

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.

Keywords: Wireless networks, Sample-path optimal scheduling, K -hop interference model.

I. INTRODUCTION

We investigate the problem of finding sample-path optimal scheduling policies for *convergecasting* [2] 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 significant importance in multi-hop wireless networks with a centralized node to which packets are sent. A number of applications in wireless networks involve convergecasting to one or more central nodes. For instance, it is of importance in sensor networks where the centralized node performs fusion of measurements received from multiple sensor nodes. It is also of importance in security applications where nodes need to send authentication information to a trusted central node. Yet another example is in centralized scheduling, where a leader node gathers control information from all the network nodes, and determines an

optimal schedule. In many such applications, a tree topology is used for transmitting packets from multiple nodes to the central node. Many of these applications as well as routing and data gathering protocols involve the construction of a tree, for example, a spanning tree. Delay performance is critical in many of these applications. A sample-path optimal scheduling policy is one for which the sum of the queue lengths of all the nodes in the network is minimum among all scheduling policies for each time slot, and for any traffic arrival pattern (a formal definition is provided in the Appendix). We are interested in minimizing the sum of the queue lengths of the nodes in the network as it can be shown to minimize the long term time average delay experienced by packets in the network.

We briefly overview the existing literature. Tassiulas et al., [3] 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., [4] develop a sample-path optimal policy for generalized switches with three links, and a heavy-traffic optimal policy for switches with four links. In [5], 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., [6] 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 [7] and FH-CDMA networks [8], while the 2-hop model is appropriate for IEEE 802.11. For the one-hop interference model, we showed in [6] that there are two classes of trees for which causal sample-path optimal policies exist, and that there are no other trees for which such policies exist. We have extended this work in [9] characterizing the classes of forests (multiple sinks) for which causal sample-path optimal policies exist under the one-

S. Hariharan is with AT&T Labs, San Ramon CA. Email: sh963r@att.com

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

This grant was partially supported by the Army Research Office through grants W911NF-14-1-0368 and MURI W911 NF-12-1-0385, National Science Foundation grants ECCS-1232118, CNS-1314538, and a grant from the Office of Naval Research N00014-15-1-2166.

A preliminary version of this work was presented at the IEEE International Conference on Computer Communications (INFOCOM), 2012 [1].

hop interference model. The challenge in extending to a K -hop interference model is that, for each $K > 1$, there are significantly more classes of trees in which such policies exist, and in which they do not. We provide an algorithm that takes any tree, and K as inputs, and outputs whether a causal sample-path optimal policy exists for the given tree under the K -hop interference model. We also prove that there are at most six classes of trees in which causal sample-path optimal policies exist, and that they do not exist in any other tree structure.

A number of authors have studied the convergecasting problem in the absence of arrivals (evacuation time optimality). Florens et al., [10] have studied the problem of minimizing the time by which all the packets in a network (with a tree topology) reach the sink, for a one-hop interference model. They propose polynomial time algorithms for this problem. Bermond et al., [11] and Gargano et al., [12] have further studied this problem for disk based communication model, and arbitrary network topologies respectively.

Hajek et al., [13] have investigated the problem of minimizing the time by which all flows reach a destination for general network topologies when each node has a constant traffic-arrival rate. Venkataramanan et al., [2] have studied the problem of minimizing the sum-queue length in convergecasting packets to the root of a network with a tree topology from a large-deviations perspective. Zhao et al., [14] have studied a similar convergecasting problem where each flow has a delay constraint. These works provide practical solutions to delay optimization in wireless networks. However, the optimality achieved is relatively weaker than sample-path optimality. Sample-path optimality is a very strong metric as it implies optimality with respect to many other metrics such as evacuation time optimality, large deviations optimality, etc. Apart from our earlier work on tree topologies for the one-hop interference model, it can be seen that sample-path optimal policies have only been shown to exist in very restricted network topologies. Therefore, by characterizing the existence of causal sample-path optimal scheduling policies in trees under a K -hop interference model, this work provides valuable insights into networks where sample-path optimality can be achieved, and those where we need to investigate relatively weaker optimality metrics.

Our contributions in this work are summarized below.

- 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 correctly classifies whether a causal sample-path optimal policy exists for the given tree under the K -hop interference model.
- We show that causal sample-path optimal policies only exist in six classes of trees. For each class, we show that the optimal scheduling policy is similar to scheduling in an appropriately constructed equivalent tandem network under a K -hop interference model. Further, we prove that for any other tree structure, there exists a traffic pattern such that no causal sample-path optimal scheduling policy can exist for the K -hop interference model.

