On Sample-Path Optimal Dynamic Scheduling for Sum-Queue Minimization in Trees under the K-Hop Interference Model

Srikanth Hariharan and Ness B. Shroff

Abstract—**We investigate the problem of minimizing the sum of the queue lengths of all the nodes in a wireless network with a tree topology. Nodes send their packets to the tree's root (sink). We consider a time-slotted system, and a K-hop interference model. We characterize the existence of *causal sample-path optimal scheduling policies* in these networks, i.e., we wish to find a policy such that at each time slot, for any traffic arrival pattern, the sum of the queue lengths of all the nodes is minimum among *all* policies. We provide an algorithm that takes any tree and K as inputs, and outputs whether a causal sample-path optimal policy exists for this tree under the K-hop interference model. We show that when this algorithm returns FALSE, there exists a traffic arrival pattern for which no causal sample-path optimal policy exists for the given tree structure. We further show that for certain tree structures, even non-causal sample-path optimal policies do not exist. We provide causal sample-path optimal policies for those tree structures for which the algorithm returns TRUE. Thus, we completely characterize the existence of such policies for all trees under the K-hop interference model. The non-existence of sample-path optimal policies in a large class of tree structures implies that we need to study other (relatively) weaker metrics for this problem.**

I. Introduction

We investigate the problem of finding sample-path optimal scheduling policies for minimizing the sum of the queue lengths of all the nodes for *convergecasting* [1] in a wireless network with a tree topology. In the convergecasting problem, nodes send their packets to a sink (which is the root of the tree). The convergecasting problem is of importance in multi-hop wireless networks with a centralized node to which packets are sent. For instance, it is of importance in sensor networks where the centralized node performs fusion of data received from multiple sensor nodes. We are interested in minimizing the sum of the queue lengths of all the nodes in the system as it can be shown to minimize the long term time average delay experienced by packets in the system.

We briefly overview the existing literature. Tassiulas et al., [2] first studied the problem of dynamic scheduling for convergecasting in tandem networks with the sink at the root of the chain. They showed that for the primary (or 1-hop) interference model (where two links that share a node cannot be active at the same time), for any traffic arrival pattern, any maximal matching policy that gives priority to the link closer to the sink is optimal in the sense that the sum of the queue lengths of all the nodes in the network is minimum at each time slot. This is a very strong result because for any sample-path (arrival pattern), this policy is optimal. Further, the policy is causal as it does not require knowledge of future arrivals. Ji et al., [3] develop a sample-path optimal policy for generalized switches with three links, and a heavy-traffic optimal policy for switches with four links. In [4], Gupta et al., have provided a sample-path delay optimal policy for a clique wireless network where only one link can transmit at any time, and there are multi-hop flows. Hariharan et al., [5] characterized the existence of causal sample-path optimal policies in trees under the 1-hop interference model. In this work, we generalize this result for the K-hop interference model. In the K-hop interference model, no two links that are separated by less than K links can be active during the same time slot. The 1-hop and 2-hop models are well known in the literature, and have been used to model interference in wireless systems. For instance, the 1-hop model is appropriate for Bluetooth [6] and FH-CDMA networks [7], while the 2-hop model is appropriate for IEEE 802.11.

Apart from the literature considering traffic arrivals, [8]–[10] study the convergecasting problem in the absence of arrivals (evacuation time optimality).

Our contributions in this work are the following.

- While previous works have mostly studied the primary interference model, we characterize the existence of sample-path optimal policies for the convergecasting problem in trees under the K-hop interference model.
- We provide an algorithm that takes any tree and K as inputs, and returns a decision on whether a causal sample-path optimal policy exists for the given tree under the K-hop interference model.
- We prove the correctness of this algorithm, i.e., whenever the algorithm returns FALSE, we show that there exists a traffic arrival pattern such that no causal sample-path optimal policy exists for the given tree under the K-hop interference model. When the algorithm returns TRUE, we show that there exists a causal sample-path optimal policy for the given tree. Thus, we characterize the existence of causal sample-path optimal policies for *all* tree structures under the K-hop interference model.

The rest of this paper is organized as follows. In Section II,

S. Hariharan is with the Department of ECE, The Ohio State University. Email: harihars@ece.osu.edu

N. B. Shroff is with the Departments of ECE and CSE, The Ohio State University. Email: shroff@ece.osu.edu

This work was supported in part by NSF grants CNS-0721236, CNS-0721434, CT-0831060, and ARO MURI W911NF-08-1-0238.

we describe the model and notations. In Section III, we provide an algorithm that classifies whether a given tree has a causal sample-path optimal policy under the K-hop interference model. In Sections IV and V, we prove the correctness of the algorithm by showing that no causal sample-path optimal policy can exist when the algorithm returns FALSE, and provide a policy when the algorithm returns TRUE, respectively. In Section VI, we apply our results for the 1-hop and 2-hop interference models. Finally, we conclude the paper in Section VII.

II. System Model and Notations

We model the network as a graph $G(V, E)$, where V is the set of nodes, $|V| = N$, and E is the set of links. The graph G is a tree. We denote 0 to be the sink which is the root of the tree. The sink does not make any transmissions. We assume a time-slotted and synchronized system, and consider a K-hop interference model where two links that are separated by less than K links cannot be active at the same time. As in [2], [8], [5], we assume unit capacity links, i.e., a node can at most transmit one packet to its parent during each time slot. The external packet arrival pattern at nodes is arbitrary and unknown. All packets in the network have sink 0 as the eventual destination.

We use the following notations. Whenever we consider a tandem (or linear) network, we denote a node that is i hops away from the root as node i. For a given node r, m_1^r represents the depth of the tree rooted at r, i.e., it is the length of the deepest branch of r. m_2^r represents the length of the second deepest branch rooted at r. Note that the deepest branch and the second deepest branch belong to different children of r.

III. Classification Algorithm

In this section, we propose an algorithm that identifies whether a causal sample-path optimal policy exists for a given tree under the K-hop interference model. We prove the correctness of this algorithm in later sections.

In algorithm (A_{sp}) given in Table I, we use the *continue* statement to skip the current iteration and start the next iteration. This algorithm uses a subroutine sp (Table I).

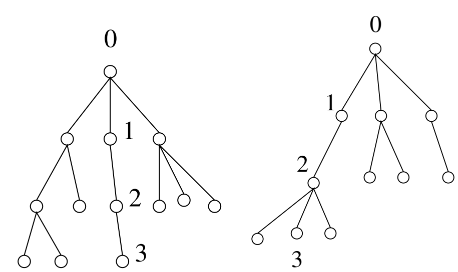
A_{sp} identifies a line in the tree rooted at the sink that is of maximum depth. If there are multiple lines of equal length, the algorithm picks one of them arbitrarily. The nodes in this line are labeled from 0 to m_1^0, where m_1^0 is the length of the deepest branch of the tree. We are only interested in the first $\lfloor \frac{K}{2} \rfloor$ nodes in this line (when $m_1^0 > \lfloor \frac{K}{2} \rfloor$). Starting from the last such node, i.e., node $l = \min(m_1^0, \lfloor \frac{K}{2} \rfloor)$, we investigate the deepest and second deepest branches rooted at l. Note that the deepest branch rooted at l has length $m_1^0 - l$. If m_1^l and m_2^l satisfies certain conditions, we move to node $l - 1$. Otherwise, the algorithm returns that there is no causal sample-path optimal policy for the given tree structure. If the conditions are satisfied at all nodes from 0 to $\min(m_1^0, \lfloor \frac{K}{2} \rfloor)$, the algorithm returns that a causal sample-path optimal policy exists for the given tree structure.

