainder of this paper is organized as follows. In section II, complexity issues of the optimal scheduling for TDD mesh networks are discussed. Section III presents the novel heuristic framework and elaborates the computation complexity. Section IV shows numerical results to portray the properties of the proposed framework, and section V concludes the paper.

## II. #P COMPLETENESS

We consider a mesh network composed of backhaul nodes and directional links among them. The network nodes are randomly located, and two oppositely directional links exist between two nodes only when they are physically sufficiently close to each other. We assume each directional link is associated with a utility which represents the benefit of activating (i.e., transmitting on) this link in a time slot. It is worth noting that the “utility” here must be a comprehensive representation of both throughput and QoS achievement. However, we do not investigate the precise utility definition in this paper. Instead, our focus here is to find a scheduling framework applicable to any utility definitions for TDD WMNs. Thus, for the network with TDD operations, a feasible schedule is a conflict-free link set where a node can only be involved in at most one active link, i.e., scheduled links do not share common end nodes. Under this assumption, we define the optimal schedule for TDD WMN to be the set of conflict-free links which generate the highest aggregated utility for the network among all possible conflict-free link combinations. Then, the scheduling problem is formally formulated as follows.

Given a random network composed of a node set  $\mathbf{N} = \{n_i\}, i = 1, 2, \dots, N$  and a link set,  $\mathbf{L} = \{l_{ij}\}, i, j = 1, 2, \dots, N$ , and each link  $l_{ij}$  is assigned with a time-varying utility

value  $U_{ij}$ , then the optimal schedule is the link set  $\Xi^*$  which maximizes  $\sum_{l_{ij} \in \Xi} U_{ij}$  over all feasible schedules  $\Xi$ .

Before proceeding to the proof, let us state some mathematical concepts as follows.

**#P-complete** [5]: It belongs to the set of counting problems associated with NP decision problems. A #P problem must be at least as hard as the corresponding NP problem.

**Matching** [6]: A matching  $M$  in graph  $G$  is a set of non-adjacent edges; that is, no two edges in the set share a common vertex.

**Maximal matching**: A maximal matching is a matching  $M$  such that if any edge not in  $M$  is added to  $M$ , it is no longer a matching.

**Perfect matching**: A perfect matching is a matching which covers all vertices of the graph. Perfect matching is a subset of maximal matching.

From a graph-theoretic perspective, if we map the network into a graph such that the vertex set and the edge set correspond to the node set  $\mathbf{N}$  and the link set  $\mathbf{L}$  respectively, each feasible schedule  $\Xi$  is a matching of the graph, and the optimal schedule is one of the maximal matchings which maximizes the objective function. Then one way to identify the optimal schedule is to compare  $\sum_{i \in \Xi} U_i$  for all possible maximal matchings  $\Xi$ . In other words, it is a centralized method enumerating all maximal matchings.

*Theorem 1*: To enumerate all the maximal matchings for a graph is #P-complete.

**Proof**: It is known that the permanent of the adjacency matrix of a graph  $G$  counts the number of perfect matchings of the graph [6]. However, to compute the permanent for a, even well-defined, matrix is #P-complete [7]. Therefore, to identify the number of perfect matchings for a graph is #P-complete. Because the problem “to enumerate all the perfect matchings” can be reduced to the problem “to tell the number of perfect matchings”, i.e., the number of perfect matchings can be found by a subroutine of the algorithm enumerating out all the perfect matchings of a graph, then to enumerate all the maximal matchings for a graph is #P-complete. Since perfect matchings is a subset of maximal matchings, if one knows all the maximal matching of a graph, he/she will be able to enumerate all the perfect matchings as well. Therefore, to enumerate all the maximal matchings can be reduced to the problem of enumerating all the perfect matchings which is #P-complete. In other words, to enumerate all the maximal matchings for a graph is at least #P-complete. ■

Theorem 1 shows that there is no efficient way to find out all the maximal matchings for a graph in polynomial time. This further suggests the high complexity of the scheduling method which optimizes the aggregated utility by searching over maximal matchings. Based on this fact, we propose a distributed scheduling algorithm in this paper.