The rest of this paper is organized as follows. In Section II, we describe the model and notations. In Section III, we construct the class of tree structures for which no causal sample-path optimal policy can exist under the K -hop interference model. Based on the intuition obtained from Section III, we develop an algorithm that classifies whether a given tree has a causal sample-path optimal policy under the K -hop interference model in Section IV. In Section V, we show that causal sample-path optimal policies exist for six classes of trees, and prove the correctness of the algorithm in Section IV. 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 [3], [10], [6], 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 . In any tree, for a given node r , we define m_1^r as the depth of the tree rooted at r , i.e., it is the length of the deepest branch of r . Suppose that $C(r)$ is the set of children of r , and that $r_d \in C(r)$ is the child of r in the deepest branch of r . We define m_2^r as $1 + \max_{\{i:i \in C(r), i \neq r_d\}} m_1^i$. m_2^r denotes the depth of r after removing r_d and the sub-tree rooted at r_d .

III. TREES WITH NO CAUSAL SAMPLE-PATH OPTIMAL POLICY

In this section, we construct the class of trees for which no causal sample-path optimal policy can exist under the K -hop interference model. We prove that no causal sample-path optimal policy can exist by generating appropriate traffic arrival patterns. Further, we also observe that, in some cases, even a non-causal sample-path optimal policy does not exist.

Theorem 1. *For a given tree, 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., $m_2^l > l$, there exists no causal sample-path optimal policy for the given tree structure under the K -hop interference model if either of the following conditions are true.*

- 1) $m_2^l \leq \lfloor \frac{K}{2} \rfloor$ and $m_1^l + m_2^l > K + 1$
- 2) $m_2^l > \lfloor \frac{K}{2} \rfloor$ and $m_1^l + m_2^l > K + 2$

Proof: We prove this result by contradiction.

Suppose that the result was not true, i.e., there exists a causal sample-path optimal scheduling policy for tree structures that satisfy either condition (1) or (2).

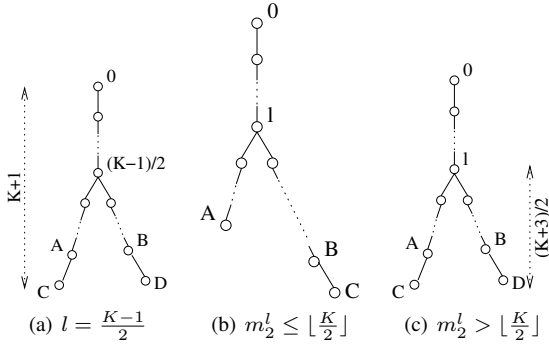


Fig. 1. No causal sample-path optimal policy

For a given l , $0 \leq l < \lfloor \frac{K}{2} \rfloor$, we consider the simplest tree structures corresponding to condition (1) and condition (2). We will now construct traffic arrival patterns to show that no causal sample-path optimal scheduling policy can exist for such tree structures, resulting in a contradiction.

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 1(b)), and with two links 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 $m_1^l + l - 1$. On the other hand, if we schedule node A , the time for the first packet to exit the system will only be $m_2^l + l < m_1^l + l - 1$. 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 $m_1^l = \frac{K+5}{2}$. By reusing the traffic arrival pattern in Case 1 and setting the number of packets to zero for the additional node at distance $\frac{K+1}{2}$ from l in its second deepest branch, 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 1(c) 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.

Note that the proof easily extends to tree structures for which $m_1^l + m_2^l > K + 2$ for condition (1), and $m_1^l + m_2^l > K + 3$ for condition (2). This is because we simply need to set the number of packets to zero for the additional nodes in the tree, and use the same traffic pattern shown above for the remaining nodes.

Hence, Theorem 1 follows. \blacksquare

Theorem 2. *Let K be an odd number. Let l be the node at distance $\frac{K-1}{2}$ from the sink 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 $\frac{K+1}{2}$, i.e., $m_2^l > \frac{K+1}{2}$, then there exists no causal sample-path optimal policy for this tree structure.*

Proof: We prove this result by contradiction.

Suppose that the result was not true. We consider the simplest tree structure for which $m_2^l > \frac{K+1}{2}$ at node $l = \frac{K-1}{2}$ (Figure 1(a)). In this tree, $m_1^l = K + 1$, $m_1^l = \frac{K+3}{2}$, and $m_2^l = \frac{K+3}{2} > \frac{K+1}{2}$.