Figure 1 illustrates two examples explaining the functioning of A_{sp} for the 3-hop interference model. Consider Figure 1(a). The depth of the tree is 3, and suppose that A_{sp} chooses the line $0-1-2-3$. Since $\frac{K-1}{2} = 1$, A_{sp} starts at node 1. At node 1, $m_2^1 = 0$. Hence, A_{sp} will *continue* to the previous node in the line. At node 0, $m_2^0 = 3$, and $m_1^0 = 3$. Therefore, $m_1^0 + m_2^0 = 6 > K + 2 = 5$. Hence, subroutine sp will return FALSE, and hence A_{sp} will return FALSE. Now, consider

Inputs: Tree, K
Select a line of maximum depth in the tree
m_1^0 = Length of tree rooted at 0
for $l = \min(m_1^0, \lfloor \frac{K}{2} \rfloor)$ to 0
    Consider the node in the line that is l hops from 0
    $m_1^l = m_1^0 - l$
    m_2^l = Length of the second deepest branch rooted at l
    if $l == \frac{K}{2}$
        continue
    else if $l == \frac{K-1}{2}$
        if $m_2^l \leq \frac{K+1}{2}$
            continue
        else
            return FALSE
        end
    else
        if $m_2^l \leq l$
            continue
        else
            $t = sp(m_1^l, m_2^l, K)$
            if t
                continue
            else
                return FALSE
            end
        end
    end
end
return TRUE

boolean $sp(m_1, m_2, K)$
    if $m_2 \leq \lfloor \frac{K}{2} \rfloor$
        if $m_1 + m_2 \leq K + 1$
            return TRUE
        else
            return FALSE
        end
    else
        if $m_1 + m_2 \leq K + 2$
            return TRUE
        else
            return FALSE
        end
    end

TABLE I
Algorithm A_{sp}



(a) A_{sp} returns FALSE (b) A_{sp} returns TRUE

Fig. 1. Examples illustrating the functioning of A_{sp} for $K = 3$

Figure 1(b). Suppose that A_{sp} chooses the line $0-1-2-3$. At node 1, we again have $m_2^1 = 0$, and hence A_{sp} will *continue* to node 0. At node 0, we now have $m_2^0 = 2$, and $m_1^0 = 3$. Hence, $m_1^0 + m_2^0 = 5 = K+2$, and $m_2^0 = 2 \geq \frac{K+1}{2}$. Hence, subroutine sp will return TRUE, A_{sp} will exit out of the loop (since node 0 has been reached), and will return TRUE for this tree.

We now provide some intuition behind A_{sp}. The following result shows why the choice of the line (of maximum depth) does not affect the outcome of A_{sp}.

Theorem 1. *If there are two or more lines of maximum depth in the tree, and A_{sp} returns TRUE (or FALSE) for an arbitrarily chosen line, it will return TRUE (or FALSE, respectively) even if any other line of equal depth is chosen.*

Proof: Suppose that there are n lines of maximum depth in the tree, p_1, p_2, ..., p_n. WLOG, suppose that p_1 was chosen, and that A_{sp} had returned FALSE for p_1. We have the following cases.
Case 1: K is odd, and at $l = \frac{K-1}{2}$ in p_1, $m_2^l > \frac{K+1}{2}$. Then, $m_2^l \geq \frac{K+3}{2}$. Consider any line p_i, $i > 1$. Consider the node A at which p_1 branches away from p_i. If node A is closer to the root than node $l = \frac{K-1}{2}$, then two longest lines at node A are those corresponding to p_1 and p_i. Further, $m_1^A \geq \frac{K+3}{2}$ and $m_2^A \geq \frac{K+3}{2}$. Therefore, $m_1^A + m_2^A \geq K+3 > K+2$. Hence, subroutine sp would have returned FALSE at node A even if the line p_i had been chosen instead of p_1. If node A is the same as node l, or is farther away from the root than node $l = \frac{K-1}{2}$, then it immediately follows that whether p_1 or p_i had been chosen, A_{sp} would have returned FALSE at node $l = \frac{K-1}{2}$.
Case 2: At a node $l < \lfloor \frac{K}{2} \rfloor$ in p_1, sp returns FALSE, and hence A_{sp} returns FALSE. As before, consider any line p_i, $i > 1$, and consider the node A at which p_1 branches away from p_i. If node A is closer to the root than node l, then two longest lines at node A are those corresponding to p_1 and p_i. Since sp returns FALSE at node l, and $m_1^A + m_2^A > m_1^l + m_2^l$, sp would have returned FALSE at node A even if we had chosen line p_i instead of p_1.

Hence, we have shown that if A_{sp} returns FALSE for an arbitrarily chosen line of maximum depth, then it will return FALSE even if any other line (of maximum depth) is chosen.

Suppose that A_{sp} had returned TRUE for p_1. We show by contradiction that it cannot return FALSE even if any other line (of maximum depth) had been chosen. Assume that A_{sp} returns FALSE for line p_j, $j > 1$. By the previous result, it follows that if A_{sp} returns FALSE for p_j, it will return FALSE for p_1, which contradicts the fact that it returned TRUE for p_1. ■

Remark 1: A_{sp} only considers the two longest branches at any node (at distance at most $\lfloor \frac{K}{2} \rfloor$ from the sink) in a line of maximum depth. The intuition behind this is as follows. Suppose we select the deepest node in the longest branch, and the deepest node in the second longest branch at a node l. If these two nodes cannot simultaneously transmit according to the K-hop interference model, then it implies that no two nodes in different branches of l can simultaneously transmit according to the K-hop interference model. In fact, we will see that in many tree structures, there exists no causal sample-path optimal policy even if there is only a possibility of having simultaneous transmissions (under the K-hop interference model).

Remark 2: At any node $l \leq \lfloor \frac{K}{2} \rfloor$ (in a line in the deepest branch of the tree), if $m_2^l \leq l$, A_{sp} skips to the next node closer to the sink in that line. The intuition is that if $m_2^l \leq l$, no two nodes in two different branches of l need to simultaneously transmit even if they can potentially do so under the K-hop interference model (refer to [11]). The implication of this is that we can hope to convert the tree into an equivalent line network [8], [5], and schedule the tree as though the schedule is in a line network. Section V explains this in detail.

