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

Abdul Kabbani, Theodoros Salonidis, and Edward W. Knightly
Department of Electrical and Computer Engineering
Rice University
Houston, TX 77005

***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 (i) even if global topology information is available, the problem of computing the optimal link transmission set at each slot is NP-complete (ii) no bounds exist on the number of steps required for such a computation (per-slot overhead). For the logical tree structures induced by the 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 the 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 arbitrary wireless networks 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 [17], 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. This result holds for arbitrary topologies and link access contraints. However, it 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 exceed the time it takes the nodes to solve *any* instance of the optimal scheduling problem, thereby requiring a deterministic upper bound on the number of computation 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 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 that mutually interferes with *all* of its interfering neighbors. We utilize this property to devise a two-phase centralized algorithm that computes a MWIS by traversing the network in forward and reverse order. The algorithm visits each link exactly once in each phase, regardless of the MWIS problem instance at each frame. This bounds the size of the scheduling phase to twice the number of nodes in the WBN. 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, the bound on the scheduling phase duration of the protocol can still be large in practice. We provide an algorithm that solves the offline design optimization problem of determining the minimum scheduling phase duration for a given WBN topology.

We compare our protocol to a protocol that computes a *maximal* link transmission set at each frame. For the case of arbitrary wireless networks, maximal scheduling has been shown to guarantee at least half the region of the feasible arrival rates [11], [14], [19] and can be implemented using distributed protocols with a linear bound on the scheduling phase [7]. We show that maximal scheduling remains suboptimal in the case of WBNs, and existing maximal scheduling protocols can yield up to 80% higher overhead depending on the desired degree and depth of the WBN tree structure.

Finally, we evaluate the protocols stability properties under the effect of transmission errors occuring in control and data packets. Of course, the maximum throughput region is essentially reduced due to lost data packets; however, the new

[1] The tree structure is logical (for forwarding and routing) vs. physical (interference). Protocols such as IEEE 802.16j mandate such a structure.

maximum throughput region is not reduced any further as errors occur in the scheduling phase as well. More precisely, we find that stability is retained provided that the packet loss probability of control packets is less than that of data packets, a common design choice in practice. This illustrates the robustness of our scheduling mechanism.

The rest of the paper is organized as follows: In Section II, we review the state of the art in distributed scheduled access protocols for both arbitrary and tree-based wireless networks. In Section III, we describe the WBN interference model and define 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 centalized algorithm in Section V. We solve the overhead minimization problem in Section VI and evaluate the protocol performance in Section IX. Section X concludes.

## II. Related work

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

**Wireless networks of arbitrary topology:** Given the difficulty of MWIS problem, most works have focused on evaluating performance of suboptimal approaches based on maximal scheduling. In [11], [19], it was shown that maximal scheduling can achieve at least 50% of the maximum throughput region, for single-hop flows and multi-hop flows, respectively; while the works in [14] showed that in single-channel systems this guarantee may decrease to 1/8. Existing distributed algorithms to find maximal matchings [9] only specify the required message exchanges, but are not directly applicable to the wireless setting. In addition, they only guarantee performance in terms of average complexity. In a protocol implementation the constants may play a big role in the overhead of the scheduling phase. While all works acknowledge the need for node coordination, they do not specify a protocol to find the maximal schedule. The specification of such a protocol determines the overhead induced by the scheduling phase and plays an important role on overall performance. Recently, Modiano et al. introduced a distributed randomized scheduling algorithm for maximum throughput scheduling in multi-channel networks [12]. This work is also the first to consider the overhead issue. However, the coordination protocol to achieve the required objective has not been specified.

**Tree-structured wireless networks:** The work in [13] designs a distributed scheduled access protocol on top of 802.11 PHY and is demonstrated to achieve better performance than 802.11 DCF. The work in [18] focuses on packet scheduling for delay guarantees using techniques from wireline networks. Both [13], [18] 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 [15], designs an asynchronous distributed maximum throughput protocol for multi-channel systems. The work in [6] also shows that multi-channel systems allow distributed heuristics based on Longest Queue First (LQF) to be throughput optimal. However, such heuristics only provide average complexity. In contrast to [6], [15], our protocol achieves maximum throughput for a broader set of interference constraints of WBNs by directly solving the MWIS problem in linear time.

## 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. Links can only transmit under certain scheduling constraints. 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 [13] or by embedding the WBN tree in a hexagonal structure [8].

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 $\{a,e,k,m,r\}$ 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 $\lambda_l$. 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:

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

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, $\limsup_k E[Q_l(k)] < \infty$ for every link $l$ in the WBN.

In [17], 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 [17]:

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

The result was shown to hold for arbitrary topologies, arbitrary link scheduling constraints and iid arrival processes. It has also been extended to more general classes of arrival processes, and multi-slot frames [16]. Despite its universality, this algorithm assumes that the scheduling procedure is made by a centralized controller. Furthermore, to determine the optimal link transmission set, it requires solving a Maximum Weight Independent Set (MWIS) problem at each slot. In multi-channel systems, this 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 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. The proofs for all analytical results and algorithm pseudocodes appear in the Appendix.

## 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 endpoints 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 corresponds to a multi-channel WBN. An independent set in a CG corresponds to a feasible link transmission set in its NG. Thus, solving the MWIS problem on a CG, where the weight of each vertex is equal to the corresponding link's queue backlog, leads to the maximum throughput guarantees.

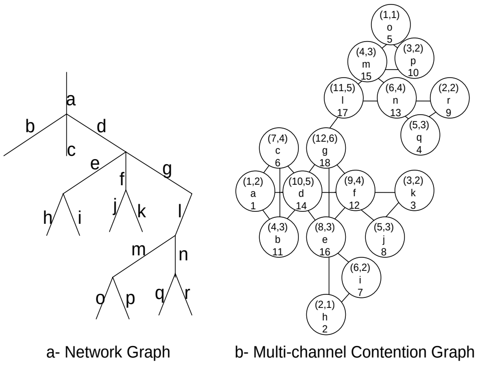

Fig. 1. A Network Graph (NG) and its corresponding Contention Graph (CG) for a multi-channel system.

The graph-theoretical properties of the WBN CG are summarized by the following Lemmas:

**Lemma 1:** *For any vertex* ***a*** *in a WBN CG, if* ***a*** *is adjacent to two other non-adjacent vertices* ***b*** *and* ***c****, then*
*(i)* ***b*** *and* ***c*** *cannot be both adjacent to a vertex* ***d*** *unless vertex* ***a*** *is adjacent to* ***d*** *itself.*
*(ii) Every path from* ***b*** *to* ***c*** *should have an intermediate vertex that is adjacent to* ***a****.*

**Lemma 2:** *Given a WBN CG, it is always possible to find 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 graph possesses the same property.*

Lemma 1 summarizes the connectivity patterns of a WBN CG (see Fig. 2). Lemma 2 reveals a recursive structure in a WBN CG. The properties summarized by the above two Lemmas are not satisfied in general CGs.

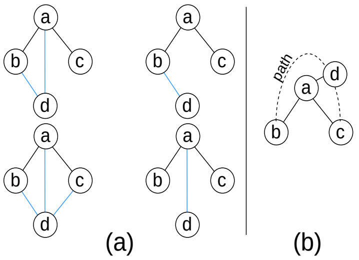

Fig. 2. (a) Lemma1-(i): Possible connectivity patterns for 4 vertices a,b,c,d in a WBN CG (b) Lemma1-(ii). Lemma 1 holds given that (1) a is adjacent to b and c (2) b and c are not adjacent.