Consider the following traffic arrival pattern. At time $t = 0$, there exists one packet each at nodes A and B which are both at depth $\frac{K+1}{2}$ from node l . Note that due to the K -hop interference model, A and B cannot be scheduled simultaneously

since they are separated by only $K - 1$ links. Without having future knowledge of traffic arrivals, we can either choose to schedule A or B in the first time slot. Suppose we schedule A , and a packet arrives at node D at time $t = 1$. Also, assume that there are no further packet arrivals in the system. It can be easily seen that only one packet can be scheduled during any time slot in the network due to the K -hop interference model. Hence, from slot $t = 1$, the total time for the three packets to reach node 0 is $(K - 1) + K + (K + 1) = 3K$. On the other hand, if we had known that a packet was going to arrive at D at time $t = 1$, we could have scheduled B during the first time slot. Note that A and D are separated by K links, and can hence simultaneously transmit. Therefore, from slot $t = 1$, the total time for the three packets to reach node 0 in this case is $(K - 1) + K + K = 3K - 1 < 3K$. This contradicts our assumption that the result does not hold.

Thus, this traffic arrival pattern shows that without having future knowledge of arrivals, there can exist no sample-path optimal policy for this tree structure. As argued in the proof of Theorem 1, this immediately implies that there can exist no causal sample-path optimal policy in any tree that contains this structure. ■

As a result of Theorems 1 and 2, we have a class of tree structures in which no causal sample-path optimal policy can exist under the K -hop interference model. In Section V, we will show that we can find a causal sample-path optimal policy for every other tree structure under this interference model.

We can now visualize an algorithm that takes any tree and K as inputs, and outputs whether a causal sample-path optimal scheduling policy exists for the given tree under the K -hop interference model. We develop and discuss this algorithm in the next section.

IV. CLASSIFICATION ALGORITHM

Based on the results in Section III, we develop an algorithm that identifies whether a causal sample-path optimal policy exists for a given tree under the K -hop interference model.

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 $m_1^l - l$. 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 2 illustrates two examples explaining the functioning of A_{sp} for the 3-hop interference model. Consider Figure 2(a).

```

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
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
end

```

TABLE I
ALGORITHM A_{sp}

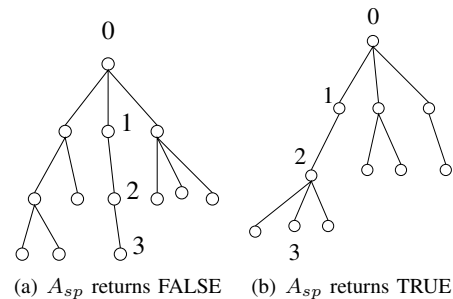


Fig. 2. Examples illustrating the functioning of A_{sp} for $K = 3$

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 Figure 2(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 3. *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, \dots, 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 if there is 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. The implication of this is that we can hope to convert the tree into an equivalent line network [10], [6], and schedule the tree as though the schedule is in a line network. This can be reasoned as follows. Suppose that we have a node A at distance $a \leq l$ in one branch, and another node B at distance $K + 2 - a$ from node l in the other branch. Note that these nodes can potentially transmit simultaneously according to the K -hop interference model. However, their parents cannot transmit simultaneously. Suppose that A and B each had a packet. If they transmit simultaneously, the packets will reach the respective parents of A and B , and after this slot, they cannot be simultaneously transmitted. Therefore, if we keep transmitting the packet from node A , the packet from node B cannot transmit until the former packet reaches the sink. Even if we had initially not transmitted simultaneously and just transmitted the packet from A , since A is at a distance $a + l$ from the sink, and B is at a distance $K + 2 - a + l$, when the packet from node A reaches the sink's child, the distance between the sink's child and B is at least $K + 1$, and hence the sink's child and B can simultaneously transmit. Therefore, at the slot the packet from A reaches the sink, the packet from B will be at B 's parent, which is the same situation as before. The implication of this is that we can hope to convert the tree into an equivalent line network ([10], [6]), 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.

We now show that Algorithm A_{sp} correctly identifies tree structures for which no causal sample-path optimal policy can exist under the K -hop interference model.

