Distributed Low-Complexity Maximum-Throughput Scheduling for Wireless Backhaul Networks

Abdul Kabbani
EE Dept, Stanford University
Palo Alto, CA 94305
Email: akabbani@stanford.edu

Theodoros Salonidis
Thomson Paris Research Lab
Boulogne Cedex, France 92648
Email: theodoros.salonidis@thomson.net

Edward W. Knightly
ECE Dept, Rice University
Houston, TX 77005
Email: knightly@rice.edu

Abstract— **We introduce a low-complexity distributed slotted MAC protocol that can support all feasible arrival rates in a wireless backhaul network (WBN). For arbitrary wireless networks, such a maximum throughput protocol has been notoriously hard to realize because even if global topology information is available, the problem of computing the optimal link transmission set at each slot is NP-complete. For the logical tree structures induced by WBN traffic matrices, we first introduce a centralized algorithm that solves the optimal scheduling problem in a number of steps at most linear in the number of nodes in the network. This is achieved by discovering and exploiting a novel set of graph-theoretical properties of WBN contention graph. Guided by the centralized algorithm, we design a distributed protocol where, at the beginning of each slot, nodes coordinate and incrementally compute the optimal link transmission set. We then introduce an algorithm to compute the minimum number of steps to complete this computation, thus minimizing the per-slot overhead. Using both analysis and simulations, we show that in practice our protocol yields low overhead when implemented over existing wireless technologies and significantly outperforms existing suboptimal distributed slotted scheduling mechanisms.**

I. INTRODUCTION

Wireless Backhaul Networks (WBNs) forego costly wired infrastructure via wirelessly multi-hopping to and from high-speed wired Internet entry points. We consider WBN architectures that use scheduled Medium Access Control (MAC) protocols such as IEEE 802.16. Scheduled access is attractive for backbone systems like WBNs because it not only avoids well-known starvation/unfairness phenomena of contention-based access (e.g., 802.11) but can also provide throughput and delay guarantees.

In a scheduled access MAC protocol, all nodes are time-slot synchronized and communication occurs in a sequence of frames. Each frame consists of a scheduling phase followed by a data phase, both of fixed duration and spanning several slots. During the scheduling phase, nodes must determine a set of conflict-free links that will transmit during the data phase.

We introduce a distributed scheduled access MAC protocol that can support all feasible arrival rates in a WBN. Such a maximum throughput protocol has been notoriously challenging to realize in wireless networks of arbitrary topologies for two reasons. First, selecting the optimal link transmission set can be a problem of extremely high complexity even if centralized information is available: in a seminal paper [14], Tassiulas and Ephremides established that selecting a conflict-free set of links having a maximum sum of queue sizes at each scheduling phase is a maximum throughput policy. Unfortunately, this requires solving an instance of the maximum weight independent set (MWIS) problem before each frame, which is NP-complete in general. Second, the duration of the scheduling phase should be long enough so that nodes can exchange all the messages needed in order to solve *any* instance of the optimal scheduling problem, thereby requiring a deterministic upper bound on the number of message exchange steps. This bound should also be as small as possible since it determines the scheduling phase duration, a fixed per-frame overhead during the protocol operation.

In this paper, we exploit the WBN tree structure and solve the MWIS scheduling problem in a number of steps at most linear in the number of nodes in the network.[1] We first prove that the WBN contention graph possesses a recursive reduction property, where at each reduction step it is always possible to find a link where *all* its interfering neighbors interfere with each other. We utilize this property to devise a two-phase centralized scheduling algorithm that computes a MWIS by sequentially executing at each link exactly once in each phase, regardless of the MWIS problem instance at each frame. Based on the centralized algorithm, we design a distributed protocol in which nodes converge to a MWIS using at most two-hop message exchanges. Despite being linear in terms of the number of links in the network, the bound on the number of the protocol messages exchanged can still be large in practice. We provide an algorithm that solves the offline design optimization problem of determining the minimum number of protocol message-exchange rounds required to compute any instance of the MWIS problem for a given WBN topology.

Finally, we compare our protocol to a protocol that computes a *maximal* link transmission set at each frame. For arbitrary wireless networks, maximal scheduling has been shown to guarantee at least one-eighth of all feasible arrival rates [7], [11], [16] and can be implemented using distributed protocols with a linear bound on the scheduling phase [2]. We show that maximal scheduling remains suboptimal in the case of WBNs, and our protocol yields 0.02% to 75% of the overhead of existing maximal protocols, depending on the degree and depth of the WBN topology structure.

The rest of the paper is organized as follows: In Section

[1] The tree structure is logical (for forwarding and routing) vs. physical (interference).

II, we review the state of the art in distributed scheduled access protocols for both arbitrary and tree-structured wireless networks. In Section III, we define the WBN interference model and the maximum throughput scheduling policy. In Section IV, we introduce the graph-theoretical properties of the WBN contention graph which lead to a centralized algorithm that solves the MWIS problem in linear time. We introduce the distributed protocol that realizes the centralized algorithm in Section V. We solve the overhead minimization problem in Section VI and evaluate the protocol performance in Section VII. Section VIII concludes.

II. Related work

The problem of providing throughput guarantees through distributed scheduling has received significant attention lately. Related work can be classified with respect to the topology structure considered under the following link interference constraints: either multi-channel systems where only primary interference exists (due to the half-duplex radio constraint), or single-channel systems where both primary interference and secondary interference (due to hidden terminals) exist.

Wireless networks of arbitrary topologies: Given the difficulty of the MWIS problem, previous work has focused on evaluating the performance of suboptimal approaches based on maximal scheduling. On one hand, it was shown in [7] and [16] that maximal scheduling in multi-channel networks can achieve at least 50% of the maximum throughput region for single-hop flows and multi-hop flows respectively. On the other hand, [11] showed that this guarantee decreases to 1/8 in single-channel systems.

In addition, while existing maximal scheduling algorithms describe required message exchanges and acknowledge the need for node coordination, they are not fully-specified medium access protocols. The specifications of such protocols determine the overhead induced by the scheduling phase and play an important role on overall performance. Recently, Modiano et al. introduced a distributed randomized scheduling algorithm for maximum throughput scheduling in multi-channel networks [8], however, the coordination protocol to achieve the required objective has not been specified there.