### B. Enumeration Algorithm

Based on the Lemma 2 criterion, we first design an enumeration algorithm that assigns a unique sequence label to each vertex in CG. The enumeration sequence determines the order CG vertices will be visited each time the MWIS scheduling algorithm needs to be executed.

**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 Lemma 2), (ii) label $v$ by the number $i$, and (iii) delete $v$ from ICG.

Fig. 1 shows 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 discussed below.

**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 $I$ as long as $I$ does not contain any neighbor of $v$.

Table I illustrates the first few iterations of the Weight Exchange Phase and Table II illustrates the first few iterations of the Independent Set Phase for the CG in Fig. 1.

**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:* 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. Note that every vertex v in an independent set should be covered by at least as many cliques as the weight of v is. Therefore, since each clique can cover one and only one element of the independent set, the cardinality of any clique cover of a vertex-integer-weighted graph is greater than or equal to the weight of any independent set. Making use of this fact, we prove now that the independent set I extracted by the Scheduling Algorithm is a maximum weight independent set by showing that we can obtain a clique cover of the same cardinality as I's weight in a straightforward manner.

By Lemma 2, the set of any vertex v and all v's neighbors that have been enumerated after v form a clique. We denote such a clique by $C_v$ (see Figure 3 for an example). Construct a set C of $RW_x$ cliques $C_x$ for every vertex x in the TRAV set which was built during the Weight-exchange phase - $RW_x$ is the *residual weight* of x right before it was traversed during the Weight-exchange phase. For example, in the case of Table I, C = $\{C_a, C_b, C_b, C_c, C_d, C_e, C_e, C_e, C_h, C_h, C_j, C_j, C_k, C_l, C_l, C_l$ $C_n, C_o, C_o, C_o, C_p, C_q, C_q, \}$. C is a clique cover: every vertex x in TRAV is covered by **exactly** as many cliques as x's weight, and there is no vertex y outside the TRAV set that is not covered by **at least** as many cliques as y's weight (otherwise, y would have been part of the TRAV set). Moreover, no clique $C_v$ in C does not contain any of I's vertices (otherwise, v would have been part of I). Therefore, and since no clique can contain more than one vertex of I, every clique in C covers one and only one vertex of I.

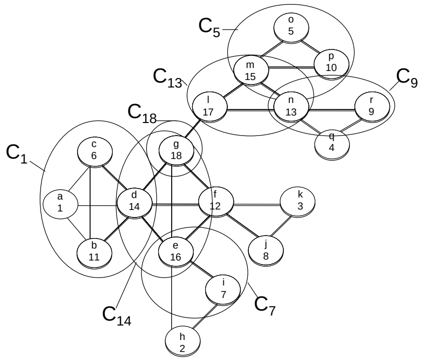

Fig. 3. Some Cliques $C_v$ as obtained after the enumeration algorithm

For the sake of contradiction, assume that the cardinality of C is strictly greater than the total weight of I. According to the Pigeonhole Principle, at least one vertex z in I is covered by more cliques than z's weight which is false.
⇒ the cardinality of C is exactly equal to the summation of the weights of I's vertices
⇒ 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 NG 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 Section V.

The frame structure of the distributed protocol is reported in Fig. 4. Each data transmission phase can accommodate several packets and is preceded by a scheduling phase. Each scheduling phase is made up of two subphases: the Weight-exchange phase and the Independent-set Construction phase, denoted

TABLE I

FIRST THREE ITERATIONS FOR THE WEIGHT-EXCHANGE PHASE OF CG IN FIG. 1 FOR AN ARBITRARY SET OF WEIGHTS. DEPICTED DURING EACH ITERATION ARE THE MODIFIED RW VARIABLES AND THE TRAV SET.

| | Contention Graph Vertex Residual Weights RW | | | | | | | | | | | | | | | | | | TRAV |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Iteration (=label #) | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | |
| 0(initialization) | 1 | 4 | 2 | 5 | 6 | 2 | 6 | 2 | 2 | 3 | 1 | 4 | 2 | 3 | 3 | 4 | 2 | 2 | |
| **1** | 0 | 3 | 1 | 4 | | | | | | | | | | | | | | | a |
| **2** | | | | | 4 | | | 0 | 0 | | | | | | | | | | a, h |
| **3** | | | | | | 1 | | | | 2 | 0 | | | | | | | | a, h, k |

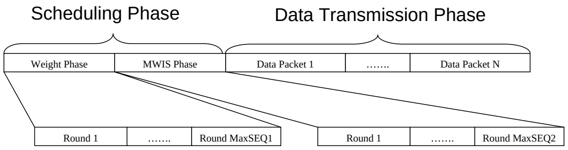

Fig. 4. Scheduling Mechanism

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 round of the scheduling phase, the end nodes of a link mapped to this round broadcast the link's residual weight (if positive) or their independent set membership status (if the link is going to be scheduled). In addition, broadcasting nodes and other nodes in their range update the residual weight values and the independent set membership status of their adjacent links. We describe the protocol steps in detail in the following subsections.

The size of each round, the number of rounds per phase, and the information about which nodes perform what actions and during which rounds are all constant and predetermined. This information is computed offline based on a variant of the enumeration procedure discussed in Section IV. The scheduling phase corresponds, therefore, to constant overhead per each transmission data phase. We discuss overhead minimization in Section VI and next present the scheduling phase mechanism.

TABLE II

FIRST THREE ITERATIONS OF THE INDPEPENDENT SET CONSTRUCTION PHASE FOR THE CG IN FIG. 1. THE MWIS IS FOUND BY VISITING THE CG VERTICES IN TRAV SET IN DECREASING LABEL ORDER.

| | 0: Not Traversed, 1: ∈MWIS, -1: ∉MWIS | | | | | | | | | | | | | I |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| labl | a | b | c | d | e | h | j | k | l | n | o | p | q | |
| | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |
| 17 | | | | | | | | | 1 | -1 | | | | l |
| 16 | | | | -1 | 1 | -1 | | | | | | | | l, e |
| 14 | | | | | | | | | | | | | | l, e |

## 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 denote the procedure that generates SEQ1 and SEQ2 by the Parallel Enumeration procedure. The Parallel Enumeration 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 as well.

Figure 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 RW 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 its 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 all their adjacent links by $RW$. As a result, the end nodes of every link whose CG vertex is adjacent and has a higher SEQ1 label than the CG vertex of link $l$ 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.

Figure 5-a shows an example of the Weight-phase during the first Weight-phase round based on the enumeration sequence shown in Figure 1-b. Broadcasting nodes are dotted in black, and their weight transmissions represented by arrows. The endpoint nodes of links a and o (both having a SEQ1 label equal to 1) are the broadcasting nodes and both sets of nodes dotted in black and white update their residual weight information for the corresponding neighboring 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.

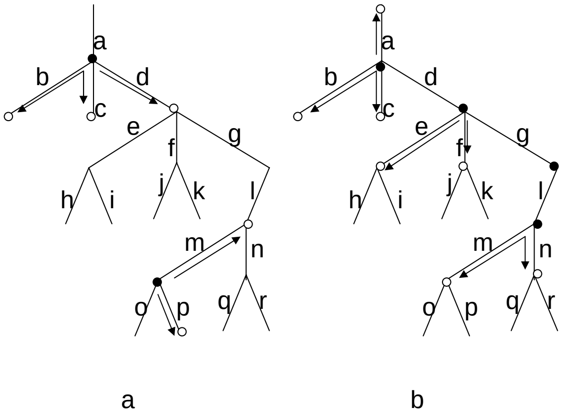

Fig. 5. Weight-phase and MWIS-phase in multi-channel WBN based on the (SEQ1,SEQ2) enumeration Sequences shown in Figure 1-b

*2) MWIS-Phase:* At the beginning of MWIS-phase round $i$, both endpoint nodes of an $i$-enumerated link $l$, broadcast a short **MWIS-membership** packet on their channels if link $l$ has not been eliminated during the previous Weight-phase or during previous rounds of the current MWIS-phase. Then, the endpoint nodes of $l$ as well as their one-hop neighbors which sensed the **MWIS-membership** transmission eliminate all their adjacent links except link $l$. Thus, by the end of round $i$, all links whose CG vertex is adjacent to the CG vertex of link $l$ are effectively eliminated from the MWIS.

