Link Scheduling in Multi-Transmit-Receive Wireless Networks

Hong-Ning Dai* Soung Chang Liew† and Liqun Fu‡
*Macau University of Science and Technology, Macau
hndai@ieee.org
†Department of Information Engineering, The Chinese University of Hong Kong, Hong Kong
soung@ie.cuhk.edu.hk
‡Institute of Network Coding, The Chinese University of Hong Kong, Hong Kong
lqfu@inc.cuhk.edu.hk

***Abstract*—This paper investigates the problem of link scheduling to meet traffic demands with minimum airtime in a multi-transmit-receive (MTR) wireless network. MTR networks are a new class of networks, in which each node can simultaneously transmit to a number of other nodes, or simultaneously receive from a number of other nodes. The MTR capability can be enabled by the use of multiple directional antennas or multiple channels. Potentially, MTR can boost the network capacity significantly. However, link scheduling that makes full use of the MTR capability must be in place before this can happen. We show that optimal link scheduling can be formulated as a linear program (LP). However, the problem is NP-hard because we need to find all the maximal independent sets in a graph first. We propose two computationally efficient algorithms, called Heavy-Weight-First (HWF) and Max-Degree-First (MDF) to solve this problem. Simulation results show that both HWF and MDF can achieve superior performance in terms of runtime and optimality.**

I. Introduction

This paper concerns the problem of link scheduling to minimize airtime usage in a new class of wireless networks called multi-transmit-receive (MTR) wireless networks. In an MTR network, a node can simultaneously transmit to a number of other nodes, or simultaneously receive from a number of other nodes. However, a node cannot simultaneously transmit and receive (i.e., the half-duplexity is still in place). Enabling this capability of MTR networks is the use of multiple directional antennas at a node [1]–[4] or the use of multiple channels on multiple radios at a node [5]. Potentially, this capability can increase the network capacity significantly [1], [2], [5] by scheduling more wireless links than conventional wireless networks. Take Fig. 1 as an example. In this four-node network, the mutually connected nodes 1 and 2 are connected with nodes 3, which is in turn connected with node 4. Since there are four edges, there are totally eight directional links. Denote the link set by $\{(1,2),(1,3),(2,1),(2,3),(3,1),(3,2),(3,4),(4,3)\}$. For a conventional network, where multiple simultaneous transmissions or receptions at a node are not allowed, at most two links can be active at a given time (e.g., $(1,2)$ and $(4,3)$). However, an MTR network allows three links to be active simultaneously (e.g., $(1,2)$,$(1,3)$,$(4,3)$).

This paper considers the link-scheduling problem of determining the minimum Time Division Multiple Access (TDMA)

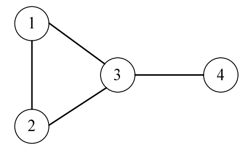

Fig. 1. A four-node network

frame length while fulfilling the traffic demands in MTR networks. Although there have been prior studies on MTR networks [1]–[4], [6], [7], this particular link scheduling problem (which is a well studied classical problem under the context of non-MTR networks [8]–[13]) has not been investigated as far as we know. Previously proposed MTR MAC protocols, such as 2P [2], WiLDNet [1] and JazzyMAC [4], are not efficient in that a node needs to maintain all of its links in transmit mode for the same time duration regardless of the actual link traffic demands.

The primary research contributions of our paper are summarized as follows.

We provide a formal specification of an MTR network, and formulate the link-scheduling problem of determining the minimum air time required to meet the traffic demands. We show that solving the link scheduling problem is NP-hard since we need find all the maximal independent sets (MIS).

To tackle this problem, we propose two computationally efficient heuristic algorithms: a Heavy-Weight-First (HWF) algorithm, which gives priority to the links with the heaviest traffic demands in its schedule and a MAX-Degree-First (MDF) algorithm, which gives priority to the links with the maximum degree in a conflict graph in its schedule.

We conduct extensive simulations based on regular and random network topologies, with symmetric and asymmetric traffic demands. The simulation results show that both HWF and MDF can obtain solutions within 90% of the optimal solutions over 1,000 simulation experiments.

The rest of the paper is organized as follows. In Section II, we present the network model, basic assumptions and problem formulation. Section III presents two heuristic algorithms. We show the simulation results in Section IV and conclude the paper in Section V.