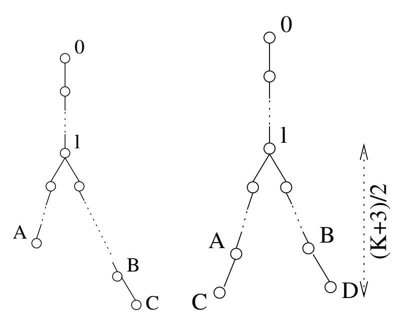
Remark 3: A_{sp} does not consider nodes that are at distance greater than $\lfloor \frac{K}{2} \rfloor$ from the sink. The reasoning is similar to the previous case. Consider any node $l > \lfloor \frac{K}{2} \rfloor$. Even if $m_2^l \geq l$, then the deepest node in the second deepest branch at l is at least $K+1$ hops away from the sink. We will see in Section V that we can schedule many of these trees according to a schedule in an equivalent linear network.

IV. Proof of "Non-Existence"

In this section, we prove one part of the correctness of Algorithm A_{sp}, i.e., we show that whenever A_{sp} returns FALSE for an input tree structure, there exists a traffic arrival pattern such that there exists no causal sample-path optimal policy for the tree. Further, we also observe that, in some cases, even a non-causal sample-path optimal policy does not exist for the given tree structure.

Theorem 2. *For a given tree and K as inputs to Algorithm A_{sp}, consider any node l, $0 \leq l < \lfloor \frac{K}{2} \rfloor$, in a line in the deepest branch in the tree. If the length of the second deepest branch rooted at node l is longer than l, i.e., $m_2^l > l$ and $sp(m_1^l, m_2^l, K)$ returns FALSE, there exists no causal sample-path optimal policy for the given tree structure under the K-hop interference model.*

Proof: We prove this result by contradiction.



(a) $m_2^l \leq \lfloor \frac{K}{2} \rfloor$ (b) $m_2^l > \lfloor \frac{K}{2} \rfloor$

Fig. 2. No causal sample-path optimal policy

Suppose that the result was not true. For a given l, $0 \leq l < \lfloor \frac{K}{2} \rfloor$, we consider the simplest tree structure that does not satisfy the conditions in the subroutine sp.

Case 1: $l < m_2^l \leq \lfloor \frac{K}{2} \rfloor$ and $m_1^l + m_2^l = K+2$.

Consider a tree with the first l links in a line (as shown in Figure 2(a)), and with two lines at node l, one of depth $m_2^l > l$, and the other of depth $m_1^l = K+2-m_2^l$.

Consider the following traffic arrival pattern. At time $t = 0$, there exists one packet each at nodes A and B. Node A is at a depth $l+m_2^l$ from the root of the tree, and node B is at a depth $l+m_1^l-1$. Since $m_2^l \leq \lfloor \frac{K}{2} \rfloor$, $m_1^l \geq K - \lfloor \frac{K}{2} \rfloor + 2 \geq \lfloor \frac{K}{2} \rfloor + 2$. Hence, $m_2^l < m_1^l - 1$. Therefore, the packet at node A is at a lower distance from the root than the packet at node B. Further, nodes A and B cannot be scheduled simultaneously under the K-hop interference model. In fact, since $l < \lfloor \frac{K}{2} \rfloor$, the only nodes in this network that can simultaneously transmit are A and C. This implies that we need to schedule A before node B. This is because if we schedule node B instead, the time for the first packet to exit the system will be $m_1^l + l$. On the other hand, if we schedule node A, the time for the first packet to exit the system will only be $m_2^l + l < m_1^l + l$. Hence, in any sample-path optimal policy, we always need to schedule the closest packet to the root of the tree. However, suppose that we schedule A at time $t = 0$, and a packet arrives at node C at time $t = 1$. Further, assume that there are no other packet arrivals in the system. Then the total time (after the slot $t = 1$) for the three packets to exit the system is $l+m_2^l-1+l+m_1^l-1+l+m_1^l = 3l+K+m_1^l$.

On the other hand, if we had scheduled B during the first time slot, since nodes A and C can transmit simultaneously, the packets at nodes A and C can be transmitted to their respective parents during the same time slot. Therefore, the total time for the three packets to exit the system is now $l+m_1^l-2+l+m_2^l+l+m_1^l-1 = 3l+K+m_1^l-1 < 3l+K+m_1^l$. Thus, we get a contradiction for this case.

We can further infer from the above counterexample that *even a non-causal sample-path optimal policy cannot exist for this tree structure under the K-hop interference model*. This is because even if we knew that a packet was going to arrive at node C at slot $t = 1$, we would still have to schedule node A at slot $t = 0$ since it is closer to the root than node B.

Case 2: $m_2^l > \lfloor \frac{K}{2} \rfloor$ and $m_1^l + m_2^l = K + 3$.

Suppose K is even. Then, $m_2^l = \frac{K}{2} + 1$ and $m_1^l = \frac{K}{2} + 2$. From Case 1, we know that for a network with $m_2^l = \frac{K}{2}$ and $m_1^l = \frac{K}{2} + 2$, there exists a traffic arrival pattern such that there exists no sample-path optimal policy for this network. Since the network with $m_2^l = \frac{K}{2} + 1$ contains the network with $m_2^l = \frac{K}{2}$ as a substructure, there exists no sample-path optimal policy for this structure as well.

Suppose K is odd. Then, $m_2^l = \frac{K+1}{2}$ and $m_1^l = \frac{K+5}{2}$, or $m_1^l = m_2^l = \frac{K+3}{2}$. For the former scenario, from Case 1, we know that there exists no sample-path optimal policy even for the network with $m_2^l = \frac{K-1}{2}$ and $m_1^l = \frac{K+5}{2}$. Hence, it follows that there exists no sample-path optimal policy for this network as well. For the latter scenario, we construct the following traffic arrival pattern.

Consider the tree shown in Figure 2(b) where the first l links are in a line and the node l has two branches, each of length $\frac{K+3}{2}$. Suppose that at $t = 0$, there is one packet each at nodes A and B which are both at depth $\frac{K+1}{2}$ from l as shown in the figure. A and B cannot simultaneously transmit under the K-hop interference model. Also, since $l < \frac{K-1}{2}$, the nodes in the network that can simultaneously transmit in the same slot are A and D, or B and C, or C and D. Since both the packets are at the same depth from the root, without having knowledge of future traffic arrivals, we can only arbitrarily choose A or B to schedule. Suppose we choose A to schedule, and a packet arrives at node D at slot $t = 1$. Then, the total time after this slot for the three packets to exit the system is $l + \frac{K-1}{2} + l + \frac{K+1}{2} + l + \frac{K+3}{2} = 3l + \frac{3K+3}{2}$.