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 1 and 2. Theorem 1 proves it for $0 \leq l < \lfloor \frac{K}{2} \rfloor$, and Theorem 2 shows the result for $l = \lfloor \frac{K}{2} \rfloor$. ■

As a sanity check, we can also verify our results for the 1-hop interference model by comparing it with the results in [6]. For the 1-hop interference model, $\lfloor \frac{1}{2} \rfloor = 0$. Hence, we only need to apply Theorem 2. This says that if $m_2^0 > 1$, there exists no causal sample-path optimal policy for the given tree structure. Theorem 3 in [6] shows that there exists no causal sample-path optimal policy for a tree structure where $m_1^0 = m_2^0 = 2$, thus verifying our result.

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

V. EXISTENCE OF CAUSAL SAMPLE-PATH OPTIMAL SCHEDULING POLICIES

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.

A. Classification of Trees

We 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, the depth of the tree must be bounded by K , i.e., $m_1^0 \leq K$, unless 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 any arbitrary quantity 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 showing that m_1^0 cannot be larger than K if either of the conditions are not satisfied for a tree for which A_{sp} returns true.

We first consider the case where K is odd. Suppose that for some l such that $0 \leq l < \frac{K-1}{2}$, $m_2^l > l$, and $m_1^0 > K$. We have the following possibilities.

Case 1: 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.

Consider the case where K is even. Suppose that $m_1^0 > K$, and for some l such that $0 \leq l < \frac{K}{2}$, $m_2^l > l$. We have the following possibilities.

Case 1: $m_2^l \leq \frac{K}{2}$: Again, $m_1^l = m_1^0 - l > K - l$. Therefore, following the same argument as in the previous case, $m_1^l + m_2^l \geq K + 1 - l + l + 1 = K + 2$. Hence, *sp* will return FALSE, and we similarly get a contradiction.

Case 2: $m_2^l > \frac{K}{2}$: Since $l < \frac{K}{2}$, $l \leq \frac{K}{2} - 1$. Hence, $m_1^l = m_1^0 - l > \frac{K}{2} + 1$. So, $m_1^l \geq \frac{K}{2} + 2$ and $m_2^l \geq \frac{K}{2} + 1$. Thus, $m_1^l + m_2^l \geq K + 3 > K + 2$. Therefore, subroutine *sp* will return FALSE, A_{sp} will consequently return FALSE, and we obtain 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 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.*

Based on the above theorem and corollary, we now classify trees for which A_{sp} returns TRUE into six classes. The first three classes correspond to the scenario in which the depth of the tree must be bounded by K , and the last three classes correspond to the case in which the depth of the tree need not be bounded by K . Figure 3 shows a flow-chart describing this classification.

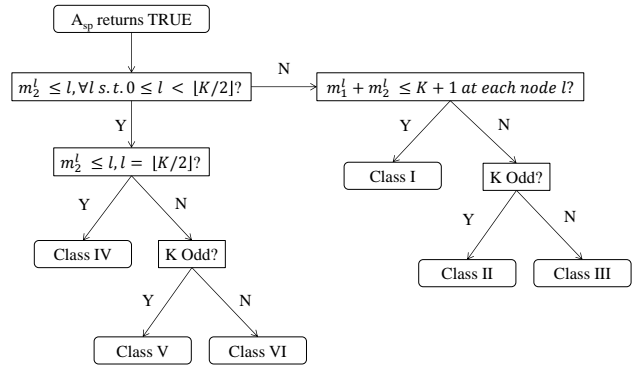


Fig. 3. Classification of trees for which A_{sp} returns TRUE

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. It can be shown that this implies that 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, $m_1^l + m_2^l \leq K + 1$.

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, and we discuss this in more detail when we provide the sample-path optimal policies for these classes. It can be shown that this condition implies that 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, $m_1^l + m_2^l \leq K + 1$.

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 $l = \lfloor \frac{K}{2} \rfloor$, $m_2^l \leq l$. We show that these trees can be scheduled using a schedule in an *equivalent linear network* [6], [10].

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 $l = \frac{K-1}{2}$, $m_2^l = \frac{K+1}{2}$.

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. ■

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 9(a) provides an example of Class I trees under the 2-hop interference model. We define the causal policy, π_I , for Class I trees as follows.

Policy π_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, $m_1^l + m_2^l \leq K + 1$.

Note: The node l in a line in the deepest branch of the tree for which $m_1^l + m_2^l = K + 2$ and $m_2^l = \frac{K+1}{2}$ is unique. It corresponds to the node in the line for which $m_1^l = \frac{K+3}{2}$.