II. Network Model and Problem Formulation

Definition 1: In a Multi-Transmit-Receive (MTR) network, each node has a set of neighbor nodes with whom it forms links. At any given time,

R1. A node can transmit simultaneously on a subset of its outgoing links.

R2. A node can receive simultaneously on a subset of its incoming links.

R3. A node cannot do operations R1 and R2 simultaneously, (i.e., a node cannot transmit and receive simultaneously).

Given an MTR network, we are interested in how to minimize the TDMA slots required to meet the underlying link traffic demands.

A. Centralized Scheduling Problem

Let the link traffics be specified by the traffic matrix $T = [t_{ij}]$, where t_{ij} is the amount of traffic from node i to its neighboring node j. At any given time, let the set of active links in the network be indicated by an indicator matrix, $M^{(k)} = [m_{ij}^{(k)}]$, where $m_{ij}^{(k)} = 1$ if link (i, j) is active, and $m_{ij}^{(k)} = 0$ if link (i, j) is inactive.

Definition 2: An indicator matrix is called a *matching matrix* if all nodes conform to rules R1, R2 and R3.

Let us consider the four-node network as shown in Fig. 1. In this network, an example of a matching matrix is:

$$M^{(1)} = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

where matrix $M^{(1)}$ indicates that node 1 simultaneously transmits to nodes 2 and 3 while node 4 transmits to node 3. Note that the matching matrix $M^{(1)}$ is *maximal* in that you cannot turn any of its 0 elements to 1 without violating R3.

Definition 3: A matching matrix is *maximal* if none of its 0 elements can be turned to 1 (while maintaining all its 1 elements at 1) without violating the rules in definition 1.

When we consider the link scheduling, we only need to consider the maximal matching matrix. Suppose there be K maximal matching matrices. Then, the problem that we are considering is as follows:

$$\begin{array}{ll} \min & \sum_{k=1}^{K} x_k \\ s.t. & \sum_{k=1}^{K} M^{(k)} x_k \geq T \\ & x_k \geq 0 \text{ for all } k \end{array} \quad (1)$$

where x_k denotes the number of time slots allocated to maximal matching matrix $M^{(k)}$.

B. Sub-optimal Scheduling

In previously proposed MAC protocols for MTR networks networks, such as 2P [2], WiLDNet [1] and JazzyMAC [4], a node is required to maintain all of its links in transmit mode for the same time duration regardless of the link traffic demands, resulting in inefficiency. In particular, the simultaneous synchronized operations in these MAC protocols indicate that when a node transmits, none of its neighbor nodes can transmit. This constraint is equivalent to (virtually) turning R1 in Definition 1 to a more restrictive requirement, as follows:

R1′ When a node transmits, it transmits on all its outgoing links.

Constraint R1′ plus the half-duplexity in constraint R3 implies that the neighbors of a node cannot transmit at the same time. With R1′, when a node i transmits, it transmits on all outgoing links, we might replace the traffic requirements for outgoing traffic from node i, $(t_{i1}, t_{i2}, ..., t_{iN})$, by one single number, $t_i = \max_j t_{ij}$. Then $t = (t_i)$ is the traffic vector describing the transmission requirements of all nodes.

Let $S^{(l)} = (s_i^{(l)})$ be a column indicator vector in which $s_i^{(l)} = 1$ if node i transmits and $s_i^{(l)} = 0$ if node i does not transmit. $S^{(l)}$ is basically an independent set if it is to conform to R1′, R2, and R3. It suffices to consider the *maximal* independent set (MIS) in our scheduling problem. Suppose that there are L MIS. Then, the scheduling problem can be formulated as follows:

$$\begin{array}{ll} \min & \sum_{l=1}^{L} x_l \\ s.t. & \sum_{l=1}^{L} S^{(l)} x_l \geq t \\ & x_l \geq 0 \text{ for all } l \end{array} \quad (2)$$

Since Eq. (2) is defined in a more restrictive way, the solution to Eq. (1) cannot be worse than that of Eq. (2).

How are maximal matching matrices related to MIS? To establish the relation, we need to model the network with a different graph. We use a *conflict* graph to describe the relationship between two conflicting links. In this graph, each directional link is denoted by a vertex, and there is an edge between two vertices if the two associated links cannot be active at the same time.