On the other hand, if we knew that a packet was going to arrive at D at slot $t = 1$, we could have scheduled B during the first time slot. In this case, A and D could have simultaneously been scheduled in a later time slot. Hence, the total time after the first slot for these packets to exit the system is $l + \frac{K-1}{2} + l + \frac{K+1}{2} + l + \frac{K+1}{2} = 3l + \frac{3K+1}{2} < 3l + \frac{3K+3}{2}$. Thus, we get a contradiction.

Hence, Theorem 2 follows. ■

Theorem 3. *Let K be an odd number. Let l be the $\frac{K-1}{2}^{th}$ node in a line in the deepest branch in the tree. If the length of the second deepest branch rooted at node $\frac{K-1}{2}$ is longer than $\frac{K+1}{2}$, i.e., $m_2^l > \frac{K+1}{2}$, then there exists no causal sample-path optimal policy for this tree structure.*

Proof: Similar to the proof of Theorem 2, we can again construct a traffic arrival pattern for which there exists no causal sample-path optimal policy. The details can be found in our technical report [11]. ■

Theorem 4. *Algorithm A_{sp} correctly identifies tree structures for which there exist no causal sample-path optimal policy under the K-hop interference model, i.e., whenever A_{sp} returns FALSE for a given tree structure, there exists no causal sample-path optimal policy for that tree structure.*

Proof: This result follows from Theorems 2 and 3. Theorem 2 proves it for $0 \leq l < \lfloor \frac{K}{2} \rfloor$, and Theorem 3 shows the result for $l = \lfloor \frac{K}{2} \rfloor$. ■

The implication of the above results is that sample-path optimal policies may only exist in restricted tree topologies.

V. "Existence" Proofs

In this section, we develop sample-path optimal policies for all tree structures for which Algorithm A_{sp} returns TRUE under the K-hop interference model. We divide the set of trees for which A_{sp} returns TRUE into six classes, and develop a sample-path optimal policy for each class.

A. Classification of Trees

We now classify the tree structures for which A_{sp} returns TRUE into six classes. The following theorem forms an initial basis for classification. It classifies trees for which the depth of the tree must be bounded by K in order for A_{sp} to return TRUE, and those for which the depth need not be bounded.

Theorem 5. *For any tree for which A_{sp} returns TRUE, m_1^0 can be greater than K if and only if the following conditions are satisfied.*

1) *If K is odd, for each l such that $0 \leq l < \frac{K-1}{2}$, $m_2^l \leq l$.*
2) *If K is even, for each l such that $0 \leq l < \frac{K}{2}$, $m_2^l \leq l$.*

Proof: We first show that m_1^0 can be greater than K if the given conditions are satisfied. Whether K is odd or even, it can be immediately seen from A_{sp} that if the corresponding conditions (for odd and even K) are satisfied, there is no constraint on m_1^l for any l (since if the conditions are satisfied, we have the *continue* statement). Therefore, the depth of the tree can be any arbitrary quantity.

We prove the converse by contradiction. We consider the case where K is odd. The proof when K is even can be found in our technical report [11].

Case 1: Suppose that for some l such that $0 \leq l < \frac{K-1}{2}$, $m_2^l > l$, and $m_1^0 > K$. First consider the case, $m_2^l \leq \frac{K-1}{2}$. Note that $m_1^l = m_1^0 - l > K - l$. Consider any m_1^l and m_2^l such that $m_1^l \geq m_2^l$, $l < m_2^l \leq \frac{K-1}{2}$, and $m_1^l > K - l$. Therefore, we have $m_1^l \geq K + 1 - l$, and $m_2^l \geq l + 1$. Then, $m_1^l + m_2^l \geq K + 1 - l + l + 1 = K + 2$. Therefore, according to subroutine sp, A_{sp} will return FALSE for this tree structure. This contradicts our assumption that A_{sp} returns TRUE.

Case 2: Consider any l such that $0 \leq l < \frac{K-1}{2}$, $m_2^l > \frac{K-1}{2}$, and $m_1^0 > K$. Since $l < \frac{K-1}{2}$, $l \leq \frac{K-3}{2}$, and hence, $m_1^l = m_1^0 - l > \frac{K+3}{2}$. Therefore, $m_1^l \geq \frac{K+5}{2}$ and $m_2^l \geq \frac{K+1}{2}$. Hence, $m_1^l + m_2^l > K + 2$. Therefore, subroutine sp will return FALSE, and hence A_{sp} will return FALSE, resulting in a contradiction. ∎

Corollary 1 (Corollary to Theorem 5)**.** *For any tree for which A_{sp} returns TRUE, m_1^0 must be bounded by K if and only if the following conditions are satisfied.*

1) *If K is odd, $m_2^l > l$ for at least one node l ($0 \leq l < \frac{K-1}{2}$) in a line in the deepest branch of the tree.*
2) *If K is even, $m_2^l > l$ for at least one node l ($0 \leq l < \frac{K}{2}$) in a line in the deepest branch of the tree.*

We now have the following classes of trees.

Class I: The tree satisfies the conditions in Corollary 1. Hence, the depth of the tree is bounded by K. In addition, it satisfies the condition that at each node l in a line in the deepest branch of the tree, $m_1^l + m_2^l \leq K + 1$. It can be easily seen that no two links in such a tree can be simultaneously scheduled (due to the K-hop interference model).

Class II: K is odd, the tree satisfies the first condition in Corollary 1, and does not satisfy the additional condition for Class I trees.

Class III: K is even, the tree satisfies the second condition in Corollary 1, and does not satisfy the additional condition for Class I trees. Simultaneous transmissions are possible among certain links in trees belonging to Classes II and III.

Class IV: The tree satisfies the conditions in Theorem 5. Hence, the tree can be of arbitrary depth. In addition, it satisfies the condition that at node $l = \lfloor \frac{K}{2} \rfloor$, $m_2^l \leq l$.

Class V: K is odd, the tree satisfies the first condition in Theorem 5, and it does not satisfy the additional condition for Class IV trees, i.e., for $l = \lfloor \frac{K}{2} \rfloor$, $m_2^l > l$. Note that since A_{sp} returns TRUE for this class, it follows that for $l = \frac{K-1}{2}$, $m_2^l \leq \frac{K+1}{2}$.

Class VI: K is even, the tree satisfies the second condition in Theorem 5, and it does not satisfy the additional condition for Class IV trees, i.e., for $l = \lfloor \frac{K}{2} \rfloor$, $m_2^l > l$.

Theorem 6. *Classes I-VI characterize all trees for which Algorithm A_{sp} returns TRUE under the K-hop interference model, i.e., for any tree that does not belong to Classes I-VI, A_{sp} returns FALSE.*

Proof: From Theorem 5, Corollary 1, and the definitions of Classes I-VI, the result follows. ∎

We now provide a causal sample-path optimal policy for each class of trees.

B. Class I

We recall that Class I is the class of trees for which the depth of the tree must be bounded by K, and no two links in the tree can simultaneously transmit. Figure 8(a) provides an example of Class I trees under the 2-hop interference model. We define the causal policy, π_0^I, for Class I trees as follows.