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 4(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_{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*

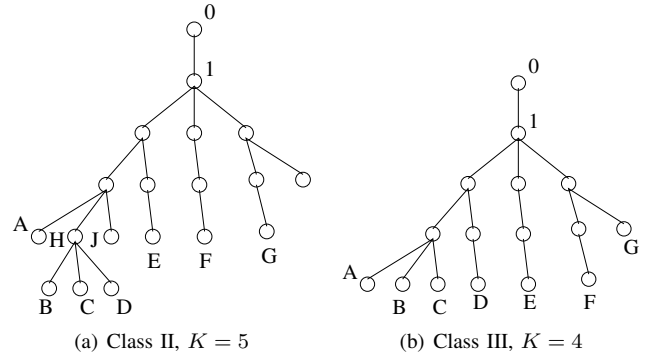


Fig. 4. Examples for Classes II and III

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 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 4(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, $m_1^l + m_2^l \leq K + 1$. As we argued in the previous section for Class II trees, at most one node l in the line can have $m_1^l + m_2^l = K + 2$.

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 , $i = 1, 2, \dots, p$, 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 $a_1 \in N_1$, $a_2 \in N_2, \dots$, $a_p \in N_p$. a_1, a_2, \dots, a_p can all transmit during the same slot since any two nodes in the set $\{a_1, a_2, \dots, a_p\}$ 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 4(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_{III} for this class of trees.

Policy π_{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 $N_1 \cup N_2 \cup \dots \cup N_p$. 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 $N_1 \cup N_2 \cup \dots \cup N_p$, select one packet in each of N_1, N_2, \dots, N_p (as long as a packet exists), and schedule these packets simultaneously during that slot.*

In Figure 4(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 7(b) and 10(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 [10].

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: $V_l = \{0, 1, \dots, N\}$, $E_l = \{(i-1, i), 1 \leq i \leq N\}$ 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 5 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.

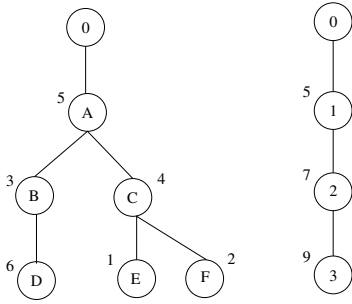


Fig. 5. Equivalent Linear Network

We now propose policy π_{IV} for Class IV trees.

Policy π_{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 $1, 2, \dots, i-1$ 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 $\{i-1, i-2, \dots, i-K\}$ 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 [3] 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 [6]).

Remark 4: According to policy π_{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_{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 5, we can arbitrarily choose to schedule one of $\{D, E, F\}$ according to π_{IV} . We can potentially simultaneously schedule D and E . However, this policy does not allow such a schedule because in the equivalent linear network, R can at most send one packet in a slot.

Remark 6: If a node i in the equivalent linear network is selected to schedule during a certain slot according to π_{IV} , none of the nodes that are $i-1$ hops away from node 0 in the original tree can transmit. Since it is possible to potentially schedule a node that is at distance $i-1$ and a node at distance i simultaneously without interference as long as the node at distance i is not a child of the node at distance $i-1$, it is interesting that even without scheduling such non-interfering links, this policy is optimal. For example, in Figure 5, we can potentially simultaneously schedule B and E . However, this policy does not allow such a schedule because in the equivalent linear network, when R makes a transmission, Q cannot make a transmission.

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_V for this class of trees.

Policy π_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 6(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.

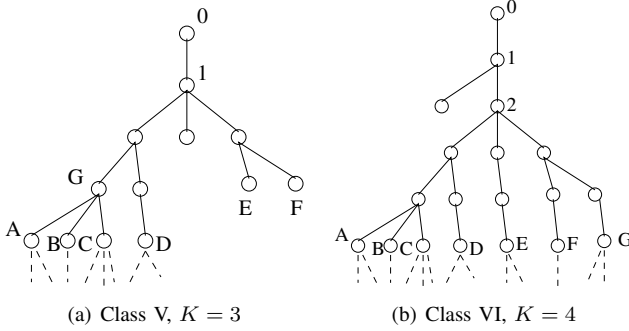


Fig. 6. Examples for Classes V and VI

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 , $i = 1, 2, \dots, p$.

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

Policy π_{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 $N_1 \cup N_2 \cup \dots \cup N_p$. For N_i , $i = 1, 2, \dots, p$, 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 6(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_i for Class i trees, $i = I, II, \dots, VI$.