Tree-structured wireless networks: Tree-structured networks have been addressed using techniques that exploit their special structure. In [10], a distributed scheduled access protocol is designed on top of the 802.11 PHY and demonstrated better performance than that achieved by 802.11 DCF. The work in [15] focuses on packet scheduling for delay guarantees using techniques from wireline networks. Both [10] and [15] use a two-frame period where half the links are activated in each frame. This technique utilizes only two link activation sets, hence cannot realize all feasible arrival rates. The work in [12], designs an asynchronous distributed maximum throughput protocol for multi-channel systems. In [1] it is also shown that multi-channel systems allow distributed heuristics based on Longest Queue First (LQF) to be throughput optimal. However, in such heuristics, the number of steps required to compute the optimal schedule at each frame is only given in terms of average complexity. In contrast to [1] and [12], our protocol achieves maximum throughput for a broader set of interference constraints of WBNs by directly solving the MWIS problem in an upper-bounded number of steps, at most linear in the number of nodes in the WBN.

III. System Model

We represent the WBN by a Network Graph (NG) $G(N, L)$ where each edge $(u, v) \in L$ signifies that nodes u and v of the set N are within range and have established a logical wireless link. Each node has a single radio transceiver and cannot transmit and receive simultaneously. This *primary interference* constraint prohibits two links sharing a node from being scheduled simultaneously. *Secondary interference* constraints arise between links that do not share a common node but at least one of their corresponding transmitter or receiver nodes are within range. This definition of secondary interference constraint refers to a communication model where each successful link transmission requires a successful DATA/ACK exchange. Secondary interference can be mitigated by preassigning different channels (orthogonal frequency bands or spread spectrum codes) to all nodes within a two-hop neighborhood. We will call WBNs with primary interference only, *multi-channel* and WBNs with both primary and secondary interference, *single channel.*

Nodes in the WBN communicate using a logical tree structure overlaid on top of the physical topology defined by the nodes' wireless proximities. This structure can be explicitly enforced by planning. It can also be implicitly induced by cycle-free routing protocols. In this case, the WBN topology changes at the time scale of the routing updates which would span several frames. For simplicity we consider the WBN topology to be static and that secondary interference among the different branches of the WBN tree has been suppressed. This holds for multi-channel WBNs, where only primary interference exists. In the single channel case, this can be achieved by directional antennas [10] or by embedding the WBN tree in a hexagonal structure [3].

A *link transmission set* is a set of links that can transmit simultaneously subject to the WBN scheduling constraints. Since a multi-channel system has primary interference constraints only, any matching in the NG is a link transmission set (e.g., the link set $\{a, e, k, l, o, r\}$ in the NG of Fig. 1). In a single-channel system, a link transmission set must also satisfy the secondary interference constraints (e.g., the link set $\{a, h, k, l\}$ in the NG of Fig. 1).

The WBN operates with a synchronized frame structure. At each frame, the scheduling phase decides a link transmission set to transmit during the entire data transmission phase. Each link transmission set is represented by a 0-1, $|L| \times 1$ vector in which each unit entry at index l corresponds to link l scheduled for transmission. We denote the set of all link transmission sets in the WBN by $\boldsymbol{I}(G)$.

Each link l is represented by a queue maintained at its transmitter node. Let $A_l(k)$ and $Q_l(k)$ be the number of arrivals at l and the number of packets already in the queue of l at the beginning of time slot k, respectively. Let the arrival rate at link l be λ_l. An arrival rate vector $\boldsymbol{\lambda} = \{\lambda_l, l \in L\}$ is *feasible* if there exists a collection of link transmission sets

$\boldsymbol{I}_j \in \boldsymbol{I}(G)$ and numbers $\alpha_j \in [0, 1)$ such that:

$$\lambda_l \leq \sum_{j=1}^{M} \alpha_j \boldsymbol{I}_j, and, \sum_{j=1}^{M} \alpha_j < 1 \qquad (1)$$

The set of all feasible rates forms the *feasibility region* denoted by $\boldsymbol{\Lambda}$. A scheduling algorithm is a maximum throughput algorithm (or stable algorithm) if for every feasible set of arrival rates, the average queue size of each link l does not grow to infinity, that is, $\limsup_m E[Q_l(k)] < \infty$ for every link l in the WBN.

In [14], Tassiulas and Ephremides considered a synchronized slotted system where each frame consists of a single slot. They established that selecting at each slot k a link transmission set with maximum sum of queue sizes is a maximum throughput algorithm [14]:

$$\boldsymbol{I}^*(k) = arg \max_{\boldsymbol{I} \in \boldsymbol{I}(G)} \boldsymbol{Q}^T(k)\boldsymbol{I} \qquad (2)$$

The result was shown to hold for arbitrary topologies, arbitrary link scheduling constraints and i.i.d arrival processes. It has also been extended to more general classes of arrival processes, and multi-slot frames [13]. Despite its universality, this algorithm assumes that the scheduling procedure is made by a centralized controller where instant information is perfectly disseminated to and from the controller. Furthermore, to determine the optimal link transmission set requires solving a Maximum Weight Independent Set (MWIS) problem at each frame, which is NP-complete in general. In multi-channel systems, the problem reduces to a Maximum Weighted Matching (MWM) computation, which is of polynomial complexity ($O(N^3)$). To the best of our knowledge, no known *distributed* algorithm exists to exactly solve either MWIS or MWM, for any topology structure.

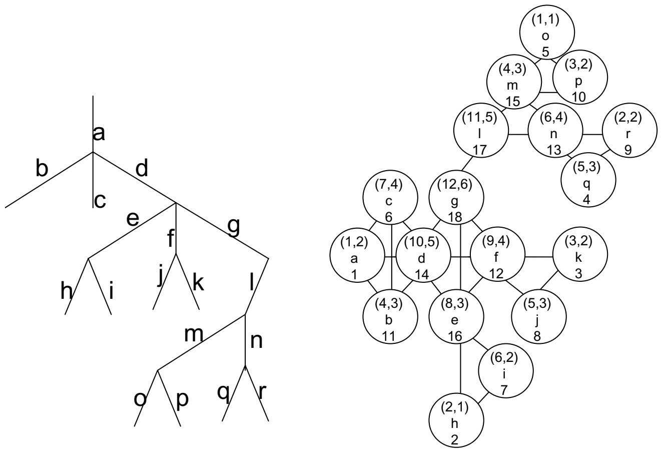
We proceed to show that the WBN structure allows to overcome the above difficulties and solve the MWIS problem distributively, in linear time, for both multi-channel and single channel systems. Due to space limitations, we refer the reader to [6] for all proofs.