### III. NEW SCHEDULING FRAMEWORK

In this section, we propose a novel distributed opportunistic scheduling framework for WMNs. The network setting has been defined in last section. Specifically, we assume:

- Each node schedules (or activates) one of the links associated with it for transmission, if any, at a time.
- For each link there is a utility value to represent the benefit of its being scheduled.
- Utility values of both incoming and outgoing links are available to the associated node, and the two ends of each link keep the same, latest utility value for making scheduling decision.

The purpose of our scheduling is to identify the specific direction (to or from which neighboring node) of the next communication in an opportunistic and distributed manner. Based on the utility function in use, for each node our proposed algorithm selects the links with the highest utility among its entire incoming and outgoing links as the first step. As a result,  $N$  nodes choose  $N$  links separately as their initial decisions. Because we assume that the two ends of a link keep the same utility value for that link, it is likely that two neighboring nodes may choose the same link to activate. Then ideally,  $N$  nodes would in pairs choose  $N/2$  links to achieve the highest aggregated utility in a distributed manner.

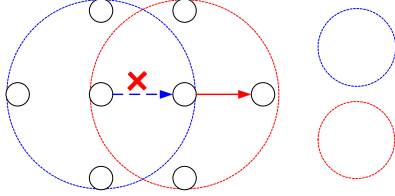


Figure 2 A conflict situation

However, as discussed above, a schedule is feasible only when it selects a link combination as a matching of the graph. If each node has no knowledge about its neighbors' duplexing status before its decision making, it is possible for the node to make a decision conflicting with one or more neighbors. Consider Figure 2 as an example where both node  $i$  and  $j$  have 4 nodes (8 links in both directions) located within their transmission range respectively. Assume that node  $i$  identifies that it is most profitable to transmit to the neighboring node  $k$ . At the same time, another neighboring node  $j$  decides to send data to node  $i$  because link  $j$  to  $i$  has the highest utility from the perspective of node  $j$ . Due to the TDD duplexing mode, node  $i$  cannot transmit and receive at the same time, and hence it is necessary to solve the conflict. Our proposed framework includes a stage to solve the conflicts. As our initial approach, our framework requires a node to give up its initial transmission after it finds out that the intended receiving node has also decided to transmit instead of receiving. The reason is the following. As shown in the figure, from the point of view of node  $i$ , the transmitting link (the solid arrow), will be more profitable than the receiving link (the dashed arrow), because it is the only reason why node  $i$  chose the solid-arrow link to transmit in the next data slot rather than the dashed one.

In the following a formal description of our framework is presented, followed by our proposal of a timeframe structure for control information exchange and the analysis of computation complexity.

#### A. The 4-phase scheduling framework

**1. Utility exchange.** Each node exchanges the utility function of each of its incoming and outgoing links with its one-hop neighboring nodes.

**2. Initial decision.** Each node chooses one of its links with the largest utility to be the initial decision of next transmission.

**3. Initial decision exchange.** Nodes with an initial decision of "transmit" – the best link of that node is an outgoing link – broadcast the initial "transmit" decision to all its one-hop neighboring nodes via a control channel (to be discussed below). The control message indicates the identities of the intended transmitting (origin) and receiving (destination) nodes associated with the initial decision.

**4. Final decision.** Each node with an initial decision of "transmit" checks if the desired receiving node also has the same "transmit" initial decision based on the control message exchanges in the Phase 3. If so, the node gives up the intended transmission. Otherwise, the node starts transmission in that direction in next slot for data traffic. Similarly, each node with an initial decision of "receive" finds out the best transmitter based on the control messages in the Phase 3, and configures its radio ready to receive data from that direction in the next time slot.

#### B. Utility exchange.

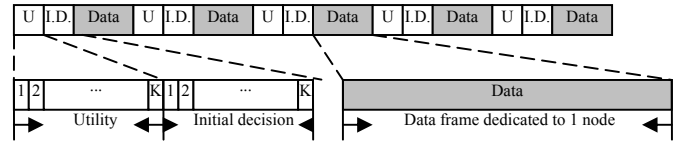


Figure 3 Timeframe structure for control message exchange.