Policy π_0^I: *At each slot, determine the packet i whose hop distance to the sink is minimum among all packets in the system, and schedule it. If there are multiple such packets, schedule one of them arbitrarily.*

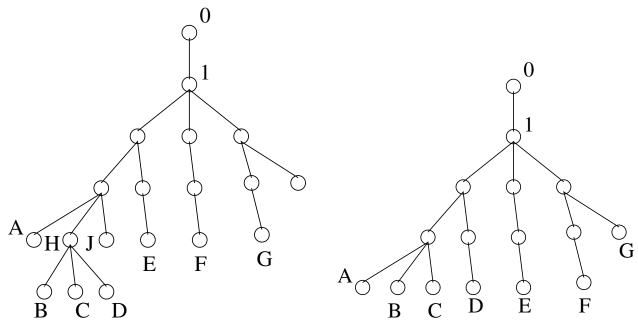
C. Class II

We study Class II trees in this section. K is assumed to be odd. At exactly one node l in a line in the deepest branch of the tree, $m_1^l + m_2^l = K + 2$, where $m_2^l = \frac{K+1}{2}$, and at all other nodes l in the line, $m_1^l + m_2^l \leq K + 1$. It is easy to see that if there are two or more nodes for which $m_1^l + m_2^l = K + 2$, A_{sp} will return FALSE.

Consider the node l for which $m_1^l + m_2^l = K + 2$. The nodes that are at depth $\frac{K+1}{2}$ in the second deepest branch of node l, and the nodes that are at depth $\frac{K+3}{2}$ in the deepest branch of node l are separated by K links. Therefore, one of these nodes in the second deepest branch and one of these nodes in the deepest branch can transmit simultaneously in a slot. Further, since $m_1^l + m_2^l \leq K + 1$ for all other nodes l, no other nodes in the tree can transmit simultaneously.

We define the following notation. Consider the node l in a line in the deepest branch of the tree for which $m_1^l + m_2^l = K + 2$. We define N_1 to be the set of packets at leaf nodes that are at depth $\frac{K+3}{2}$ from node l in the deepest branch rooted at node l, and N_2 to be the set of packets at leaf nodes at depth $\frac{K+1}{2}$ from node l in any other branch rooted at node l. For example, consider Figure 3(a). This represents a Class II tree under the 5-hop interference model. At $l = 1$, $m_1^1 + m_2^1 = 4 + 3 = 7 = K + 2$. Hence, packets at nodes B, C, and D belong to N_1, and those at nodes F, and G belong to N_2. Further, note that packets at nodes A, H, J, and E neither belong to N_1 nor to N_2 because they are in the same branch of node $l = 1$ as the packets that belong to N_1.

Policy π_0^{II}: *At each time slot, schedule a packet that is closest to the root of the tree. If multiple packets are at the same depth from the root, a packet can be arbitrarily chosen to schedule in all but the following scenario. Suppose that at node l in a line in the deepest branch of the tree, $m_1^l + m_2^l = K + 2$. Any packet that lies at a node at depth $\frac{K+1}{2}$ from l in the branch of l corresponding to nodes in N_1 is given priority over packets that lie at nodes at depth $\frac{K+1}{2}$ from l in any*



(a) Class II, $K = 5$ (b) Class III, $K = 4$

Fig. 3. Examples for Classes II and III

other branch of l. If the only packets left in the system are those that lie in the set $N_1 \cup N_2$, then select one packet from N_1 and one packet from N_2 to transmit simultaneously.

In Figure 3(a), if A and F both have a packet, then A will be given priority over F. If B and F both have a packet, they will transmit simultaneously to their respective parents.

D. Class III

We now consider Class III trees. K is assumed to be even. At exactly one node l in a line in the deepest branch of the tree, $m_1^l + m_2^l = K + 2$, where $m_2^l = \frac{K}{2} + 1$, and at all other nodes l in the line, $m_1^l + m_2^l \leq K + 1$.

Consider the node l for which $m_1^l + m_2^l = K + 2$. This means that l has at least two branches of depth $\frac{K}{2} + 1$. Assume that l has p branches of depth $\frac{K}{2} + 1$, $p \geq 2$. We define N_i, $i = 1, 2, ..., p$, to be the set of packets at leaf nodes that are at depth $\frac{K}{2} + 1$ in the i^{th} branch of l. Consider any node $a_1 \in N_1$, $a_2 \in N_2$,..., $a_p \in N_p$. $a_1, a_2, ..., a_p$ can all transmit during the same slot since any two nodes in the set $\{a_1, a_2, ..., a_p\}$ are separated by K links. Further, since $m_1^l + m_2^l \leq K + 1$ for all other nodes l, no other nodes in the tree can transmit simultaneously. We provide an example to explain this scheduling (Figure 3(b), $K = 4$). At $l = 1$, we have $m_1^1 = m_2^1 = 3$. Hence, $m_1^1 + m_2^1 = 6 = K + 2$. Also, there are three branches of depth 3 from node 1. Therefore, $p = 3$. Packets at nodes A, B, C, and D belong to N_1, those at E and F belong to N_2 and N_3, respectively. Note that packets at node G do not belong to $N_1 \cup N_2 \cup N_3$.

We now propose policy π_0^{III} for this class of trees.

Policy π_0^{III}: *At each time slot, schedule a packet that is closest to the root of the tree as long as it does not belong to $N_1 \cup N_2 \cup ... \cup N_p$. If multiple packets are at the same depth from the root, a packet can be arbitrarily chosen to schedule. If the only packets left in the system belong to $N_1 \cup N_2 \cup ... \cup N_p$, select one packet in each of $N_1, N_2, ..., N_p$ (as long as a packet exists), and schedule these packets simultaneously during that slot.*

In Figure 3(b), if there is one packet each at nodes A, E, and F, these packets will be scheduled simultaneously.

E. Class IV

We now discuss tree structures for which the depth need not be bounded by K in order for a causal sample-path optimal policy to exist. We study Class IV trees in this section. These trees satisfy the conditions in Theorem 5. Hence, the tree can be of arbitrary depth. In addition, they satisfy the condition that for each l such that $0 \leq l \leq \lfloor \frac{K}{2} \rfloor$, $m_2^l \leq l$. This additional condition ensures that no two nodes that are at distance $K + 1$ or lesser from the sink can transmit simultaneously under the K-hop interference model. Figures 6(b) and 9(a) are examples of Class IV trees for the 1-hop and 2-hop interference models, respectively.

We show that these trees can be scheduled as though the schedule is in a linear network under the K-hop interference model. We recall the definition of the equivalent linear network for a given tree below [8].