IV. Linear Complexity Scheduling

In this section, we introduce a set of graph-theoretical properties particular to the link interference relationships of the WBN structure. Using these properties, we devise a centralized algorithm that can provably solve the MWIS problem in linear time.

A. Graph-theoretical properties of WBNs

The link interference relationships in the WBN can be captured by a Contention Graph (CG) $G(V, E)$. Each vertex v in the CG corresponds to a link in the NG of the WBN and each edge in the CG signifies that the two links in NG corresponding to the two endpoint vertices of the edge are interfering. The CG allows us to abstract the specific link interference constraints and treat multi-channel and single-channel systems in a unified manner. For example, the CG in Fig. 1(b) corresponds to a multi-channel WBN. An independent set in a CG corresponds to a link transmission set in its corresponding NG. Thus, solving the MWIS problem on a CG, where the weight of each vertex is equal to the corresponding link's queue backlog at the beginning of a frame, leads to maximum throughput guarantees.



a- Network Graph

b- Multi-channel Contention Graph

Fig. 1. (a) A Network Graph (NG) (b) corresponding multi-channel Contention Graph (CG). Each CG vertex corresponds to a link in NG. The integer below each CG vertex id denotes the sequence label assigned by the enumeration algorithm (Section IV-B). The integer pairs above each CG vertex id denote the optimal enumeration pair $(SEQ1, SEQ2)$ assigned by the overhead minimization algorithm (Section VI).

The graph-theoretical properties of both single-channel and multi-channel WBN CG are summarized by the following lemma:

WBN Clique Lemma: *Given a WBN CG, there is always at least one vertex v such that v and* all *its neighbors form a clique. Moreover, if v is removed from the CG, the resulting subgraph possesses the same property.*

The Clique Lemma reveals a fundamental recursive poperty of a WBN CG and can be proven by showing that it holds for CG vertices corresponding to links connected to leaf nodes in the corresponding NG. [2] For example, when the NG in Fig. 1(a) corresponds to a multi-channel system, leaf link p interferes only with links o and m and the vertices of all three links in the corresponding multi-channel CG (Fig. 1(b)) form a clique. Similarly, when the NG in Fig. 1(a) corresponds to a single-channel system, leaf link p interferes only with links o, m, l, n and it can be shown that all links form a clique in the corresponding single-channel CG.

The Clique Lemma is crucial because it determines the correct order by which certain weight-exchange operations should be executed on a per-CG-vertex basis to provably compute the MWIS. More specifically, our scheduling algorithm requires a fixed enumeration sequence, pre-computed offline by an enumeration algorithm. The enumeration algorithm utilizes the

[2] Although it is possible to prove that the Clique Lemma also holds for non-leaf links, any link satisfying the Lemma suffices to solve the MWIS problem.

Clique Lemma to assign a unique label to each CG vertex. This pre-computed enumeration sequence of labels determines the correct order in which the CG vertices should exchange their weights. The sequence is used by the MWIS algorithm to compute a MWIS at each instance of the MWIS problem. Next, we describe the enumeration and MWIS algorithms in detail.

B. Enumeration Algorithm

Given the WBN CG, the enumeration algorithm picks any CG vertex satisfying the Clique Lemma and assigns to it label 1. Next, this vertex is eliminated and the same vertex selection-elimination step repeats for the reduced CGs by assigning labels in increasing order, until all vertices of the CG have been considered. The specific steps of the algorithm are as follows.

Enumeration Algorithm: A graph ICG is initialized to graph CG. At every iteration i, and as long as ICG is not empty: (i) select an arbitrary vertex v in ICG such that v and all its neighbors constitute a clique (guaranteed to exist by the Clique Lemma), (ii) label v by the number i, and (iii) delete v and all its adjacent edges from ICG.

Fig. 1 depicts the output of the enumeration algorithm in the multi-channel WBN CG. The sequence label of each vertex is shown below its id.

C. MWIS algorithm

The centralized MWIS scheduling algorithm takes as input a weighted CG and returns an independent set. It consists of the Weight-Exchange Phase and the Independent Set Construction Phase summarized below and presented in detail in [6]:

Weight Exchange Phase: Initialize a vertex set called TRAV to be empty. Also, initialize the residual weight RW of each vertex to its weight. Based on the enumeration sequence, traverse each vertex of the CG in increasing label order. For each vertex v traversed, if its RW is positive: (i) add it to the TRAV set, (ii) decrease its RW to zero, and (iii) decrease the RW of its neighbors in CG by v's RW; otherwise skip to the next vertex.

Independent Set Construction Phase: Initialize a vertex set I to be empty. Traverse every vertex v in TRAV in *decreasing* label order. Vertex v will be included in set I as long as I does not contain any neighbor of v.

Table I illustrates the Weight Exchange Phase and Table II illustrates the following Independent-set Construction Phase for an initial set of weights in the CG of Fig. 1(b).

MWIS Theorem: *The independent set I computed by the end of the Independent-set Construction Phase of the MWIS scheduling algorithm is a MWIS.*

Proof: We start by noting that, in a vertex-integer-weighted graph, the weight of any independent set is upper bounded by the cardinality of any clique cover[3] of this graph [9]. Given this fact, we prove that it is possible to find a clique cover in the CG whose cardinality is a tight lower bound to the weight of set I. Then, the cardinality of the clique cover equals the weight of I, and I is a MWIS.

The clique cover is constructed as follows. By the Clique Lemma, the set consisting of any vertex v and all its neighbors that have been enumerated *after* v form a clique. We denote such a clique by C_v. Now, construct a clique set C containing RW_v copies of each clique C_v of vertex v in the $TRAV$ set that was built during the Weight Exchange phase - RW_v is the *residual weight* of v when visited during the Weight Exchange phase.