The proposed scheduling framework requires exchanges of two types of control messages. One contains the utility value of each directional link, which is obtained by the transmitting node of the link based on the feedback information from the receiving end. The other one is the initial decision message of each node. Exchanges of initial decisions start after the exchanges of utility finish. A timeframe structure for the control message exchanges is presented in Figure 3. The timeframe consists of three consecutive sub-frames for transmissions of utility function values, initial decisions and user data. The first two sub-frames form the control channel, while the last sub-frame represents the data channel for user traffic. Each utility or initial-decision sub-frame has  $K$  mini-slots, each of which can carry a control message. All nodes in the WMN are assigned with a label from 1 to  $K$ . Nodes with label  $i$  are allowed to send their control messages in mini-slot  $i$  for  $i=1$  to  $K$ . Labels can be assigned to nodes by an algorithm similar to the channel assignment by graph coloring in cellular networks. The idea is to have neighboring nodes assigned with different labels so that they can send control messages simultaneously causing minimal interference. Now with this

feasible control exchange mechanism, the main scheduling framework is feasible. We also note that the proposed scheduling framework in Figure 3 can be readily modified to schedule multiple time slots for transmissions by multiple nodes in a given data frame

### C. Computation complexity

Consider that a node  $i$  has  $r$  neighbors to exchange utility and data, then there are  $2r$  links between node  $i$  and its neighbors. Every time slot, node  $i$  performs the scheduling procedure including: a) collecting utility values for every outgoing and incoming link (U-out and U-in), b) finding the maximum among the utilities (Max), c) transmitting its own initial decision (Tx-I.D.), d) receiving initial decisions from its neighbors (Rx-I.D.) and e) checking whether conflicts exist to make the final decision (Conflict?). Assuming that all the primitive operations, e.g., store, fetch and compare, take the same amount of time  $t$  to complete and that node  $i$  has  $r$  neighbors, we present the run time for the scheduling procedure for node  $i$  in Table 1.

Expression	U-out	U-in	Max	Tx-I.D.	Rx-I.D.	Conflict?	Total
Run time	$2r \cdot t$	$2r \cdot t$	$14r \cdot t + t$	$4r \cdot t$	$4r \cdot t$	$6r \cdot t$	$(32r + 1)t$

Table 1 Run time for the proposed framework.

The total run time for the one-slot scheduling procedure shown in Table 1 is the minimum length for one utility mini-slot plus one initial decision mini-slot illustrated in Figure 3. Due to the fact that one node has at most  $N-1$  neighbors, it takes less than  $K \cdot [32(N-1) + 1]$  for the whole network to accomplish the signaling of the scheduling framework for one data frame transmission. Since  $K$  is much smaller than the total number of nodes  $N$  in the network, which is the problem size, the run time for our scheduling framework is much smaller than  $N \cdot [32(N-1) + 1]$ . Given that  $N \cdot [32(N-1) + 1]$  is in the order of  $N^2$ , so far we have demonstrated that the run time of our proposed framework is of polynomial-time computation complexity. Compared with the likely non-polynomial (#P-complete) run time of the centralized algorithm, our proposed framework is faster and requires fewer computation resources.

## IV. SIMULATION RESULTS

A system of various numbers of nodes for two network topologies is simulated. One is the random topology where all nodes are randomly placed in an area. Nodes become neighbors if the distance between them is smaller than a pre-specified value. The other one is the regular hexagon cellular layout where base stations at the center of cells compose the mesh backhaul network. In both layouts, nodes are assumed to be able to communicate with their one-hop neighboring nodes only. However, a receiving node can be interfered by its non-neighboring nodes. The system is time-slotted and all nodes operate in the TDD mode. It is assumed that nodes are time synchronized so that each node knows the sub-frame and mini-slot boundary. The channel between any two nodes has independent Rayleigh fading with Doppler frequency of 5 to 20 Hz to approximate the radio environment with stationary topologies. To gather accurate statistics, the simulation is run

for 1,000 scenarios with nodes randomly placed in the area and each scenario runs for 1,000 time slots or frames.

The proposed algorithm is compared with the tree-structure (TREE) schemes which is a simplified version of the tree-mapping in [4]. In the TREE scheme, a root is randomly chosen for each topology. The position of nodes is the same as that for the proposed algorithm. In addition, nodes are mapped to different levels of the tree according to their distance from the root. Nodes at level  $n$  are children of the nearest node at level  $n-1$ . A node can only communicate with either its parent or children. The duplexing status of a node is determined by its level. That is, all nodes located at the same level (even or odd) of the tree have the same duplexing status (transmitting or receiving). In every time slot, either all nodes at the even levels transmit while nodes at the odd levels receive, or vice versa.