Figure 5-b shows an example of the MWIS-phase during the second MWIS-phase round based on the enumeration shown in Figure 1-b: the endpoint nodes of links d and l dotted in black are the broadcasting nodes (assuming that links d and l, of SEQ2 equal to 5, were part of the TRAV set by the end of the Weight-phase, and link g, of SEQ2 equal to 6, was not), and both set of nodes dotted in black and white are supposed to eliminate links a, b, c, e, f, m, and n 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, $\tau_1$ and $\tau_2$, and the control slot counts, SEQ1 and SEQ2, for the Weight-phase and MWIS-phase respectively.

The durations $\tau_1$ and $\tau_2$ depend on the physical layer. We present minimum slot durations for an 802.16 PHY in Section VIII. The number of control slots SEQ1 and SEQ2 is at most linear in the number of nodes in the network. While this can still yield high overhead in practice, this bound is topology-independent and holds even in networks with no spatial reuse (e.g., star topology).

In this section, we design algorithms that exploit spatial reuse to minimize both SEQ1 and SEQ2, 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 scheduling phase of the protocol.

The determination of the optimal schedule cannot be addressed by classical graph coloring approaches because, in addition to interference constraints, there are timing constraints dictated by the protocol operation. Our solution exploits the problem structure and is based on a recursive procedure. Guided by our optimal algorithms, we also derive closed form expressions for the case of regular tree structures to gain insights on the overhead size in terms of the breadth and depth of the WBN.

### A. Enumeration Constraints

We describe the constraints that shape our problem and differentiate it from classical graph theoretical problems. Each enumeration procedure has time and spatial constraints: time constraints essentially dictate which CG vertices can be enumerated first (hence, have lower label numbers), whereas spatial constraints enforce 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 Lemma 2 when it is SEQ1-enumerated.

**SEQ1 Spatial Constraints:** In order to avoid collisions between weight-exchange packets, two links of which any node endpoints are within 2 hops in a multi-channel NG (4-hops in a single-channel NG) cannot be enumerated by the same SEQ1 label number. Under both single-channel and multi-channel access constraints, this is equivalent to the restriction that CG vertices within a shortest distance of 3 cannot be enumerated by the same label number. For example, in the network shown in Figure 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 the reverse order (with respect to the SEQ1 sequence). Hence, for every pair of adjacent CG vertices $(V, V')$,

$$\text{if } SEQ1_V < SEQ1_{V'} \Rightarrow SEQ2_V < SEQ2_{V'} \quad (3)$$

**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-ecxhange 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 SEQ2 label number.

### B. Minimum-Enumeration Algorithms

Algorithms to compute the minimum number of Weight-phase rounds (SEQ1 size) and MWIS-phase rounds (SEQ2 size) needed for any multi-channel WBN are presented below. The single-channel case is similar and can be found in Section VII. We first introduce the Maximum-enumeration index theorem, crucial in solving the problem and deriving the proofs of the optimal algorithms:

**Highest-Index Vertex Theorem:** *If a particular CG vertex $v$ is constrained to have the highest SEQ1 enumeration label, then $v$ and only $v$ will have the highest SEQ1 and SEQ2 enumeration labels.*

The Highest-Index 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. Then the solution is the minimum of all problem instances.

We represent each problem instance by a pivot tree derived from the network graph and rooted at each node $R$. We then design algorithms that operate on the pivot tree to compute the minimum possible SEQ1 and SEQ2 Parallel Enumeration sizes. For a particular pivot tree $T_R$, our algorithms make use of the Maximum-enumeration index theorem and constrain the set of links in the network corresponding to the root vertex $R$ to contain a link corresponding to such a vertex $v$ with the highest index. The algorithms find the minimum SEQ1 and SEQ2 enumeration labels such a vertex $v$ can obtain. Denote these minimum SEQ1 and SEQ2 labels by SEQ1$_R$ and SEQ2$_R$ respectively. An optimal SEQ1 and SEQ2 size pair is essentially the pair that minimizes the following overhead equation for every vertex $R$ being the root of a pivot tree:

$$SEQ1_R \times \tau_1 + SEQ2_R \times \tau_2 \tag{4}$$

Focusing on a WBN node R, we proceed to describe the methodology that solves its corresponding problem instance using three algorithms: the Pivot Tree Construction Algorithm, Min-SEQ1 computation algorithm, Min-SEQ2 computation algorithm.

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

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

Figure 6 illustrates the construction of the pivot trees $T_R$ for the WBN in Figure 1.

*2) Min-SEQ1 computation algorithm:* Delete the leaf vertices of the pivot tree, denote the lowest level number of the remaining pivot 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 Parallel Enumeration SEQ1 numbers of all the network graph links represented by $V$ while $Children_V$ contains the Parallel Enumeration SEQ1 numbers of all the links corresponding to every child vertex $V'$ of $V$ in the remaining pivot tree. The contents of these two sets can vary during the minimum-size computation procedure. However, at the end of the computation procedure, the maximum-valued element in the $Self$ set of $R$ is equal to SEQ1$_R$. This is given the restriction that the root vertex $R$ in the pivot tree should represent the link with the highest SEQ1 label. The details of how to arrive to the $Self$ set of the root vertex of $T_R$ are shown next.

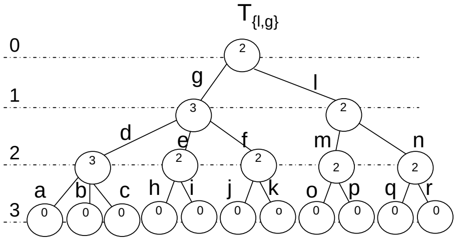

Fig. 6. The $T_{l,g}$ pivot tree construction for the network shown in Figure 1

For a pivot tree vertex $V$, denote $V$'s children vertices and $V$'s 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 level 0 (the root of the pivot tree), traverse every tree vertex on the same level of the tree. For every vertex $V$ traversed, traverse each child vertex $V'$ of $V$ and increase $Self_{V'}$ label values as necessary. 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; however, this was done assuming that the enumeration labels in the $Self_{V'}$ are independent from the $Self$ sets' labels of $V$'s other children (which is generally not the case according to the SEQ1 Spatial Constraints discussed earlier). Therefore, for SEQ1-size's 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:

$$Self_{V'} \cap (Children_{V'} \cup Self_{V''}) = \phi$$
$$\forall\, (V', V'') \in Ch_V \times Ch_V \tag{5}$$

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 lexographic set of $W_V$ elements such that both spatial and time constraints are captured as follows:

$$Self_V \cap (Children_V \cup Self_{V''} \cup Children_{V''}) = \phi, \tag{6}$$

and

$$\forall\, vertex\, V' \in Ch_V \, : \exists\, x \in Self_V \, such\, that$$
$$x.SEQ1 > x'.SEQ1 \, \forall x' \in Self_{V'} \tag{7}$$

Figure 7 shows the final $Self$ and $Children$ sets of each vertex in the pivot tree shown in Figure 6.