Clique set C is a clique cover because (i) every vertex v in the $TRAV$ set is covered by exactly as many cliques in C as its initial weight W_v (by the construction of C) and (ii) every vertex v' not it the $TRAV$ set is covered by at least as many cliques as its initial weight (otherwise, v' would have been part of the $TRAV$ set).

We now proceed to show by contradiction that the weight of I is tightly lower-bounded by the cardinality of C. Suppose I has strictly less weight than the cardinality of C. Then either (1) one vertex in I is covered by fewer cliques of C than its weight or (2) there are cliques in C which do not include any vertex in I.

Claim (1) does not hold because I is a subset of $TRAV$ and, by construction, any vertex v in $TRAV$ is covered by exactly as many cliques in C as v's weight RW_v.

Claim (2) does not hold because every clique in C covers exactly one vertex of I. This can be shown as follows. Since CG vertices are traversed in reverse when building I, every clique C_v in C contains at least one of I's vertices: if none of v's adjacent $TRAV$ neighbors with higher sequence number is in I (i.e, v has not been eliminated yet from I by the time it is traversed), then v itself would have been part of I. In addition, no clique can contain more than one vertex of an independent set.

Based on the above, C is a clique cover and a lower-bound to the weight of I. Thus, I is a MWIS. ■

Both phases of the MWIS scheduling algorithm require a number of iterations equal to the number of nodes in CG, which is in turn equal to the number of links L in the WBN. Since the WBN topology is a tree, the scheduling algorithm requires a total of 2(N-1) steps, which is linear in the number of nodes in the network.

V. Distributed protocol

The MWIS algorithm of the previous section is centralized and executed on the vertices of the WBN CG. We now use this algorithm to design a slotted *distributed* protocol executed by the nodes in the WBN that still solves the MWIS problem in linear time. In the following, we focus on the protocol description for the multi-channel case. The single channel case follows similar steps and is described in detail in [6].

The frame structure of the distributed protocol is reported in Fig. 2. Each data transmission phase can accommodate several packets for each sender and is preceded by a scheduling phase. Each scheduling phase is made up of two sub-phases: the Weight-exchange phase and the Independent-set Construction

[3] In a vertex-integer-weighted graph G, a set C of cliques is called a clique cover if every vertex of G is an element of at least as many elements of C as its weight is.

TABLE I

WEIGHT EXCHANGE PHASE FOR CG OF FIG. 1(B) FOR SOME INITIAL WEIGHTS. CG VERTICES SHOWN AND VISITED IN INCREASING LABEL ORDER.

	Inc-label-sorted CG vertices and their residual weights RW																		TRAV
Iteration (=label #)	a	h	k	q	o	c	i	j	r	p	b	f	n	d	m	e	l	g	
Initial weights	1	2	1	2	3	2	2	3	2	4	4	2	3	5	2	6	4	6	
1	0					1					3			4					a
2		0					0									4			a, h
3			0					2				1							a, h, k
...																			...
18	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	-1	0	0	-1	a, h, k, q, o, c, j, p, b, n, d, e, l

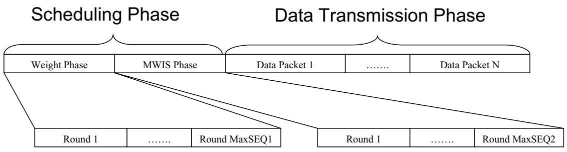


Fig. 2. The distributed protocol frame structure.

phase denoted by Weight-phase and MWIS-phase respectively. The Weight-phase and MWIS-phase consist of several synchronized rounds each. Each link in the WBN NG is mapped to a particular Weight-phase round and to a particular MWIS-phase round.

During each Weight-phase round, the end nodes of the link mapped to this round broadcast the residual weight of this link (if positive) and both these nodes and the nodes within their range update the residual weights of their adjacent links. During each MWIS-phase round, the end nodes of the link mapped to this round broadcast their independent set membership status (if the link has not so far been excluded from the scheduling phase) and the nodes within their range update the independent set membership status of their adjacent links.

The size of each round, the number of rounds per phase, and the information about which round each node should start performing the protocol actions are all constant and predetermined. This information is computed offline based on a variant of the enumeration procedure discussed in Section IV. Thus, the scheduling phase yields constant overhead per frame. We discuss overhead minimization in Section VI and next present the scheduling phase mechanism.

TABLE II

INDEPENDENT-SET CONSTRUCTION PHASE AFTER WEIGHT-EXCHANGE PHASE OF TABLE I. MWIS IS A SUBSET OF THE TRAV SET. VERTICES IN TRAV SET SHOWN AND VISITED IN DECREASING LABEL ORDER.

	MWIS membership: 1: ∈ MWIS, -1: ∉MWIS												
Iter	l	e	d	n	b	p	j	c	o	q	k	h	a
1	1												
2		1	-1										-1
3				1						-1			
...													
6	**1**	**1**	-1	**1**	**1**	**1**	**1**	-1	-1	-1	-1	-1	-1

A. Offline Parallel Enumeration Procedure

The scheduling phase is based on two enumeration sequences SEQ1 and SEQ2 generated offline for the Weight-phase and the MWIS-phase respectively. We call the procedure that generates SEQ1 and SEQ2 the Parallel Enumeration procedure. This procedure assigns labels sequentially to CG vertices based on the order each link is allowed to exchange its weight and MWIS status information. The end nodes of two links that have the same SEQ1 (SEQ2) enumeration label are allowed to transmit a weight-exchange (independent set membership) control packet during the same Weight-phase (MWIS-phase) round. The details of how to enumerate the CG vertices in an optimal way is discussed in Section VI.

Fig. 1(b) shows the (SEQ1,SEQ2) enumeration pair above each link's id shown in the multi-channel CG. The details of the distributed protocol scheduling phase (Weight-Phase and MWIS-Phase) are given next.

B. Online Scheduling Phase

We demonstrate the protocol operation for a multi-channel implementation based on a transmitter-oriented channel assignment scheme. Each node has been assigned a channel for transmissions and channels have been assigned such that there is no secondary interference. When a node is scheduled to receive from a particular transmitter during a frame, it switches to the channel of that transmitter. For this purpose, nodes maintain a local schedule for the weight-phase and for the MWIS-phase that indicates when and which link they should transmit for or receive information from.