C. Problem Restatement

In optimization problem defined in Eq. (1), we represent a matching by a matrix $M^{(k)}$ for pedagogical purposes. We now define a more economical representation.

Definition 4: A matching A in an MTR network is a subset of links that conform to R1, R2 and R3.

Definition 5: A matching is said to be maximal if it is not contained in any other matching.

Let $E = \{E_j : 1 \leq j \leq |E|\}$ be the set of all the feasible matchings. The number of time slots allocated to each feasible matching E_j is denoted by a non-negative variable u_j.

Let N be the total number of links in the network. We introduce an $N \times |E|$ incidence matrix Q with elements q_{ij} such that

$$q_{ij} = \begin{cases} 1, & \text{if link } i \text{ is in matching } E_j, \\ 0, & \text{otherwise.} \end{cases}$$

where each column in Q indicates the links in a matching.

We also convert the traffic matrix $T = [T_{ij}]$ to a vector $\mathbf{f} = (f_{ij})^T$, where $f_{ij} = T_{ij}$ for i, j such that $T_{ij} \neq 0$. Then, the problem defined in Eq. (1) can be casted as a linear

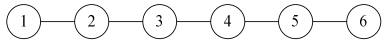

Fig. 2. The linear network

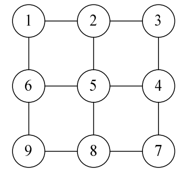

Fig. 3. The grid network

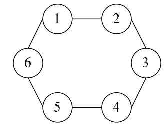

Fig. 4. The ring network

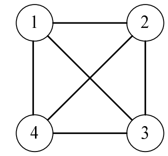

Fig. 5. The fully-connected network

program as follows:

$$\begin{aligned} \min \quad & \mathbf{e}^T \mathbf{u} \\ s.t. \quad & Q \cdot \mathbf{u} \geq \mathbf{f} \\ & \mathbf{u} \geq 0 \text{ for all } k \end{aligned} \tag{3}$$

where $\mathbf{e}$ is a vector whose components are all 1's, $\mathbf{f} = (f_1, f_2, ..., f_N)^T$ and $\mathbf{u} = (u_1, u_2, ..., u_{|E|})^T$. The difficulty of the above problem lies in how to find all matchings Q (equivalent to finding all the independent sets in the associated conflict graph, which is NP-complete). This motivates us to study heuristic algorithms to solve this problem.

III. Heuristic Algorithms

HWF is a greedy algorithm that always chooses links with the maximum traffic demand into the scheduling set during each round until all the traffic is satisfied. MDF, on the other hand, chooses links with the maximum degree in the conflict graph during each round. Both HWF and MDF make use of a conflict graph to capture constraints R1, R2 and R3. We refer readers to [14] for more details about HWF and MDF.

In Heavy-Weight-First algorithm (HWF), we first sort the links according to their traffic demands in a descending order. To construct a matching, E_i, we go through the link one by one. A link will be included into E_i if it does not conflict with the existing links in E_i according to the conflict graph. Once we have gone through all the links in the sorted list, we then identify the link in E_i with the least amount of traffic. Let us say this is link k, with traffic f_k . We then assign f_k time slots to matching E_i. We subtract f_k from the traffic of all the links in E_i, and remove link k and other links in E_i with the same amount of traffic (if any) from further consideration: there is not traffic left to be scheduled for these links. The links are then resorted according to their remaining traffics. The above process is iterated until all traffic demands are met.

In MDF, we sort the links according to their degrees in the conflict graph in a descending order. Other than the different way of sorting the links, the algorithm of MDF is essentially the same as that of HWF. In particular, at least one link will be removed at the end of each iteration. The degrees of the neighbors to this link will be updated. The remaining links will also need to be resorted accordingly before the next iteration. During each iteration, at least one link will be removed.

IV. Simulation Results

We consider several types of networks: (1) regular networks, including the linear network in Fig. 2, the grid network in Fig. 3, the ring network in Fig. 4, and the fully-connected network in Fig. 5; (2) random networks with varying degrees of connectivity.

A. *Performance in Regular Networks*

We have conducted 1,000 simulation experiments (with different traffic demands) for each of the following networks: the linear network (Fig. 2), the grid network (Fig. 3), the ring network (Fig. 4) and the fully-connected network (Fig. 5).