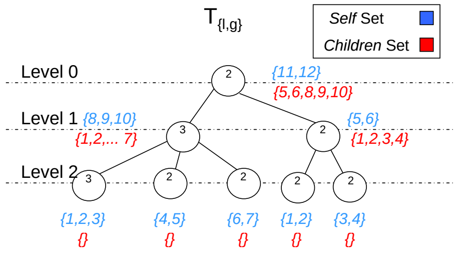

Fig. 7. Final $Self$ and $Children$ Sets of the Pivot Tree $T_{l,g}$ shown in Figure 6

*3) Min-SEQ2 Computation Algorithm:* Denote the lowest level number of the pivot tree $T_R$ by $L$ and the set of the children of a vertex $V$ by $Ch_V$. Sort $V$'s children by their weight in decreasing order and denote the position of $V$'s child $V'$ 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 on the same level. For every vertex $V$ traversed, update the weight of $V$ $w_V$ 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 $r$. The pseudocode for minimum SEQ2 size computations is shown in the appendix, and Figure 7 shows the final weight of each vertex in the pivot tree shown in Figure 6.

### C. Minimum-enumeration Size-computation for Regular Trees

Simple expressions can be derived from the general size-computation algorithms for regular trees where each non-leaf node has the same number of children, and all leaf nodes are equidistant from the tree's root node. Denote the uniform neighborhood degree of the tree by $D$, where $D$ is greater than or equal to 2, and the distance between the leaf node and the root node by $L$, where $L$ is greater than or equal to 2. The minimum size computation formulas are reported in table III. As the table shows, the order of overhead-increase as a function of the neighborhood degree for a fixed-depth tree is quadratic (polynomial of order 3) in multi-channel networks (single-channel networks). The order of overhead-increase as a function of the depth level for a fixed-neighborhood-degree tree is linear in multi-channel and single-channel networks. Hence, trees that grow by depth tend to have less overhead with our protocol than trees that go by breadth.

TABLE III
MINIMUM (SEQ1,SEQ2) OVERHEAD FOR REGULAR TREES

| | Min SEQ1 (rounds) | Min SEQ2 (rounds) |
|---|---|---|
| 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. SINGLE CHANNEL WBNS

The centralized MWIS algorithm is identical for both single-channel and multi-channel systems because it is executed on the vertices of a CG. However, the distributed protocol operates on the NG which is characterized by different interference constraints in each system. In a single-channel NG, one-hop vertex neighbors of a vertex $v$ in CG, correspond to links both one-hop and two-hop away from its corresponding link $l$ (Fig. 8), as opposed to links only one-hop away in a multi-channel NG (Fig.1-b). This implies further propagation of control messages for the distributed protocol and different time and spatial (SEQ1,SEQ2) constraints for overhead minimization. We proceed to describe the modifications and extensions for the multi-channel distributed protocol and overhead minimization algorithms for single channel systems.

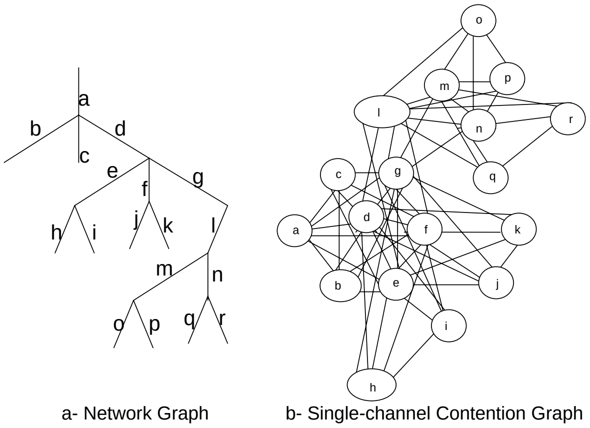

Fig. 8. A Network Graph (NG) and its corresponding Contention Graph (CG) for a single-channel system.

### A. Distributed Protocol

The distributed single-channel protocol uses the same frame structure as the multi-channel protocol (Fig. 4), with two main differences. First, all transmissions occur over a single channel in both scheduling and data phase. Hence, the node local schedules in the scheduling phase only determine whether nodes are in transmit or receive mode. Second, each iteration $i$ for link $l$ of the centralized algorithm in both Weight-Phase and MWIS-phase is mapped to two rounds, $i$ and $i+1$ for link $l$: During round $i$, the link $l$ node endpoints broadcast a control variable (RW or MWIS membership, depending on the phase); during round $i+1$, the one-hop neighbors broadcast the same control variable. Finally link $l$ node endpoints and all nodes within their two-hop neighborhood make the corresponding adjustments on their adjacent links. The specific steps taken during rounds $i$ and $i+1$ at each phase are described below:

*1) Weight-Phase:* At the beginning of each weight-phase round $i$, both end nodes of every $i$-enumerated link $l$ broadcast its residual weight $RW$ if $RW$ is positive. Once they receive it, the one-hop neighbors act as forwarding nodes and rebroadcast the RW value during round $i+1$ to their own one-hop neighbors. At the end of round $i+1$ the link $l$ endpoint nodes and their one-hop neighbors decrement the residual weights of all their adjacent links by $RW$. The two-hop neighbors only decrement the residual weights of their adjacent links from which they received the RW message during round $i+1$ [2]. As a result, the end nodes of every link whose CG vertex is adjacent and has a higher SEQ1 label than the CG vertex of link $l$ 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.

*2) MWIS-Phase:* At the beginning of MWIS-phase round $i$, both endpoint nodes of an $i$-enumerated link $l$, broadcast a short **MWIS-membership** packet on their channels if link $l$ has not been eliminated during the previous Weight-phase or during previous rounds of the current MWIS-phase. Then, during round $i+1$, the one-hop neighbors act as forwarding nodes and rebroadcast the MWIS-membership packet. At the end of round $i+1$ the link $l$ endpoint nodes and their one-hop neighbors eliminate all their adjacent links (except link $l$). The two-hop neighbors only eliminate their adjacent links which correspond to one-hop neighbors that transmitted the MWIS-membership packet during round $i+1$.

### *B. Overhead Minimization*

As briefly discussed in Section VI, the main difference between single-channel and multi-channel systems for the overhead minimization problem lies in the more restrictive spatial enumeration constraints for the single channel case. The overhead minimization algorithms for the single channel case follow the same methodology as in the multi-channel systems. The Pivot Tree Construction Algorithm is common to both systems because it is independent of interference constraints as it is constructed based on the logical tree of the WBN. We proceed to show the modifications required for the Min-SEQ1 Computation Algorithm and Min-SEQ2 Computation Algorithm for the single channel case. The proofs of the algorithms' optimality can be found in the Appendix.

*1) Min-SEQ1 Computation Algorithm:* The same terminology is used and nodes are traversed in the same way as in the multi-channel case except that the spatial constraints are reflected as follows when updating the $Self$ set of the traversed pivot tree vertex $V$ and of every children vertex $V'$:

$$Children_{V'} \cap Children_{V''} = \phi$$

and

$Self_{V'} \cap (Children_{V'} \cup Children_{V''} \cup Children_{V'''} \cup Self_{V''}) = \phi$

and

$Children_{V'} \cap Self_{V'} = \phi$

and

$Self_{V} \cap (Children_{V} \cup Children_{V'} \cup Children_{V''} \cup Children_{V'''}) = \phi$

$\forall$ combinations of vertices $V'$ and $V'' \in Ch_V$ and $V''' \in Ch_{V''}$

The pseudocode for the minimum SEQ1 size calculations is shown in the appendix.

*2) Min-SEQ2 Computation Algorithm:* The same terminology is used and nodes are traversed in the same way as in the multi-channel case except that the weight of each traversed pivot tree vertex $V$ is updated as follows:
$w_V = w_V + max(w_{V'}) \forall V' \in Ch_V$.
The pseudocode for the minimum SEQ2 size calculations is shown in the appendix.