1) Weight-Phase: Each node maintains a per-link residual weight table where each residual weight entry is initialized to be equal to the link's queue size at the beginning of the scheduling phase.

At the beginning of each weight-phase round i, both end nodes of every i-enumerated link l broadcast l's residual weight RW on their channels if RW is positive. Then, both endpoint nodes of link l and all nodes within their range decrement the residual weight of their adjacent links by RW. As a result, the end nodes of every link whose CG vertex is adjacent to link l's CG vertex and has a higher SEQ1 label will decrement the residual weight of that link. Moreover, a link for which the residual weight becomes zero or negative is eliminated by its end nodes from the rest of the weight-phase and the MWIS-phase that will follow. Consequently, the end nodes of such a link will not broadcast its weight and that link will not be scheduled to transmit during the data transmission phase.

Fig. 3(a) shows an example of the Weight-phase during the first Weight-phase round based on the enumeration sequence shown in Fig. 1(b). Broadcasting nodes are dotted in black, and their weight transmissions are represented by arrows. The black endpoint nodes of links a and o (both having a SEQ1 label equal to 1) are the broadcasting nodes and both black and white nodes update their residual weight information for their adjacent links b, c, d, m, and p. Note that the root node of the tree and the leaf nodes do not need to broadcast the weight of their links, and hence they are preconfigured not to do so.

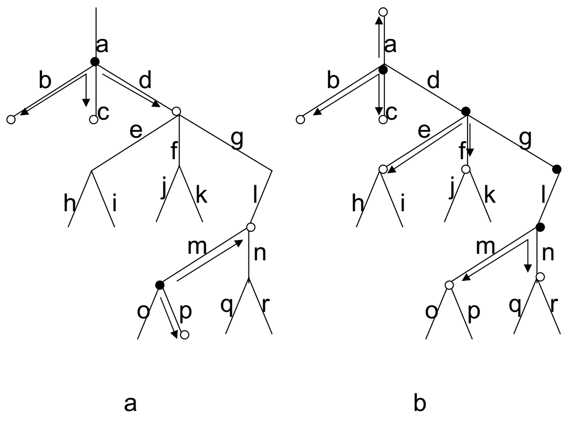


Fig. 3. Weight-phase and MWIS-phase in multi-channel WBN based on the $(SEQ1, SEQ2)$ enumeration sequences shown in Fig. 1(b)

2) MWIS-Phase: This phase consists of sequential local broadcasts of the links' MWIS-membership status in the conflict-free data transmission set. The objective of this phase is to incrementally obtain the conflict-free transmission set using the $TRAV$ set calculated in the Weight-phase.

At the beginning of each round i, both end nodes of every i-enumerated link l broadcast a MWIS-membership packet only if link l has been included in the MWIS transmission set during the weight phase: l broadcasted its positive RW value during the Weight-phase, **and** l is not a neighbor of any link which has been already scheduled for data transmission. By the end of each MWIS-phase round, both end nodes of each broadcasting link l and each node that could sense the MWIS-membership transmission of the end nodes of l, eliminate each link whose CG vertex is adjacent to the CG vertex of link l.

Fig. 3(b) shows an example of the MWIS-phase during the first MWIS-phase round based on the enumeration shown in Fig. 1(b): the end nodes of link g dotted in black are the broadcasting nodes, and both set of nodes dotted in black and white eliminate links d, e, f, and l from the MWIS transmission set.

VI. Overhead Minimization

The overhead of the distributed protocol is dictated by the duration required to complete the MWIS computation for the scheduling phase. This consists of control slot durations, τ_1 and τ_2, and the number of control slots, SEQ1 and SEQ2, in the Weight-phase and MWIS-phase respectively:

$$Overhead = SEQ1 \times \tau_1 + SEQ2 \times \tau_2 \qquad (3)$$

The durations τ_1 and τ_2 depend on the physical layer. We present minimum slot durations for an 802.16 PHY in [6]. The number of control slots SEQ1 and SEQ2 is at most linear in the number of nodes in the network. This bound is topology-independent and holds even in networks with no spatial reuse (e.g., star topology), however it can still yield high overhead as the network size increases.

In this section, we design algorithms that exploit spatial reuse to minimize the overhead given by Eq. (3) by minimizing both SEQ1 and SEQ2 for a given a WBN topology. In contrast to the distributed protocol, these algorithms are centralized and executed at the much slower time scale of topology changes in the WBN. The algorithms determine the minimum constant per-frame overhead as well as the optimal transmission schedule during the control phase of the protocol.

The determination of the optimal schedule cannot be addressed by existing graph coloring approaches because, in addition to spatial constraints, there exist time constraints dictated by the protocol operation. Our solution toward finding the minimum overhead exploits the problem structure and is based on a set of optimal recursive procedures. Guided by our optimal algorithms, we also derive closed form expressions for the case of regular D-ary trees of depth L to gain insights on the amount of overhead as a function of the breadth and depth of the WBN.

A. Problem definition: Enumeration Constraints

The minimum overhead computation problem has both time and spatial enumeration constraints. Time constraints dictate which CG vertices can be enumerated first (hence, have lower labels), whereas spatial constraints ensure that no control messages will collide during the scheduling phase.

SEQ1 Time Constraints: As discussed in the centralized algorithm, any CG vertex is constrained to satisfy the Clique Lemma when it is SEQ1-enumerated.

SEQ1 Spatial Constraints: In order to avoid collisions between weight-exchange packets, two links that are within 2 hops in multi-channel networks and 4 hops in single-channel networks cannot be enumerated by the same SEQ1 label number. Under both access constraints, this is equivalent to the restriction that CG vertices that are within a shortest distance of 3 cannot be enumerated by the same label number. For example, in the network shown in Fig. 1(a) under multi-channel access constraints, link g cannot be enumerated by the label number of any other link in the network.