In order to compare the solutions obtained by the proposed algorithms with optimal solutions, we introduce the *percentage cost penalty* [9] as a performance measure, which is defined as follows:

$$P = \frac{T - T_{opt}}{T_{opt}} \times 100\% \tag{4}$$

where T denotes the total number of time slots obtained by the heuristic algorithm and T_{opt} is the total number of time slots in the optimal solution.

We compute P of HWF and MDF over the 1,000 experiments and present the averaged P values in Table I. In each experiment, we generate a random traffic demand vector $\mathbf{f}$.

The results in Table I show that MDF outperforms HWF in the linear network, the grid network and the ring network. But HWF performs better in the fully-connected network. The above results can be explained intuitively as follows. Recall that a link is removed at the end of each iteration in MDF or HWF. The nature of MDF is such that the link being removed has a high degree in the conflict graph. In this sense, MDF tends to remove many edges in the conflict graph. As a result, in a sparsely connected network (e.g., the linear network, the grid network and the ring network, many links become conflict-free after several iterations.

TABLE I

Random traffic in varied regular networks

	Average P of HWF	Average P of MDF
Linear network (Fig. 2)	5.49%	0%
Grid network (Fig. 3)	8.16%	0%
Ring network (Fig. 4)	7.97%	0%
Fully-connected (Fig. 5)	4.04%	9.15%

B. *Performance in Random Networks*

For comparison purposes, we carry out *exhaustive search* to find optimal solutions. We compare the average runtime of HWF and MDF with that of exhaustive search. To reduce the runtime of exhaustive search, we use a branch-and-bound algorithm, first proposed in [15].

We generate random network topologies. In the first set of simulations, we consider symmetric traffic demands. If there is a link between node i and node j (i.e., $g_{ij} = 1$), then the traffic

TABLE II
SYMMETRIC TRAFFIC OVER RANDOM NETWORKS

	No. of obtained solutions within 100% optimality	No. of obtained solutions with P within 10%	Average P	Average runtime (second)
Exhaustive Search	1,000	1,000	0%	1.7522
HWF	540	781	6.40%	0.0015
MDF	549	786	5.59%	0.0017

between them, $f_{ij} = f_{ji}$, is randomly generated according to the discrete uniform distribution with values ranging from 1 to 10. We conduct 1,000 experiments and present the results in Table II. Each experiment is based on one random network G and one associated random demand $\mathbf{f}$. Table II gives the statistics of the 1,000 experiments.

Table II shows that both HWF and MDF achieve reasonably good performance. In particular, Table II shows that there are nearly 800 solutions obtained by HWF and MDF with penalty cost no greater than 10%. On average, HWF and MDF have average P of 6.40% and 5.59%, respectively. Table II also shows that the average runtime of the two algorithms is much smaller than that of exhaustive search.

We have also conducted 1,000 simulations based on asymmetric traffic demands. The simulation results are presented in Table III. The traffic demand f_{ij} of each link (i, j) is randomly generated according to the discrete uniform distribution with values ranging from 1 to 10. But the traffic in the opposite direction, f_{ji} is not set to f_{ij}; rather, it is generated anew using the same distribution. It is shown in Table III that HWF outperforms MDF in this asymmetric traffic scenario. In particular, Table III shows that HWF obtain 872 solutions with P less than 10% versus 779 obtained by MDF. On average, HWF has a lower average P of 3.42% versus 5.32% of MDF.

TABLE III
ASYMMETRIC TRAFFIC OVER RANDOM NETWORKS

	No. of obtained solutions within 100% optimality	No. of obtained solutions with P within 10%	Average P	Average runtime (second)
Exhaustive Search	1,000	1,000	0%	1.8511
HWF	655	872	3.42%	0.0017
MDF	568	779	5.32%	0.0019

HWF outperforms MDF in the asymmetric case because HWF can "compact traffic demands" as it runs. By compacting traffic demands, we mean HWF can decrease the range of the traffic demands in the network after each iteration. To see this, suppose we have a traffic demand, $\mathbf{f} = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]$, where the traffic ranges from 1 to 10. Suppose that in the first iteration, links with traffic demands, 8, 9, and 10 have been chosen for scheduling. Then, we have an updated demand, $\mathbf{f} = [0, 1, 1, 2, 2, 3, 4, 5, 6, 7]$ after this iteration. Now the traffic demands to be scheduled have a narrower range (i.e., 1 to 7) in the future. With compact traffic, in the later iterations, more scheduled links can be removed in each iteration because they have the same traffic demands. MDF, on the other hand, does not have such an advantage.