For a tree network $G(V, E)$ with V nodes and E edges, where each node i has β_i packets during a given time slot, the equivalent linear network $G(V_l, E_l)$ is defined as follows: $V_l = \{0, 1, ..., N\}$, $E_l = \{(i-1, i), 1 \leq i \leq N\}$ where $N = \max_{i \in V}(d(0, i))$. $d(0, i)$ represents the distance of node i from the sink node 0. Further, each node $j \in V_l$ has α_j packets during the same time slot, where $\alpha_j = \sum_{i \in V : d(0,i) = j} \beta_i$.

Figure 4 gives an example of this transformation. The farthest node in the tree is 3 hops away from the sink. Therefore, the equivalent linear network has 3 nodes and the sink. The number of packets at each node is mentioned in the figure. The total number of packets from nodes that are 2 hops away from the sink is 7 (=3+4), and that from nodes that are 3 hops away from the sink is 9 (=6+1+2). Therefore, the equivalent linear network has 7 packets in node 2, and 9 packets in node 3.

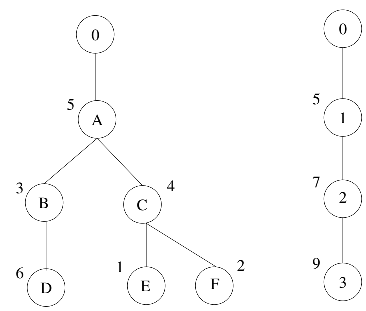


Fig. 4. Equivalent Linear Network

We now propose policy π_0^{IV} for Class IV trees.

Policy π_0^{IV}: *Consider node 1 in the equivalent linear network. If node 1 has a packet, schedule it. Else, go to the next node. For any node $i \leq K + 1$, if none of the nodes $1, 2, ..., i - 1$ have been scheduled, and if node i has a packet, schedule node i. Otherwise, go to the next node. For any node $i > K + 1$, if none of the nodes in the set $\{i - 1, i - 2, ..., i - K\}$ have been scheduled, and if node i has a packet, schedule node i. Otherwise, go to the next node.*

This policy is a generalized version of the policy in [2] for a linear network under the 1-hop interference model.

We recall some of the implications of this policy using the 1-hop interference model as an example (as noted in [5]).

Remark 4: According to policy π_0^{IV}, any node i in the equivalent linear network can schedule at most one packet during any time slot. This means that among all nodes that are i hops away from node 0 in the original tree, at most one packet will be scheduled. Note that multiple nodes (at the same distance from the sink) can potentially schedule their transmissions simultaneously if they don't have the same parent (under the 1-hop interference model). This implies that even without scheduling a *maximal* set of non-interfering links, this policy is optimal.

Remark 5: Suppose that a node i in the equivalent linear network is selected to schedule during a certain slot according to π_0^{IV}. Consider nodes that are i hops away from node 0 in the original tree that have at least one packet to schedule. One of these nodes can be chosen *arbitrarily* to schedule its packet during that slot. This means that the optimal solution neither depends on the structure of the Class IV tree nor the number of packets at each node. For example, in Figure 4, we can arbitrarily choose to schedule one of $\{D, E, F\}$ according to π_0^{IV}.

F. Class V

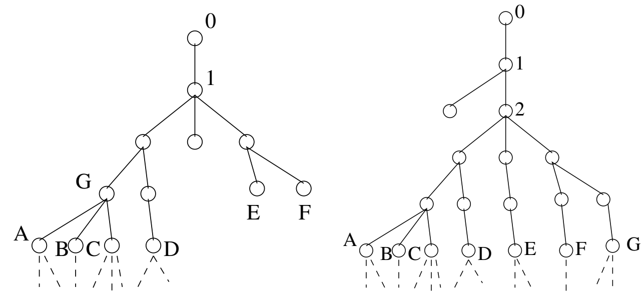
We investigate causal sample-path optimal policies for Class V trees in this section. K is odd, the tree satisfies the first condition in Theorem 5, and does not satisfy the additional condition that Class IV trees satisfy. Therefore, at node $l = \frac{K-1}{2}$ in a line in the deepest branch of the tree, $m_2^l = \frac{K+1}{2}$.

We first define a similar notation as used for Class II trees. Consider the node $l = \frac{K-1}{2}$ in a line in the deepest branch of the tree. We define N_1 to be the set of packets at nodes at depth $\geq \frac{K+3}{2}$ from node l in the deepest branch rooted at node l, and N_2 to be the set of packets at leaf nodes at depth $\frac{K+1}{2}$ from node l in any other branch rooted at node l. A packet in N_2 and a packet in N_1 can potentially simultaneously transmit according to the K-hop interference model.

We now propose policy π_0^V for this class of trees.

Policy π_0^V: *At each time slot, do the following. For packets that are at distance $\leq K-1$ from the sink, schedule a packet that is closest to the sink, say, at distance $d \leq K-1$ to the sink. Do not schedule any packets at distance $\leq d+K$ from the sink. Schedule packets in N_1 at distance $> d+K$ from the sink according to a schedule in an equivalent linear network. If there are no packets at distance $d \leq K-1$ from the sink, consider node $l = \frac{K-1}{2}$ in a line in the deepest branch of the tree. Any packet that lies at a node at depth $\frac{K+1}{2}$ from l in the branch of l corresponding to nodes in N_1 is given priority over packets that lie at nodes at depth $\frac{K+1}{2}$ from l in any other branch of l, and the rest of the schedule for packets in N_1 (at distance $> 2K$ from the sink) is according to one in an equivalent linear network. If the only packets left in the system are those that lie in the set $N_1 \cup N_2$, then select the packet closest to the sink from N_1 and one packet from N_2 to transmit simultaneously. Schedule the rest of the packets in N_1 according to a schedule in an equivalent linear network.*

We provide an example to explain this policy. Consider Figure 5(a). This is an example of a Class V tree under the 3-hop interference model. At node $l = 1$, $m_1^1 \geq 3$, and $m_2^1 = 2$. Packets in nodes A, B, C, D, and in the sub-trees rooted at these nodes belong to N_1. Packets at nodes E and F belong to N_2. If there is one packet each at G and E, then G will be given priority over E. If there is one packet each at A and E, then these packets will be scheduled simultaneously. If B is scheduled during a particular slot, then since B is 4 hops away from the sink 0, only packets that are at least 8 hops away from the sink will be scheduled. This schedule is equivalent to one in an equivalent linear network.



(a) Class V, $K = 3$ (b) Class VI, $K = 4$

Fig. 5. Examples for Classes V and VI

G. Class VI

Finally, we investigate Class VI trees. K is even, the tree satisfies the second condition in Theorem 5, and does not satisfy the additional condition that Class IV trees satisfy. Therefore, at node $l = \frac{K}{2}$ in a line in the deepest branch of the tree, the branches originating from this node can be of arbitrary depth. Since these trees do not satisfy the additional condition that Class IV trees satisfy, there exist at least two branches at node l whose depth from l is greater than l.

We define a similar notation as used for Class III trees. Consider the node $l = \frac{K}{2}$ in a line in the deepest branch of the tree. Suppose that l has p branches whose depth from l is greater than l. We define N_i to be the set of packets at nodes at depth $\geq \frac{K}{2} + 1$ from node l in branch i, $i = 1, 2, ..., p$.