## VIII. Implementation Considerations

Being frame-based, our MAC protocol is best suited for a scheduling structure similar to 802.16 MAC as opposed to contention-based access of 802.11 MAC. However, in addition to 802.16 physical layer (PHY) [1] our MAC protocol can also operate over 802.11a PHY [5]. 802.11a PHY is designed to provide speeds up to 54Mbps in the 5 GHz unlicensed band. It also provides robustness and multi-path immunity for time-slotted systems (small guard times of 0.8$\mu$s being sufficient) and has been strongly suggested to be integrated with 802.16 MAC into a wireless MAN system that employs TDM [3].

We therefore analyze the feasibility of our protocol based on both 802.16 PHY and 802.11a PHY. More specifically, we estimate the duration of each round in the Weight-phase and the MWI-phase. Combined with the number of rounds given by the overhead minimization algorithms of Section VI, this fully specifies the overhead required by our protocol for existing wireless technologies.

An 802.16 (802.11a) PHY frame consists of a 8 (16) $\mu$s preamble, followed by a 4-$\mu$s-SIGNAL field that carries the length of the packet and the transmission rate (6, 9, 12, 18, 24, 36, 48 or 54 Mbps). The first 4 (8) $\mu$s in the preamble are used to acquire the automatic gain control (AGC), coarse frequency offset recovery, and coarse symbol synchronization. The next 4 (8) $\mu$s are used for fine offset recovery, symbol synchronization, and channel equalization. In 802.16 as well as in 802.11a, the first 4 $\mu$s in the preamble are enough to acquire the AGC, which essentially means that a transmission has been detected. More specifically, such a 4-$\mu$s transmission would be long enough to be sensed during one of the MWI-phase rounds and would signify a positive MWI-membership status. In addition, the SIGNAL field can be replaced by the actual weight value in the case of control packets sent during the Weight-phase. A weight-control packet would be assumed therefore to be always sent at the base rate. In addition to the transmission time and sensing time requirements, the length of each round in the Weight-phase and the MWI-phase should take into account the Transmit-to-Receive switching time. 802.16 and 802.11a PHY specify this time, together with a reasonably big propagation delay period and a single guard slot, not to exceed 5 $\mu$s. Therefore, it is sufficient to allocate 17 (25) $\mu$s for each weight-phase round and 9 $\mu$s for each MWI-phase round in the case of an 802.16 (802.11a) PHY.

Given that we have investigated both, the number of control rounds needed to accomplish the MWI schedule and the size of each round slot, we can now evaluate the exact amount of overhead introduced by our protocol in a number of particular topologies.

We assume data frame sizes to be between 4 and 20 msec as this has been suggested for the 802.16 protocol in the TDM mode [4], [2]. Tables IV and V show numerical overhead

[2] The knowledge about which adjacent links to decrement upon hearing a message on a specific round is preconfigured as part of the offline Parallel Enumeration procedure. This is because for efficiency purposes, both weight-exchange and MWIS-membership packets do not contain the address of their sender.

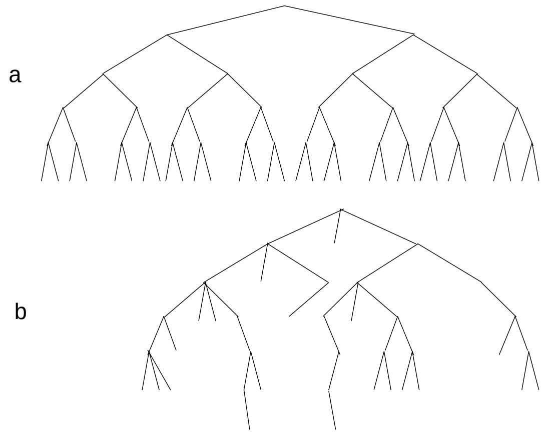

Fig. 9. Two Example Topologies for Overhead Illustration as Shown in Tables IV and V

TABLE IV
OVERHEAD UNDER MULTI-CHANNEL ACCESS CONSTRAINTS

| | Overhead with Different Slot Sizes | | | | |
|---|---|---|---|---|---|
| Topology | 4msec | 8msec | 12msec | 16msec | 20msec |
| Figure 1 | 7.41% | 2.85% | 2.60% | 1.96% | 1.57% |
| Figure 9-a | 11.23% | 5.95% | 4.06% | 3.07% | 2.47% |
| Figure 9-b | 8.32% | 4.34% | 2.94% | 2.22% | 1.78% |

values for the topologies show in Figures 1 and 9 under multi-channel and single-channel access constraints respectively while varying the slot size. The values calculated are based on the 802.16 PHY standard, and we assume the minimum sufficiently large weight-phase and MWI-phase round slot sizes of 17 $\mu$s and 9 $\mu$s respectively.

## IX. PERFORMANCE EVALUATION

### A. Comparison with Maximal Scheduling

We compare our protocol with maximal scheduling protocols since they can be implemented in a distributed manner. The work in [11] specifies a protocol to compute maximal schedules for single-channel systems in $2N$ rounds. For multi-channel systems, Lin and Shroff outline a maximal protocol in [7] based on Hoepman's algorithm for distributed weighted matchings [9]. This algorithm does not have a deterministic upper bound as [7]. Its time complexity is $O(N)$ for wired networks and increases to $O(N^2)$ if the primary interference constraints are taken into account [12]. We therefore denote the protocol in [7] as *Maximal* and compare it with the single-channel version of our protocol.

TABLE V
OVERHEAD UNDER SINGLE-CHANNEL ACCESS CONSTRAINTS

| | Overhead with Different Slot Sizes | | | | |
|---|---|---|---|---|---|
| Topology | 4msec | 8msec | 12msec | 16msec | 20msec |
| Figure 1 | 17.18% | 9.40% | 6.47% | 4.93% | 3.98% |
| Figure 9-a | 26.74% | 15.43% | 10.85% | 8.36% | 6.21% |
| Figure 9-b | 20.05% | 11.14% | 7.71% | 5.90% | 4.78% |

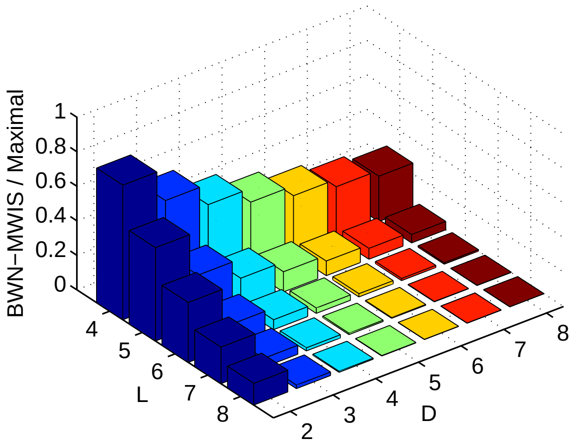

Fig. 10. MWI Maximal Overhead Ratio in Single-channel Networks

**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, another 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 $\tau_1$ ($\tau_2$). Hence, the overhead of *Maximal* is equal to $(\tau_1 + 2\tau_2)N$ where $N$ is the number of nodes in the network.