Although the optimal utility definition of the distributed scheduling for WMNs is still an open issue, we adopt two commonly used opportunistic utility definitions to illustrate the performance of our scheduling framework, namely the Signal-to-Interference-Ratio (SIR) and the Proportional-Fair (PF) [8;9] utilities. In particular, besides the comparison between different scheduling frameworks, we also examine the properties of these two commonly used utility definitions, which reflect the intrinsic tradeoffs between fairness and network throughput.

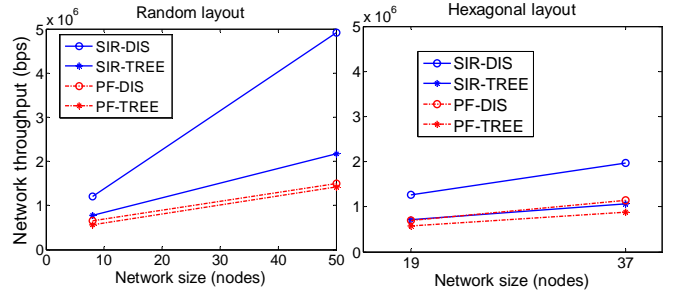


Figure 4 Comparison of network throughput with various network size.

First we compare the network throughput of two scheduling schemes with varying network size in both random and hexagonal network layouts with Doppler frequency of 20 Hz. The performance of the proposed algorithm is uniformly better than that of the TREE as shown in Figure 4. Especially in the case of random network topology using SIR as the utility function, the proposed algorithm provides a throughput enhancement of 150% for the 50-node network. The result also suggests that the larger the network is in terms of the number of nodes, the greater the performance gain of the proposed framework is. This is so because the network aggregated instantaneous throughput can be seen as a measurement of the opportunistic gain achievement. As the number of candidate links increases in large networks, the proposed opportunistic algorithm will exploit them for maximal performance gain

As shown in Figure 4, the proposed algorithm offers reduced performance gain over TREE in the case of hexagon topology when compared to the random topology. The reason is as follows. In the hexagon topology the distance between



any two neighbors are identical. Thus, there is no variation in propagation gains and hence leading to little room for the opportunistic algorithm to exploit.

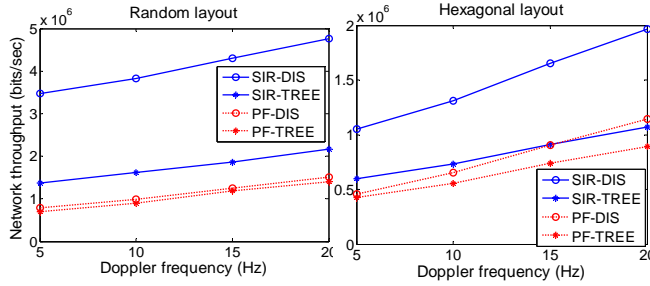


Figure 5 Comparison of network throughput with various Doppler frequency.

Next we depict the throughput achievement of two scheduling approaches with various Doppler frequencies from 5 to 20 Hz in Figure 5. We see that the throughputs for both distributed and TREE scheduling increase as the channel fluctuates faster because as discussed previously, both the distributed and TREE methods are opportunistic approaches. However, due to the static tree mapping and the lack of horizontal links in the tree, the network throughput obtained by TREE method is uniformly below that produced by our proposed scheme. As a result, using the very greedy SIR utility, our distributed scheduling algorithm performs much better than the TREE method for both random and hexagonal network layouts in the entire Doppler frequency range of interest.

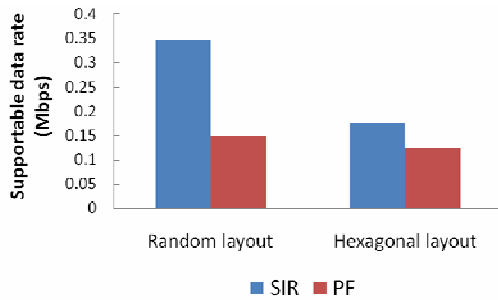


Figure 6 Average data rate of chosen links.