We now propose policy π_0^{VI} for this class of trees.

Policy π_0^{VI}: *At each time slot, do the following. For packets that are at distance $\leq K$ from the sink, schedule a packet that is closest to the sink, say, at distance $d \leq K$ to the sink. Do not schedule any packets at distance $\leq d+K$ from the sink. Consider packets at distance $\geq d+K+1$ from the sink. These packets belong to $N_1 \cup N_2 \cup ... \cup N_p$. For N_i, $i = 1, 2, ..., p$, schedule packets in N_i at distance $\geq d+K+1$ from the sink according to a schedule in an equivalent linear network. The schedule of packets in N_i is independent of the schedule of packets in N_j for any $i \neq j$.*

We provide an example to explain this policy. Consider Figure 5(b). This represents a Class VI tree for $K = 4$. The tree rooted at node $\frac{K}{2} = 2$ can be arbitrary. This node has 3 branches of depth at least $\frac{K}{2} + 1 = 3$. Therefore, $p = 3$. Packets in nodes A, B, C, D, and in their subtrees belong to N_1. Those in E, and its subtree belong to N_2, and those in F, G, and their subtrees belong to N_3. If there is one packet each at nodes C, E, and G, these packets will be transmitted simultaneously to their respective parents. The three branches of node 2 can be converted into three equivalent linear networks, and the schedule in each branch till the packet reaches a node at distance $K+1 = 5$ from the sink is according to a schedule in an equivalent linear network for that branch.

The following result shows the optimality of policy π_0^i for Class i trees, $i = I, II, ..., VI$.

Theorem 7. *For Class i of trees, $i = I, II, ..., VI$, policy π_0^i minimizes the sum of the queue lengths of all the nodes in the given tree under the K-hop interference model at each time slot and for any traffic arrival pattern.*

Proof: Due to space limitations, and the lengthy nature of sample-path optimality proofs in general, we provide the details in our technical report [11]. The proof for each class consists of three components: a recursive relationship for the time at which each packet leaves the system, a proof for optimality in the absence of arrivals, and finally a proof for optimality when there are packet arrivals in the system. ∎

Remark 6: One of the key intuitions to the fact that there exists causal sample-path optimal policies for these six classes of trees is the relationship of the scheduling policy to that in an equivalent linear network (or some extensions of it). Classes I and IV can be scheduled according to a schedule in an equivalent linear network, while the optimal schedules for the other classes is a modification of a schedule in an equivalent linear network.

Remark 7: It can be shown from Theorem 7, and from Little's law that π_0^i minimizes the long term time average delay in the system for Class i trees, $i = I, II, ..., VI$.

Theorem 8. *Algorithm A_{sp} correctly classifies trees for which a causal sample-path optimal policy exists, and those for which such a policy does not exist.*

Proof: The result for non-existence of causal sample-path optimal policies follows from Theorem 4, and that for existence follows from Theorems 6 and 7 for Classes I-VI. ∎

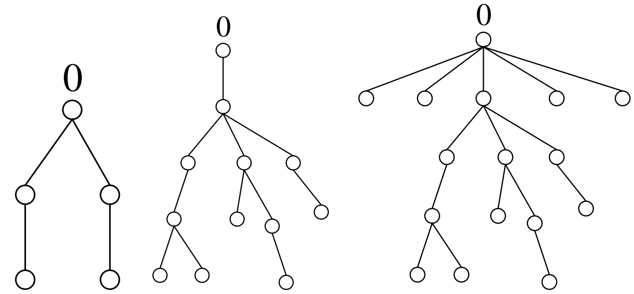
Thus, we have completely characterized the existence of causal sample-path optimal policies for all trees under the K-hop interference model for the convergecasting problem.

VI. Examples - 1-hop and 2-hop

In this section, we apply the results in Sections IV and V to the 1-hop and 2-hop interference models, and completely characterize the tree structures for which causal sample-path optimal policies exist for these interference models. The results for the 1-hop interference model derived in [5] serves as a sanity check for the results in this work.

A. 1-hop

Since, $K = 1$, we have $\frac{K-1}{2} = 0$, and $\frac{K+1}{2} = 1$. We first look at the tree structures for which no causal sample-path optimal policy exists. From Theorem 3, at node $l = 0$ in a line in the deepest branch of the tree, if $m_2^0 > 1$, there exists no causal sample-path optimal policy for the given tree. Thus, if the root of the tree has more than one child that is not a leaf node, there exists no causal sample-path optimal policy for the given tree. This implies that the tree in Figure 6(a) has no causal sample-path optimal policy. This verifies Theorem 3 in [5].



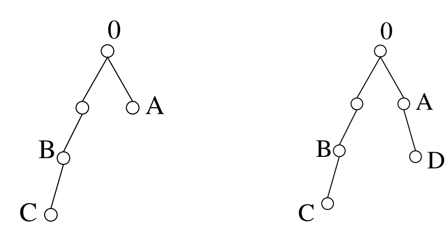
(a) No optimal policy (b) Class IV (c) Class V

Fig. 6. Existence of sample-path optimal policies for $K = 1$

We now consider trees for which a causal sample-path optimal policy exists. Since K is odd, we only need to consider Classes I, II, IV, and V. Since $\frac{K-1}{2} = 0$, from Corollary 1, it follows that there are no Class I and Class II trees under the 1-hop interference model. For Class IV trees, for each $0 \le l \le \frac{K-1}{2}$, $m_2^l \le l$. This means that at the root ($l = 0$), $m_2^0 \le 0$. Therefore, the root can have only one child. The rest of the tree can be arbitrary. For such trees, we can transform the tree into an equivalent linear network, and schedule the equivalent linear network according to the 1-hop interference model. This concurs with Theorem 1 in [5]. Finally, for Class V trees, at node $l = \frac{K-1}{2} = 0$, $m_2^0 \le 1$. This means that if the root has at most one non-leaf child, then the tree has a causal sample-path optimal policy. Further, the optimal policy (for Class V trees) is to always give priority to that child of the root that is not a leaf node (when there is contention among the root's children), and to schedule the rest of the tree according to the equivalent linear network schedule. From Theorem 2 in [5], we can verify the correctness of both the optimal policy, and the structure of this class of trees. Figures 6(b) and 6(c) show examples of Class IV and Class V trees for the 1-hop interference model, respectively.

B. 2-hop

Consider tree structures for which no causal sample-path optimal policy exist under the 2-hop interference model. Since $\frac{K}{2} = 1$, by Theorem 2, $m_2^0 > 0$. $sp(m_1^0, m_2^0, 2)$ will return FALSE if $m_1^0 + m_2^0 > K + 1 = 3$ when $m_2^0 = 1$, and $m_1^0 + m_2^0 > K + 2 = 4$ when $m_2^0 = 2$. Therefore, there exists no causal sample-path optimal policy for tree structures for which $m_1^0 = 3$ and $m_2^0 = 1$, and $m_1^0 = 3$ and $m_2^0 = 2$. Figure 7 shows an example of such tree structures.