For regular trees, the overhead of *Maximal* becomes $(\tau_1 + 2\tau_2)(D^{L+1} - 1)/(D - 1)$. Figure 10 shows the overhead ratio of our single channel protocol over *Maximal* for various regular trees. The overhead is computed using the SEQ1 and SEQ2 formulas in Table III and the same slot durations $\tau_1$ and $\tau_2$ we compute in Section VIII assuming 802.16 PHY specifications. The trend is that the ratio is quickly decreasing when either the degree or the depth of the tree increases. When $d$ is fixed and $L$ is varied, the decrease in the overhead ratio is exponential of order $O(L(D^{-L}))$. When $L$ is fixed and $d$ is varied, the decrease in the ratio is polynomial of order $O(D^{1-L})$. Hence, deep trees are worse for *Maximal*s protocol than for our protocol.

**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 Figure 1. The capacity of each transmission link is normalized to one packet per slot and all packets are of fequal size. Using the technique in [10], 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 our protocol, 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 our maximum throughput protocol.

### B. Robustness

Here we evaluate the effect of transmission errors on the stability of our protocol. As discussed in Section V, weight-phase and MWI-phase control messages are transmitted in the form of a broadcast. A broadcast link transmission fails whenever the weight or the MWI information cannot be received correctly over one of the links where the control message broadcast is taking place. In our simulations, links fail independently and failures can occur with equal probability for data packets and scheduling packets.

When a node fails to receive a broadcasted weight information for a link $L$ during its weight-phase round, it assumes the most recent value received for this link and updates the residual weight information accordingly. For a MWI-phase control message transmission, control packets are only transmitted if a link is broadcasting the information it has scheduled for the following data-phase. Therefore, broadcast link transmission failures during the MWI-phase indicate that neighboring nodes falsely assume the opposite scenario. Such inconsistency potentially leads to collisions during the data phase.

Figure 11 shows the stability region with respect to different arrival rates as a fraction of the Max-Min fair allocation vector: the average queue sizes were monitored while varying the probability of having a link transmission error (x-axis) and the arrival rates for each probability value (y-axis) were varied as a fraction of the Max-Min fair arrival rates until stability was achieved. Interestingly, that same stability region was found to coincide with the stability region obtained when assuming that an error can occur during the data phase only. This was verified with respect to other rate allocation vectors on the boundary of the maximum throughput region as well.

The significance of this result lies in illustrating that the maximum throughput region guaranteed by our protocol is not compromised due to our protocol's mechanism itself (although errors during the scheduling phase can lead to faulty MWIS computations or data collisions). This robustness result was further verified in a hypothetical set of simulations in which errors could occur during the scheduling phase but not during the data phase. In that case, stability was achieved at all error rates except the case of 100% failure. Such a stability result can be verified by noting that having non-zero error rates on control packets is equivalent to statistically delaying the scheduling computations while still performing them periodically on average. This has the same stability effect as increasing the data transmission size, which has been analytically shown to maintain stability [16].

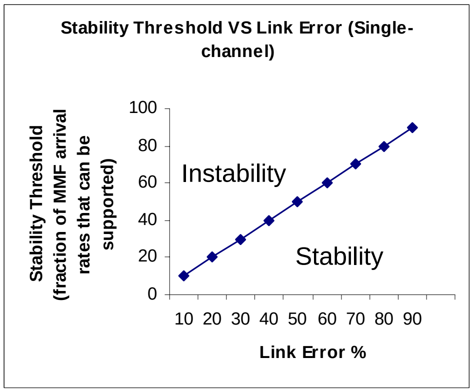

Fig. 11. Stability Region vs Link Error Rate

## X. Conclusions

In this paper, we introduced a distributed low-complexity maximum-throughput scheduling protocol for Wireless Backhaul Networks. 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 and is robust to errors in the control and data channel.

## References

[1] IEEE 802.16 - 2004 - IEEE Standard for Local and metropolitan area networks, Part 16: Air Interface for Fixed Broadband Wireless Access Systems.

[2] IEEE 802.16-02/33r2 - 2002 - Supplement to STANDARD for Information Technology Telecommunications and information exchange between systems Local and metropolitan area networks Specific Requirements IEEE 802.16 Working Group Confirmation Ballot #7a Announcement.

[3] IEEE 802.16.4c- 2001 - Supplement to STANDARD for Information Technology Telecommunications and information exchange between systems Local and metropolitan area networks Specific Requirements Proposed modifications of 802.16 MAC and 802.11a PHY for a WirelessHUMAN Standard employing TDD/TDM.

[4] IEEE C802.16a-02/46 - 2002 - Supplement to STANDARD for Information Technology Telecommunications and information exchange between systems Local and metropolitan area networks Specific Requirements MAC Frame Sizes versus System Performance.

[5] IEEE P802.11a/D7.0 - 1999 - Supplement to IEEE Std. 802.11-1999 DRAFT, Supplement to STANDARD for 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: High Speed Physical Layer in the 5GHz Band.

[6] 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.

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

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

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

[10] 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.

[11] 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.

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

[13] 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.

[14] 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.

[15] 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.

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

[17] 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.

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

[19] 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.

## APPENDIX A

**Algorithm 1** Pseudo-code of the enumeration procedure

```
$i = 1$;
$ICG = CG$;
while $(ICG \neq \phi)$ do
    $enumerated\_vertex =$ any vertex in ICG that satisfies lemma 2;
    $enumerated\_vertex.sequence = i$;
    $ICG = ICG - \{enumerated\_vertex\}$;
    $i++$;
end while
```

**Algorithm 2** Pseudo-code of the procedure that builds the TRAV set

```
$i = 1$;
$v.RW = v.weight \; \forall \; v \; \in \; CG$;
$TRAV = \phi$;
while ($v.RW > 0 \; \forall \; v$ in CG) do
    if ($v$.sequence == $i$ && $v.RW > 0$) then
        $traversed\_node = v$;
        $v.RW = v.RW - traversed\_vertex.RW$; $\forall v$ neighbor of $traversed\_vertex$
        $TRAV = TRAV \bigcup \{traversed\_vertex\}$;
    end if
    $i++$;
end while
```

**Algorithm 3** Pseudo-code of the procedure that finds the MWIS set

```
$i = numberoflinks$;
$I = \phi$;
$ITRAV = TRAV$;
while $(ITRAV \neq \phi)$ do
    if (($v$.sequence)== i && $v \in$ ITRAV) then
        $traversed\_vertex = v$;
        $I = I \bigcup \{traversed\_node\}$;
        $ITRAV = ITRAV - \{\; v : \forall \; v$ adjacent to $traversed\_vertex$ in $TRAV \;\}$;
    end if
    $i--$;
end while
```

**Algorithm 4** Pseudo-code of the Procedure for the Minimum SEQ1 Size Computations for Multi-channel Networks

```
$Self_V = \{1, 2, \ldots |C|\} \forall$ leaf vertex $V$
$CurrentLevel = L - 1$;
while ($CurrentLevel \geq 0$) do
    for (vertex $V$ on level# CurrentLevel) do
        $Adjust\_V's\_Children\_Self\_Set$
        $Children_V = \bigcup Self_{V'} \forall V' \in Ch_V$
        $Adjust\_V's\_Self\_Set$
    end for
    $CurrentLevel--$;
end while
$SEQ1_R = max(i) \forall i \in Self_{V_{T_R},R}$;
```

**Algorithm 5** Pseudo-code of the Procedure for the Minimum SEQ2 Size Computations for Multi-channel Networks

```
$CurrentLevel = L - 1$;
while ($CurrentLevel \geq 0$) do
    for (vertex $V$ on level# CurrentLevel) do
        $w_V = max(w_{V'} + order_{V'}) \forall \, V' \, \in \, Ch_V$
    end for
    $CurrentLevel--$;
end while
$SEQ2_R = w_V$;
```

## APPENDIX B

### A. Proof of Lemma 1

To prove Lemma 1, we follow the methodology below.