V. CONCLUSION

In this paper, we have investigated MTR networks in which a node may simultaneously send to a number of other nodes; or simultaneously receive from other nodes. This capability can potentially improve the network capacity substantially. We have (i) provided a formal specification of MTR networks for a systematic study; (ii) formulated the link-scheduling problem of minimizing the airtime usage in an MTR wireless network as a linear program (LP) and demonstrated that it is NP-hard; (iii) proposed two computationally efficient heuristic algorithms, HWF and MDF to solve this LP; and (iv) presented extensive simulation results to show that both HWF and MDF algorithms achieve good optimality and runtime performance.

ACKNOWLEDGMENT

This work is supported in part by the General Research Fund (Project No: 414507) established under the University Grant Committee of the Hong Kong Special Administrative Region, China, and in part by a Direct Grant from the Chinese University of Hong Kong (Project No: 2050464).

REFERENCES

[1] R. Patra, S. Nedevschi, S. Surana, A. Sheth, L. Subramanian, and E. Brewer, "WiLDNet: Design and Implementation of High Performance WiFi Based Long Distance Networks," in *USENIX NSDI*, 2007.

[2] B. Raman and K. Chebrolu, "Design and Evaluation of a new MAC Protocol for Long-Distance 802.11 Mesh Networks," in *ACM MobiCom*, 2005.

[3] S. Surana, R. Patra, S. Nedevschi, M. Ramos, L. Subramanian, Y. Ben-David, and E. Brewer, "Beyond Pilots: Keeping Rural Wireless Networks Alive," in *USENIX NSDI*, 2008.

[4] S. Nedevschi, R. K. Patra, S. Surana, S. Ratnasamy, L. Subramanian, and E. A. Brewer, "An Adaptive, High Performance MAC for Long-distance Multihop Wireless Networks," in *ACM MobiCom*, 2008.

[5] S. Kakumanu and R. Sivakumar, "Glia: A practical solution for effective high datarate wifi-arrays," in *Proc. of ACM MobiCom*, 2009.

[6] L. Subramanian, S. Surana, R. Patra, S. Nedevschi, M. Ho, E. Brewer, and A. Sheth, "Rethinking Wireless for the Developing World," in *ACM SIGCOMM Hotnets-V*, 2006.

[7] A. Sheth, S. Nedevschi, R. Patra, S. Surana, L. Subramanian, and E. Brewer, "Packet Loss Characterization in WiFi-based Long Distance Networks," in *IEEE INFOCOM*, 2007.

[8] A. K. Das, R. J. Marks, P. Arabshahi, and A. Gray, "Power Controlled Minimum Frame Length Scheduling in TDMA Wireless Networks with Sectored Antennas," in *IEEE INFOCOM*, 2005.

[9] L. Fu, S. C. Liew, and J. Huang, "Joint Power Control and Link Scheduling in Wireless Networks for Throughput Optimization," in *IEEE ICC*, 2008.

[10] S. A. Borbash and A. Ephremides, "The Feasibility of Matchings in a Wireless Network," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2749 – 2755, 2006.

[11] J. Tang, G. Xue, C. Chandler, and W. Zhang, "Link Scheduling with Power Control for Throughput Enhancement in Multihop Wireless Networks," *IEEE Transactions on Vehicular Technology*, vol. 5, no. 3, pp. 733 – 742, 2006.

[12] L. Fu, S. C. Liew, and J. Huang, "Power Controlled Scheduling with Consecutive Transmission Constraints: Complexity Analysis and Algorithm Design," in *IEEE INFOCOM*, 2009.

[13] S. A. Borbash and A. Ephremides, "Wireless Link Scheduling With Power Control and SINR Constraints," *IEEE Transactions on Information Theory*, vol. 52, no. 11, pp. 5106 – 5111, 2006.

[14] H.-N. Dai, S. C. Liew, and L. Fu. Link Scheduling in Multi-Transmit-Receive Wireless Networks. [Online]. Available: http://arxiv.org/abs/1107.1941

[15] A. H. Land and A. G. Doig, "An Automatic Method for Solving Discrete Programming Problems," *Econometrica*, vol. 28, pp. 497–520, 1960.