SEQ2 Time Constraints: Regarding SEQ2 enumeration constraints, CG vertices are supposed to inform their neighbors of their independent transmission set membership status in reverse order with respect to the SEQ1 sequence. Hence, for each pair (v, v') of adjacent CG vertices:

$$\text{if } SEQ1_v < SEQ1_{v'} \Rightarrow SEQ2_v < SEQ2_{v'} \qquad (4)$$

SEQ2 Spatial Constraints: The MWIS phase allows more spatial reuse than the Weight-phase because it only requires nodes to sense MWIS-membership transmissions instead of decoding weight-exchange packets. Sensing the correct MWIS-membership transmissions requires links subject to primary interference in a multi-channel NG (subject to both primary and secondary interference in a single-channel NG) not to be enumerated by the same SEQ2 label number. Under both single-channel and multi-channel access constraints, this is equivalent to the restriction that adjacent CG vertices cannot be enumerated by the same label.

B. Solution: Minimum-Enumeration Algorithms

We now present algorithms to compute the minimum number of Weight-phase rounds (SEQ1) and MWIS-phase rounds (SEQ2) for any multi-channel WBN. The single-channel case is treated in a similar fashion and can be found in [6]. We first introduce the Highest-label Vertex Theorem, crucial in solving the problem and deriving the algorithms' optimality proofs:

Highest-label Vertex Theorem: *In the minimum-overhead solution, both highest SEQ1 and SEQ2 enumeration labels are occupied by a single and the same CG vertex.*

The Highest-label Vertex Theorem provides a starting point toward the solution. Since the highest label for both SEQ1 and SEQ2 will be occupied by a single vertex in the CG, it suffices to consider a problem instance that minimizes SEQ1 and SEQ2 starting from a vertex. The minimum-overhead solution is the minimum solution over all problem instances.

We associate the problem instance of each WBN node R with a *candidate tree* T_R rooted at R. We then design algorithms that operate on each candidate tree T_R to compute the minimum SEQ1$_R$ and minimum SEQ2$_R$, thus minimizing Eq. (3) for the node R problem instance. Given T_R, our algorithms utilize the Highest-label Vertex Theorem to constrain the set of links in T_R assuming the root R has the highest label. Focusing on a WBN node R, we proceed to describe the methodology that solves its corresponding problem instance using three algorithms: the Candidate Tree Construction Algorithm, Min-SEQ1 computation algorithm, Min-SEQ2 computation algorithm. The optimality proofs can be found in [6].

1) Candidate Tree Construction Algorithm: For any non-leaf node R in the network, a candidate tree T_R is constructed as follows:

- The candidate tree is initialized to be the same as the NG except that R is designated to be the root of the candidate tree.
- Every node in the candidate tree is assigned a weight equal to the number of children is has, and every link in the network graph is represented by the corresponding non-leaf parent vertex in the candidate tree.

Fig. 4 illustrates the construction of the candidate tree T_R for the WBN in Fig. 1(a), when the root R is the node adjacent to links l and g.

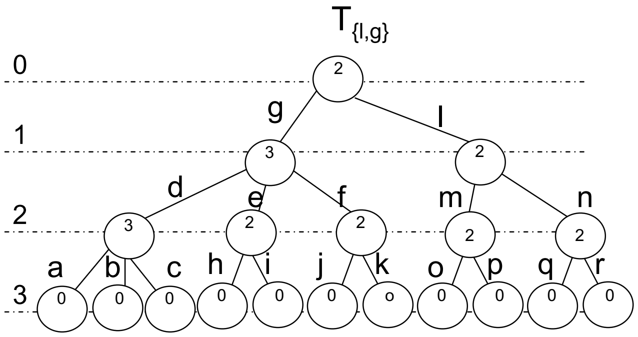


Fig. 4. The $T_{l,g}$ candidate tree construction for the network shown in Fig. 1(a) when the root is the node adjacent to links l and g.

2) Min-SEQ1 computation algorithm: This algorithm finds the minimum SEQ1 size assuming the highest index link is adjacent to R, the root of the candidate tree. The intuition behind the algorithm is to find the minimum SEQ1 size for the smallest subtrees of the candidate tree and then to recursively find the minimum SEQ1 size for each combination of subtrees as they join to build larger subtrees at the higher levels, culminating at the root of the candidate tree. The algorithm proceeds as follows.

Delete the leaf vertices of the candidate tree, denote the lowest level number of the remaining candidate tree T_R by L, and associate with each vertex v in the tree two distinct-element sets: Self$_v$ and Children$_v$. Self$_v$ contains the SEQ1 numbers of all the links adjacent to node v in the NG while Children$_v$ contains the SEQ1 numbers of all the links corresponding to every child vertex v' of v in the remaining candidate tree. The contents of these two sets vary during the minimum-size computation procedure as they are adjusted to satisfy the SEQ1 spatial and time constraints. At the end of the computation procedure, the maximum-valued element in the Self set of R is equal to the desired minimum value SEQ1$_R$. This holds assuming that the root vertex R in the candidate tree represents the link with the highest SEQ1 label, according to the Highest-label Vertex Theorem. We now proceed to describe the steps needed to arrive to the Self set of the root R of T_R.

For each vertex v of the candidate tree, denote its children vertices and weight by Ch_v and W_v respectively. For every vertex v in the remaining subtree, initialize the Self$_v$ set to $\{1, 2, 3 \ldots W_v\}$ and the Children$_v$ set to be empty.

Starting from level $L - 1$ of the tree recursively up to the root level 0, traverse each vertex on the same level of the tree. For each such vertex v, traverse each child vertex v' of v and increase Self$_{v'}$ label values such that the SEQ1 Spatial Constraints are satisfied. More specifically, before increasing the label values of a Self$_{v'}$ set, the SEQ1 enumeration labels for the links represented by the subtree rooted at v' were optimized for maximal spatial reuse during previous iterations; however, this was done assuming the labels in the Self$_{v'}$ are independent from the labels in the $Self$ sets of the other children of v (which is generally not the case according to the SEQ1 Spatial Constraints discussed earlier). Therefore, to