**1-** Enumerate all (a,b) combinations such that a is adjacent to b.

**2-** Given each (a,b) combination above, enumerate all (c,d) combinations such that:

- **i-** c is adjacent to a but not to b
- **ii-** d is adjacent to b but not to a

Without loss of generality, we show in Figure 12 all the combinations that result from the above methodology (shown on the network graph level for simplicity).

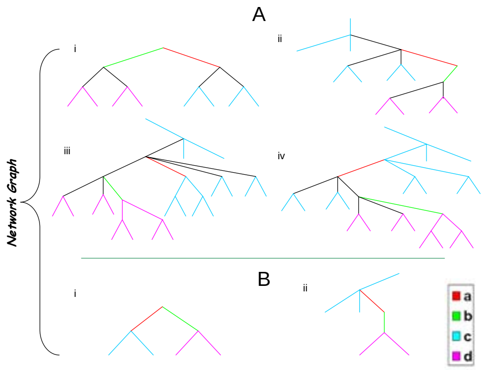

Fig. 12. All (a,b,c,d) combinations under single-channel access constraints (A) and multi-channel access constraints (B) such that a is adjacent to b, c is adjacent to a but not to b, and d is adjacent to b but not to a

As we can notice in Figure 12 a node $c$ can never be connected to a node $d$. In addition, deleting the two nodes constituting link $a$ in the network graph will make node $c$ unreachable from node $b$, hence it will break all the connecting paths between $b$ and $c$ in the contention graph. Essentially, these paths are either traversing $a$ itself or are having an element vertex other than $b$ and $c$ that is connected to $a$.

### *B. Proof of Lemma 2*

If CG is a clique, then any vertex in CG satisfies the lemma. Otherwise, let $x$ and $y$ be two non-adjacent vertices in CG and let $S_{CG}$ be a minimum-cardinality-set of vertices such that every path between $x$ and $y$ has at least one element in $S_{CG}$. Let $D$ be a maximum-cardinality-set of disjoint paths between $x$ and $y$ (see part a - Figure 13). On one hand, it is impossible that $|D| > |S_{CG}|$ for any $D$ as each element of $S_{CG}$ can cover at most one path in $D$. On the other hand, if $|D| < |S_{CG}|$ for some $D$, then a set $S'_{CG}$ of vertices with $|S'_{CG}| = |D|$ can be trivially constructed where every path between x and y has an element in $S'_{CG}$. However, it is impossible that there exists such a set $S'_{CG}$ since $S_{CG}$ is a minimum-cardinality-set of vertices with every path between $x$ and $y$ having at least one element in $S_{CG}$. Therefore, $D$ has exactly $|S_{CG}|$ disjoint paths between $x$ and $y$. In addition, for any set $D$, every path in $D$ should contain one element vertex from $S_{CG}$ by the definition of $S_{CG}$, and no vertex element in $S$ can belong to two paths in $D$ by the definition of $D$. Therefore, for any set $D$, every vertex element in $S_{CG}$ is an element of one and only one path element in $D$, and there is no other vertex element in $S$ that is an element of the same path.

Since every vertex element in $S_{CG}$ is an element of one and only one path element in $D$, for every vertex element $v$ in $S$, there exists a path $P_{xv}$ and a path $P_{yv}$ that connects $v$ to $x$ and $y$ respectively, and $P_{xv} \bigcap P_{xv'} = \phi$, $P_{yv} \bigcap P_{yv'} = \phi$ for every $v' \neq v$ in $S_{CG}$.

For every pair of vertex elements $(v, v')$ in $S_{CG}$, delete all the other vertices in $S_{CG}$ and denote the node subgraph by CG'. The path $P_{xv}P_{xv'}$ connects $v$ to $v'$ through $x$, the path $P_{yv}P_{yv'}$ connects $v$ to $v'$ through $y$, and $P_{xv}P_{xv'}$ and $P_{yv}P_{yv'}$ are disjoint paths. Note that, by the definition of $S_{CG}$, deleting all the vertices in $S_{CG}$ creates at least two disjoint components in CG: one component $G_{x,CG}$ that has the vertex $x$, and another component $G_{y,CG}$ that has the vertex $y$ (see part b - Figure 13). Therefore, we can further conclude that in CG' none of the vertices on the path $P_{xv}P_{xv'}$ is connected to any of the vertices on the path $P_{yv}P_{yv'}$.

We prove the rest of the lemma by contradiction: assume that $v$ and $v'$ are not adjacent. Among the vertices on the path $P_{xv}P_{xv'}$, consider the subset $SP_x$ of vertices that constitute the shortest path between $v$ and $v'$. Because of the assumption, the shortest path is made up of one or more intermediate nodes. Denote the first two consecutive nodes after $v$ on $SP_x$ by $u$ and $u'$ respectively (see part c - Figure 13). $u'$ can be either $v'$ or just another intermediate node on the shortest path. In either cases, $u'$ is not connected to $v$ (otherwise, we are either contradicting the assumption or the shortest path definition). Therefore, according to Lemma 1, since $u$ is directly connected to $v$ and $u'$ while $v$ and $u'$ are not directly connected, then every path connecting $v$ to $u'$ should have at least one element that is connected to $u$. This is not the case as the path $P_{yv}P_{yv'}P_{v'u'}$ does not have any element connected to $u$ ($P_{yv}P_{yv'}$ is part of the $G_{y,CG}$ component while $P_{v'u'}$ is part of the shortest path). This contradicts Lemma 1 and shows that our assumption that $v$ and $v'$ are not connected is wrong.

$\Rightarrow$ any arbitrary pair of vertices $v$ and $v'$ in $S_{CG}$ is adjacent; hence, $S_{CG}$ constitutes a clique.

If $S_{CG} \bigcup G_{x,CG}$ constitutes a clique, then $x$ satisfies Lemma 2, otherwise, we can similarly find another pair of non-adjacent vertices $x'$ and $y'$ in $S_{CG} \bigcup G_{x,CG}$ ($x'$ is chosen such that none of its neighbors in CG has been deleted yet). We also find a minimum-cardinality-set $S_{S_{CG} \bigcup G_{x,CG}}$ of vertices such that every path between $x'$ and $y'$ has at least one element in $S_{S_{CG} \bigcup G_{x,CG}}$. $S_{S_{CG} \bigcup G_{x,CG}}$ is a clique, and if $S_{S_{CG} \bigcup G_{x,CG}} \bigcup G_{x',S_{CG} \bigcup G_{x,CG}}$ is not a clique then we recursively repeat the same procedure until we end up with a clique structure (note that in every iteration at least one node from the remaining graph is eliminated). In conclusion, we can always find at least one vertex in $G_{x,CG}$ that satisfies Lemma 2. Similarly, we can find such a vertex in $G_{y,CG}$. Therefore, whenever CG is not a clique itself, we can always find at least two vertices that satisfy Lemma 2. This result is particularly important for the design of the distributed version of the MWI scheduling protocol discussed in Section V.

### *C. Proof of the Highest-index Vertex Theorem*

We prove the Theorem by contradiction.