(a) $m_1^0 = 3$ and $m_2^0 = 1$ (b) $m_1^0 = 3$ and $m_2^0 = 2$

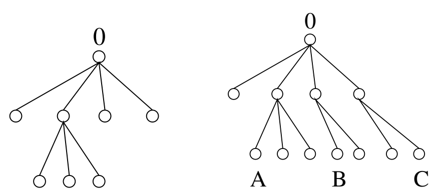
Fig. 7. No causal sample-path optimal policy, $K = 2$

Indeed, if $m_1^0 = 3$ and $m_2^0 = 1$ (Figure 7(a)), suppose that there is one packet at each node A and B at time slot 0. Since node A is closer to the sink, we must schedule node A during the first slot. However, if we do this, and a packet arrives at node C at the beginning of the first slot, it would take five additional time slots for the packets at B and C to reach the sink. On the other hand, if we had scheduled B during the first slot, then since A and C could have been scheduled together, it would only take an additional four time slots for all the packets to reach the sink. Thus, even a non-causal optimal policy does not exist for this tree structure. Since there doesn't exist a sample-path optimal policy when $m_1^0 = 3$ and $m_2^0 = 1$, there cannot exist a sample-path optimal policy when $m_1^0 = 3$ and $m_2^0 = 2$, for we can simply assume the same arrival pattern in A, B, and C, and no packets in node D in Figure 7(b).

We now look at trees for which a causal sample-path optimal policy exists under the 2-hop interference model. Since K is even, we need to consider Classes I, III, IV, and VI. By Corollary 1, if $m_2^0 > 0$, the depth of the tree must be bounded by $K = 2$. Therefore, we have the following two cases for trees whose depth is bounded by K.

Class I: $m_2^0 = 1$ and $m_1^0 \leq 2$, so that $m_1^0 + m_2^0 \leq 3$. Figure 8(a) shows Class I trees for the 2-hop interference model. Clearly, no two nodes in this tree can simultaneously transmit.

Class III: $m_2^0 = 2$ and $m_2^0 = 2$, so that $m_1^0 + m_2^0 = K + 2 = 4$. Figure 8(b) shows Class III trees. The only nodes in this tree that can simultaneously transmit are those at depth 2 from node 0, and in different branches of node 0. For instance, A, B, and C can simultaneously transmit.



(a) Class I (b) Class III

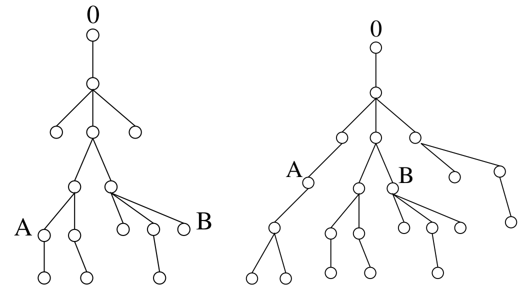
Fig. 8. Examples of Class I and Class III trees for $K = 2$

For trees whose depth need not be bounded, we have the following cases.

Class IV: At $l = 0$ and $l = 1$, we must have $m_2^l \leq l$. Therefore, $m_2^0 = 0$, and $m_2^1 \leq 1$. Figure 9(a) shows an example of this class of trees. These trees can be scheduled according to a schedule in an equivalent linear network. Therefore, for instance, nodes A and B in Figure 9(a) will not transmit simultaneously even though they can potentially do so.

Class VI: For this class, we only have the condition that $m_2^0 = 0$. m_2^1 can be arbitrary. Figure 9(b) illustrates this class. The different branches of node 1 can be scheduled according to an equivalent linear network in each branch, as explained for Class VI trees. However, the entire tree cannot be scheduled according to an equivalent linear network, since, for instance, node A and node B in Figure 9(b) must transmit simultaneously if they both have packets to transmit.

Thus, we have illustrated our results for completely characterizing the existence of causal sample-path optimal policies



(a) Class IV (b) Class VI

Fig. 9. Examples of Class IV and Class VI trees for $K = 2$

for trees under the 1-hop and 2-hop interference models.

VII. Conclusion

We have studied the existence of causal sample-path optimal policies that, at each time slot, minimize the sum of the queue lengths of all the nodes in a multi-hop wireless network with a tree topology under the K-hop interference model, for any sample-path traffic arrival pattern. We provided necessary and sufficient conditions for the existence of such policies, and rigorously proved their correctness. We observed that causal sample-path optimal policies exist for a large class of trees. Surprisingly, in many cases, the tree can be scheduled as if as the schedule is in an equivalent linear network. On the other hand, the class of trees for which such policies do not exist is also large. Further, we showed that there are tree structures for which no sample-path optimal policy (even policies that are not necessarily causal) exists. This is a limitation of the sample-path metric, and hence this emphasizes the need to study other metrics for delay.

References

[1] V. J. Venkataramanan and X. Lin, "Low-complexity scheduling algorithm for sum-queue minimization in wireless convergecast," in *IEEE INFOCOM*, 2011.

[2] L. Tassiulas and A. Ephremides, "Dynamic scheduling for minimum delay in tandem and parallel constrained queue networks," *Annals of Operations Research*, 1993.

[3] T. Ji, E. Athanasopoulou, and R. Srikant, "Optimal scheduling policies in small generalized switches," in *IEEE INFOCOM*, 2009.

[4] G. R. Gupta and N. B. Shroff, "Delay analysis and optimality of scheduling policies for multi-hop wireless networks," *IEEE/ACM Transactions on Networking*, 2010.

[5] S. Hariharan and N. B. Shroff, "On optimal dynamic scheduling for sum-queue minimization in trees," in *IEEE WIOPT*, 2011.

[6] B. Miller and C. Bisdikian, *Bluetooth Revealed: The Insider's Guide to an Open Specification for Global Wireless Communications*. Prentice Hall, 2000.

[7] B. Hajek and G. Sasaki, "Link scheduling in polynomial time," *IEEE Transactions on Information Theory*, 1988.

[8] C. Florens, M. Franceschetti, and R. J. McEliece, "Lower bounds on data collection time in sensory networks," *IEEE Journal on Selected Areas in Communications*, 2004.

[9] L. Gargano and A. A. Rescigno, "Optimally fast data gathering in sensor networks," *Lecture Notes in Computer Science*, 2006.

[10] J. C. Bermond, L. Gargano, and A. A. Rescigno, "Gathering with minimum delay in tree sensor networks," *Lecture Notes in Computer Science*, 2008.

[11] S. Hariharan and N. B. Shroff, "On sample-path optimal dynamic scheduling for sum-queue minimization in trees under the k-hop interference model," http://www.ece.osu.edu/∼harihars/spreport.pdf, Tech. Rep., 2011.