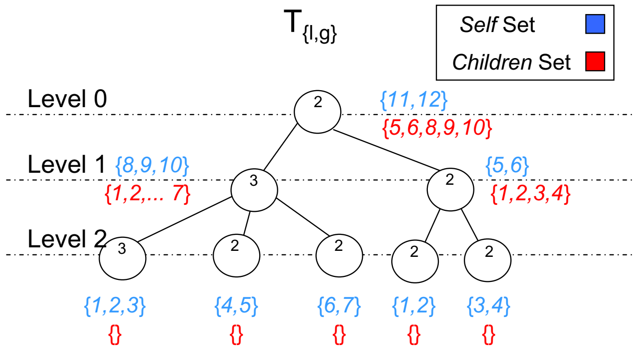


Fig. 5. Final Self and Children Sets of the Candidate Tree $T_{l,g}$ shown in Fig. 4

maintain SEQ1-size optimality, increasing the label values in a Self$_{v'}$ set should be done while preserving as much spatial reuse as possible. To achieve this objective, we increase the value of every element member in Self$_{v'}$ by the minimum value such that the spatial constraint is not violated:

$$\text{Self}_{v'} \cap (\text{Children}_{v'} \cup \text{Self}_{v''}) = \phi$$
$$\forall\, (v', v'') \in Ch_v \times Ch_v \tag{5}$$

The Children set of v is now made equal to the union of all the Self sets of its children. The Self set of v is then made equal to the minimum lexicographic set of W_v elements such that both spatial and time constraints are captured as follows:

$$\text{Self}_v \cap (\text{Children}_v \cup \text{Self}_{v''} \cup \text{Children}_{v''}) = \phi, \tag{6}$$

and

$$\forall\, vertex\, v' \in Ch_v : \exists\, x \in \text{Self}_v \; such\, that$$
$$SEQ_{1x} > SEQ_{1x'}\, \forall x' \in \text{Self}_{v'} \tag{7}$$

Fig. 5 shows the final Self and Children sets of each vertex in the candidate tree of Fig. 4.

3) Min-SEQ2 Computation Algorithm: This algorithm recursively finds the minimum SEQ2 size using similar intuition as that for finding the minimum SEQ1 size: start by finding the minimum SEQ2 size for the smallest subtrees of a candidate tree and then recursively find the minimum SEQ2 size for each combination of subtrees as they join to build larger subtrees at the higher level of the candidate tree. The algorithm is as follows:

Denote the lowest level number of the candidate tree T_R by L and the set of the children of each vertex v by Ch_v. Sort the children of each vertex v by their weight in decreasing order and denote the position of each child v' of v in the sorted list by $order_{v'}$. Starting from level $L-1$ of the tree recursively up to level 0, traverse all the tree vertices at the same level. For every vertex v traversed, update its weight w_v as follows:

$$w_v = max(w_{v'} + order_{v'}) \forall v' \in Ch_v$$

The final weight of the root vertex is the minimum SEQ2 Parallel Enumeration index for R. Fig. 5 shows the final weight of each vertex in the candidate tree of Fig. 4 after execution of the Min-SEQ2 Computation Algorithm.

C. Minimum overhead of D-ary Trees of depth L

Following the steps of the minimum overhead computation algorithms in the previous section, it is possible to derive simple closed form expressions for the special case of D-ary trees of depth L where each non-leaf node has the same number of children D, and all leaf nodes are L hops away from the tree's root node. The minimum size computation formulas for both single-channel and multi-channel WBNs are reported in table III. We observe that, for fixed depth L, the order of overhead increase as a function of the node degree D is quadratic in multi-channel WBNs and order-3 polynomial in single-channel WBNs. On the other hand, the order of overhead increase as a function of depth L for a fixed degree D is linear in both multi-channel and single-channel networks. Hence, trees that grow by depth yield less overhead with our protocol than trees that grow by breadth.

TABLE III
MINIMUM (SEQ1,SEQ2) OVERHEAD FOR D-ARY TREES WITH DEPTH L

	Min SEQ1	Min SEQ2
Multi-channel	$D + LD^2 - D^2 - 1$	$LD - 1$
Single-channel	$D + D^2 + 2D^3(L-2) - 1$	$2LD - D^2 - D - 1$

VII. PERFORMANCE EVALUATION

We compare our protocol with maximal scheduling protocols since they can be implemented in a distributed manner.

For multi-channel systems, Lin and Shroff outline a maximal protocol in [7] based on Hoepman's algorithm for distributed weighted matchings [4]. The time complexity of this algorithm is is $O(N)$ for wired networks and its complexity increases by an order if the primary interference constraints are taken into account [8]. More importantly, complexity is only given in the average sense with no deterministic upper bound on the overhead. To the best of our knowledge, upper bounds are only provided by the work in [7], which specifies a protocol to compute maximal schedules in $2N$ rounds for single-channel systems. We therefore denote the protocol in [2] as *Maximal* and compare it with the single-channel version of our protocol.

Overhead. The *Maximal* protocol is time slotted and uses a transmission architecture similar to ours: a scheduling phase followed by a data transmission phase for each frame. The scheduling phase of *Maximal* consists of a constant number of rounds where nodes are scheduled to broadcast an ID message during one round and a schedule inclusion / exclusion message during another round. More specifically, each node in the network is allocated one round to broadcast an ID message, a second round to potentially broadcast an exclusion message, and a third round to potentially broadcast an inclusion message (if the link has not transmitted an exclusion message yet). Therefore, based on our PHY parameter values for the control slot durations, each ID message (inclusion and exclusion message) should be allocated a round of at least τ_1 (τ_2). Hence, the overhead of *Maximal* is equal to $(\tau_1 + 2\tau_2)N$ where N is the number of nodes in the network.