Assume that the remaining graph to be enumerated has two or more vertices with the highest enumeration index (equivalent to having two or more cliques that contain a highest-index vertex). The remaining contention subgraph to be enumerated is essentially a connected component since no vertex can be deleted unless it is connected to two adjacent vertices. Select any pair $(a, b)$ of such highest-index vertices such that the shortest path between $a$ and $b$ does not contain any other highest-index vertex. Let vertex $c$ be an intermediate

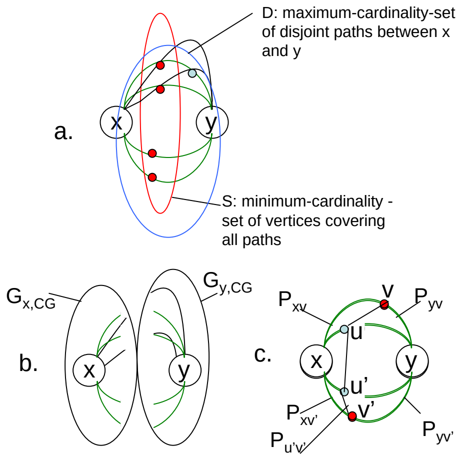
D: maximum-cardinality-set of disjoint paths between x and y
a.
x
y
S: minimum-cardinality - set of vertices covering all paths
G_{x,CG}
G_{y,CG}
b.
x
y
c.
P_{xv}
v
P_{yv}
u
x
y
u'
v'
P_{xv'}
P_{u'v'}
P_{yv'}

Fig. 13. Steps and Terminology for Proving Lemma 2

vertex on this shortest path. $c$ could not have satisfied Lemma 2 before $a$ and $b$ were enumerated since it would be always part of more than one maximal clique (otherwise, knowing that $a$ and $b$ cannot be connected, $c$ would not have been on the shortest path by the definition of the shortest path). Therefore, $c$ has an enumeration index higher than that of $a$ and $b$.
$\Rightarrow$ There is one and only one clique $C$ in the contention graph that can have the highest SEQ1 and SEQ2 enumerated vertex in the case of minimum enumeration.

### *D. Proof of Correctness for the Optimal-SEQ1 Size-computation Algorithms*

We first prove the feasibility of the computations after which we prove their optimality.

Consider an arbitrary link $L$ in the network such that $L$ is element of two maximal set of links sharing the same node each. Let $L$'s SEQ1 enumeration index be $SEQ1_L$ and denote the maximal set that was deleted when containing $L$ by $S_0$. Consider the remaining network graph after $S_0$'s deletion during the pivot tree construction. In the other set $S$ of $L$'s maximal set of links connected to the same node, all the other vertices in $S$ must have a SEQ1 enumeration index less that $SEQ1_L$. Otherwise, the Lemma 2 requirement in the SEQ1 enumeration procedure is being violated. Therefore, a necessary condition for all the contention graph vertices representing $S_0$ to be SEQ1 enumerated (according to the Lemma 2 requirement) is to have all the vertices corresponding to all $V_{T_R,S_0}$'s children enumerated first. $V_{T_R,S_0}$'s parent vertex has been excluded among the other vertices adjacent to $V_{T_R,S}$ as such a requirement would essentially lead to contradicting the Highest-index Vertex Theorem when not contradicting Lemma 2 itself.

Now consider the set of links $W$ in the network graph that correspond to a set of a pivot tree vertex and its children vertices. Any two links in $W$ have their end nodes within three hops in the network graph. Therefore, no two links in the network graph represented within a set of a pivot tree vertex and its children can be enumerated by the same SEQ1 value as this would violate the Weight-Phase SEQ1 Enumeration Requirements under multi-channel access constraints. This condition, together with the Lemma 2 necessary condition for multi-channel networks, makes SEQ1 enumeration always feasible in multi-channel contention graphs and is considered by the Minimum-SEQ1 Size-computations algorithm for Multi-channel Networks.

For the case of single-channel access constraints, we consider the set of links $W$ in the network graph that correspond to a three-level sub-tree in the pivot tree graph. Any two links in $W$ have their end nodes within three hops in the network graph. Therefore, no two links in $W$ can be enumerated by the same SEQ1 value as this would violate the Weight-Phase SEQ1 Enumeration Requirements under single-channel access constraints. This condition together with the Lemma2 necessary condition is also sufficient and necessary for SEQ1 enumeration feasibility in single-channel contention graphs and is also considered by the Minimum-SEQ1 Size-computations algorithm for Single-channel Networks.

At this point, we have proven that the SEQ1 computation achieved under both access constraints is feasible. We now proceed to prove by induction that the two algorithms compute the optimal sizes as well:

- In the case of a three-level pivot subtree for multi-channel networks and five-level pivot subtree for single-channel network, the optimal-size computation is correct and the $Self$ set constructed has the minimum lexographic order for the computed size by construction.
- Assume that optimal-size computation is optimal and the $Self$ set constructed has the minimum lexographic order up to level i in an i-level pivot subtree
- By construction, the $Self$ set constructed at level i+1 has the minimum lexographic order. Moreover, the highest-value element in the $Self$ set has the minimum value given the computation values done for the subtrees of every vertex at level i+1. Therefore, the highest-value element in the $Self$ set has the minimum feasible value for among all possible feasible enumerations. Otherwise, the $Self$ sets constructed for at least one of the subtrees at the lower levels does not either satisfy the optimality assumption made above or does not have the minimum lexographic order given its optimality (given that the $Self$ set constructed at level i+1 has the minimum lexographic order and is constructed as mentioned in the computation algorithms). This leads to a contradiction in either cases. Therefore, the $Self$ set constructed at level i+1 has the minimum lexographic order, and the highest-value element in the $Self$ set has the minimum value among all possible feasible enumerations.

$\Rightarrow$ The size computation performed by the two algorithms achieves the optimal values.

### *E. Proof of Correctness for the Optimal-SEQ2 Size-computation Algorithms*

Consider an arbitrary link $L$ in the network such that $L$ is element of two maximal set of links sharing the same node each. Let $L$'s SEQ1 enumeration index be $SEQ1_L$ and denote the maximal set that was deleted when containing $L$ by $S_0$. Consider the remaining network graph after the deletion of

$S_0$ during the pivot tree construction. In the other maximal set $S$ of $L$'s adjacent links, all the other vertices in $S$ must have a SEQ2 enumeration index less that $SEQ2_L$. This is sufficient and necessary to meet the Lemma 2 requirement for multi-channel networks. For single-channel networks, it is sufficient and necessary in every three-level subtree of the pivot tree graph to have each link in the root vertex enumerated after all the links represented in the corresponding child vertex and all the child vertex children are enumerated first. Note that $L$'s adjacent link in $V_{T_R,S_0}$'s parent vertex has been excluded among the other links adjacent to $L$. This is because enumerating the link in $V_{T_R,S_0}$'s parent vertex before $L$ would essentially lead to contradicting the Highest-index Vertex Theorem when not contradicting Lemma 2 itself.

Any link in $S_0$ which was enumerated before L does not have its contention graph vertex adjacent to the contention graph vertex of any of L's adjacent links after $S_0$'s deletion in the case of multi-channel networks (otherwise, such links would have been part of $S_0$ itself). Similarly, in single-channel networks, any link in $S_0$ which was enumerated before L does not have its contention graph vertex adjacent to the contention graph vertex of any of L's two-hop adjacent links after $S_0$'s deletion. Hence, in multi-channel networks, the enumeration size can be increased by one while enumerating two links each represented in one distinct vertex of two adjacent pivot tree vertices. In single-channel networks, the enumeration size can be increased by one while enumerating two links each represented in one distinct vertex of two two-hop adjacent pivot tree vertices.

The two SEQ2 size computation algorithms presented in this paper take into consideration the necessary and sufficient conditions discussed in the first paragraph as well as the flexibility for enumerating multiple links with the same value discussed in the second paragraph. The correctness of the two algorithms can be trivially verified for two-level trees and recursively for every tree that has more levels. The recursive computation holds optimal because each subtree minimum-size SEQ2 computation depends only on the degree of the root vertex and the minimum-size SEQ2 computed for each of its children subtrees. This is the case in multi-channel and single-channel networks as can be deduced from the first two paragraphs in the proof.