However, it is noticeable in Figure 4 and Figure 5 that for both random and hexagonal layouts, the PF utility provides inferior gains than the gain provided by the SIR utility. For example in Figure 4, for the hexagonal topology, the SIR utility provides our scheduling method with an improvement of 50% for the 37-node network, whereas our scheduling with the PF utility only enhances the throughput over TREE method by about 20%. We explain the reason as follows. First of all, since the PF is not a pure opportunistic utility definition, it is not as “greedy” as the SIR utility. The goal of the PF is not purely to enhance the instantaneous throughput of the chosen link set, but also to maintain some fairness among links at the same time. Another reason is that since the PF considers fairness, it includes one more varying factor, “the smoothed obtained throughput”, into the utility update, and therefore makes the prediction of the future utility value for a given link, particularly due to interference fluctuation, more

difficult relative to the SIR cases. Prediction errors in future utility may significantly degrade the scheduling performance [10] of the distributed scheduling algorithm. As a result, the scheduler using the PF metric chooses links with smaller instantaneous supportable data rate, compared to the scheduler using the SIR metric. This is shown by Figure 6 which depicts the loss in average supportable data rate of the scheduled links of the PF schedulers relative to the SIR schedulers. This generally shows the tradeoffs between fairness and instantaneous network throughput.

## V. CONCLUSIONS

In this paper, we have proposed a novel scheduling framework for wireless mesh networks with open utility function definitions. The new scheduling framework is fully distributed and capable of selecting a combination of high utility links to achieve opportunistic gain with polynomial computational complexity. Simulation results have revealed that our algorithm is able to enhance the instantaneous network throughput by up to 150%, relative to the TREE method. In the near future, we plan to extend the proposed framework to consider networks equipped by multiple antennas such as MIMO antenna and the appropriate definitions of the utility functions.

## REFERENCES

- [1] "IST Project Summary: Multi-Element Multihop Backhaul Reconfigurable Antenna Network," found at [http://www. ic. ac. uk/membrane/documents/MEMBRANE\\_Project\\_Summary. pdf](http://www.ic.ac.uk/membrane/documents/MEMBRANE_Project_Summary.pdf), 2006.
- [2] "IEEE Std 802.11-1997 Information Technology- telecommunications And Information exchange Between Systems-Local And Metropolitan Area Networks-specific Requirements-part 11: Wireless Lan Medium Access Control (MAC) And Physical Layer (PHY) Specifications," IEEE Std 802. 11-1997, p. i-445, 1997.
- [3] "IEEE Std. 802.16-2001 IEEE Standard for Local and Metropolitan area networks Part 16: Air Interface for Fixed Broadband Wireless Access Systems," IEEE Std 802. 16-2001, pp. 0-322, 2002.
- [4] G. Narlikar, G. Wilfong, and L. Zhang, "Designing Multihop Wireless Backhaul Networks with Delay Guarantees," *Proceeding of IEEE INFOCOM*, Barcelona, Spain, 2006.
- [5] M. R. Garey and D. S. Johnson, *Computers and intractability: a guide to the theory of NP-completeness*, W. H. Freeman, 1979.
- [6] I. Lovasz and M. D. Plummer, *Matching Theory*, North-Holland, 2007.
- [7] L. G. Valiant, "The Complexity of Computing the Permanent," *Theoretical Computer Science*, pp. 189-201, 1979.
- [8] T. Klein, K. K. Leung, and H. Zheng, "Improved TCP Performance in Wireless IP Networks through Enhanced Opportunistic Scheduling Algorithms," *IEEE Global Telecommunications Conference (GLOBECOM)*, pp. 2744-2748, 2004.
- [9] D. N. C. Tse, "Multiuser Diversity in Wireless Networks: Smart Scheduling, Dumb Antennas and Epidemic Communication," Minneapolis, MN, 2001.
- [10] Y. Hou and K. K. Leung, "Performance Impacts of Erroneous Channel Predictions on Packet Scheduling in Wireless Networks," *Proceeding of Global Mobile Congress (GMC)*, Beijing, 2006.