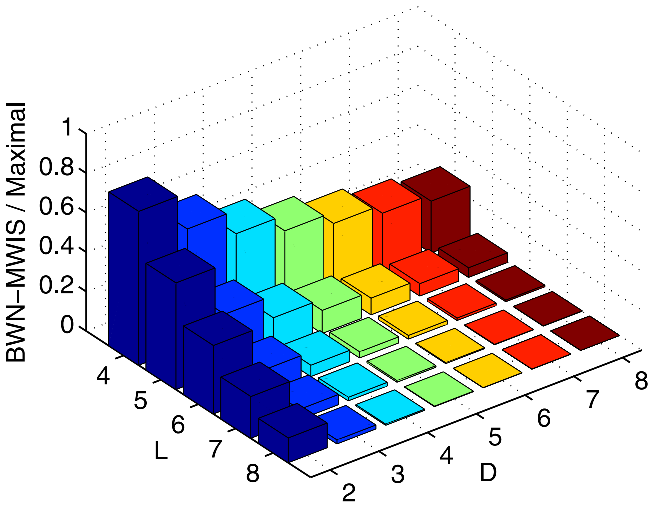


Fig. 6. MWISTS-to-Maximal Overhead Ratio in Single-channel Networks

For D-ary trees with depth L, the overhead of *Maximal* becomes $(\tau_1 + 2\tau_2)(D^{L+1} - 1)/(D - 1)$. Fig. 6 depicts the overhead ratio of our single-channel protocol (denoted by WBN-MWIS) over *Maximal* as a function of D and L parameters. The overhead of our protocol is computed using the SEQ1 and SEQ2 formulas in Table III and both protocols use the minimum slot durations $\tau_1 = 17\mu s$ and $\tau_2 = 9\mu s$ computed in [6] assuming 802.16 PHY specifications. We observe that the overhead of WBN-MWIS can be at most 75% of *Maximal* for (D=2,L=4) and at least 0.02% of *Maximal* for (D=8,L=8). In general the ratio quickly decreases when either the degree or the depth of the tree increases. When D is fixed and L varies, the decrease is exponential of order $O(LD^{-L})$. When L is fixed and D is varied, the decrease is polynomial of order $O(D^{1-L})$. Hence, deep WBNs are worse for *Maximal* protocol than for WBN-MWIS.

Stability. Single-channel maximal scheduling protocols have been analytically shown to guarantee at least 1/8 of the maximum throughput region in arbitrary wireless networks. We show through simulations that their performance limitations hold for the case of WBN topologies. We conducted experiments for the WBN shown in Fig. 1. The capacity of each transmission link is normalized to one packet per slot and all packets are of equal size. Using the technique in [5], we compute the link Max-Min Fair (MMF) rates, and use them as input CBR traffic load to the WBN links. We monitor the average queue sizes every 900,000 time units. The total running time for the experiment was 17,100,000 time units.

For WBN-MWIS, we observed that the average queue size of each link remained constant, which verifies the stability property of the scheduling algorithm on the (feasible) MMF allocation. However, *Maximal* was only found stable by scaling down all input rates to 25% of the MMF rate allocation. This fraction depends on the particular WBN topology and on that the arrival rates were scaled down as a fraction of the MMF rates. However, this example illustrates that a maximal scheduling protocol fails to schedule a significant amount of points which can be scheduled using WBN-MWIS.

VIII. Conclusions

In this paper, we introduced a distributed low-complexity maximum-throughput scheduling protocol for Wireless Backhaul Networks (WBNs). We showed that the MWIS problem for the WBN setting can be solved in linear time using novel graph-theoretical properties of the WBN contention graph. We also provided algorithms to minimize the overhead of the scheduling phase. We have demonstrated that our protocol outperforms existing protocols based on maximal scheduling both in terms of overhead as well as throughput performance.

Acknowledgements

This research was supported by NSF grants ANI-0331620 and ANI-0325971 and by Intel Corporation.

References

[1] G. Zussman A. Brzezinski and E. Modiano. Enabling distributed throughput maximization in wireless mesh networks - a partitioning approach. In *Proc. ACM MobiCom*, Los Angeles, CA, September 2006.

[2] I. Cidon and M. Sidi. Distributed assignment algorithms for multihop packet radio networks. *IEEE Transactions on Computers*, 38(10):1353–1361, October 1989.

[3] V. Gunasekaran and F. Harmantzis. Affordable Infrastructure for Deploying WiMAX Systems. In *Proc. IEEE Vehicular Technology Conference (VTC)*, Dallas, TX, 2005.

[4] J. Hoepman. Simple distributed weighted matchings. In *eprint cs.DC/0410047*, October 2004.

[5] X. Huang and B. Bensaou. On max-min fairness and scheduling in wireless ad-hoc networks: Analytical framework and implementation. In *Proceedings of ACM MobiHoc*, Long Beach, CA, October 2001.

[6] A. Kabbani, T. Salonidis, and E. Knightly. Distributed Low-complexity Maximum Throughput Scheduling in Wireless Backhaul Networks. *Technical Report, Rice University*, 2006.

[7] X. Lin and N.B. Shroff. The impact of imperfect scheduling on cross-layer rate control in wireless networks. In *Proc. IEEE INFOCOM*, Miami, FL, March 2005.

[8] E. Modiano, D. Shah, and G. Zussman. Maximizing throughput in wireless networks via gossiping. In *Proc. ACM SIGMETRICS*, Saint-Malo, France, 2006.

[9] C. Papadimitriou and K. Steiglitz. Combinatorial Optimization, Algorithms and Complexity. 1998.

[10] B. Raman and K. Chebrolu. Design and Evaluation of a new MAC Protocol for Long-Distance 802.11 Mesh Networks. In *Proc. ACM MobiCom*, Cologne, Germany, August 2005.

[11] P. Chaporkar S. Sarkar and K. Kar. Fairness and throughput guarantees with maximal scheduling in multihop wireless networks. In *Proc. WiOpt06*, Boston, MA, April 2006.

[12] T. Salonidis and L. Tassiulas. Distributed Dynamic Scheduling For End-to-end Rate Guarantees In Wireless Ad Hoc Networks. In *Proc. ACM MobiHoc*, Urbana-Champaign, IL, USA, May 2005.

[13] D. Shah and M. Kopikare. Delay bounds for approximate maximum weight matching algorithms for input queued switches. In *INFOCOM*, New York, NY, June 2002.

[14] L. Tassiulas and A. Ephremides. Stability Properties of Constrained Queueing Systems and Scheduling Policies for Maximum Throughput in Multihop Radio Networks. *IEEE Transactions on Automatic Control*, 37(12):1936–1948, December 1992.

[15] G. Wilfong, G. Narlikar, and L. Zhang. Designing multihop wireless backhaul networks with delay guarantees. In *Proc. IEEE INFOCOM*, Barcelona, Spain, 2006.

[16] X. Wu and R. Srikant. Regulated Maximal Matching: A Distributed Scheduling Algorithm for Multi-Hop Wireless Networks With Node-Exclusive Spectrum Sharing. In *Proc. IEEE CDC-ECC05*, Seville, Spain, December 2005.