Theorem 7. For Class i of trees, $i = I, II, \dots, VI$, policy π_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: 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. We prove this result for Class IV trees in the appendix. The proof for the other tree classes are of the same flavor. Due to space limitations, we refer the readers to [15] for the other tree classes. ■

Remark 7: 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 8: Theorem 7 states that the number of packets in the network under π_i for Class i is smaller than the number of packets under any other policy at all time instants. Therefore, the long term time average number of packets in the system under π_i is smaller than the corresponding number under any other policy. From Little's law, the long term time average delay is directly proportional to the average number of packets in the system. Hence, π_i minimizes the average delay in the system for Class i trees, $i = I, II, \dots, VI$.

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 convergencing problem.

VI. EXAMPLES - 1-HOP AND 2-HOP

In this section, we apply the results in Sections III 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 [6] 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 2, 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 7(a) has no causal sample-path optimal policy. This verifies Theorem 3 in [6].

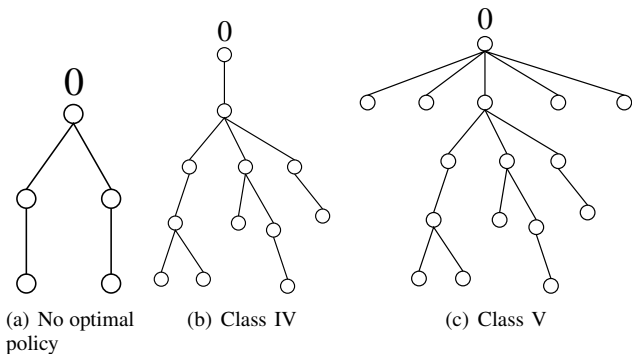


Fig. 7. Existence of sample-path optimal policies for $K = 1$

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 $0 \leq l \leq \frac{K-1}{2}$, $m_2^l \leq l$. This means that at the root ($l = 0$), $m_2^0 \leq 0$. 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 [6]. Finally, for Class V trees, at node $l = \frac{K-1}{2} = 0$, $m_2^0 \leq 1$. 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 [6], we can verify the correctness of both the optimal policy, and the structure of this class of trees. Figures 7(b) and 7(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 1, $m_2^0 > 0$. $sp(m_1^0, m_2^0, 2)$ 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 8 shows an example of such tree structures.

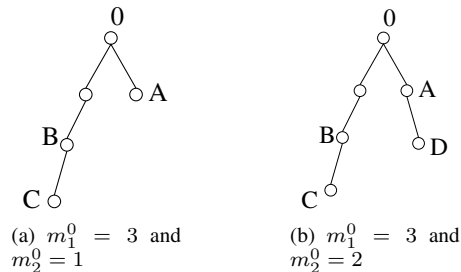


Fig. 8. No causal sample-path optimal policy, $K = 2$

Indeed, if $m_1^0 = 3$ and $m_2^0 = 1$ (Figure 8(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 8(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 9(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 $m_1^0 = 2$, so that $m_1^0 + m_2^0 = K + 2 = 4$. Figure 9(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.

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 10(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 10(a) will not transmit simultaneously even though they can potentially do so.

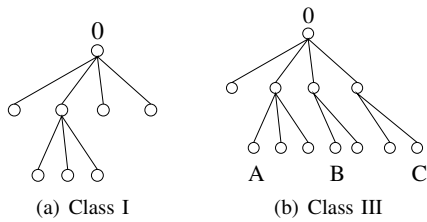


Fig. 9. Examples of Class I and Class III trees for $K = 2$

Class VI: For this class, we only have the condition that $m_2^0 = 0$. m_2^1 can be arbitrary. Figure 10(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 10(b) must transmit simultaneously if they both have packets to transmit.

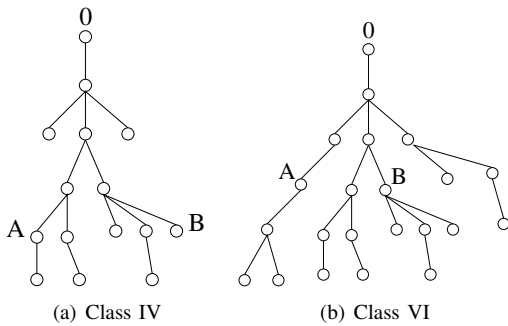


Fig. 10. Examples of Class IV and Class VI trees for $K = 2$

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

APPENDIX

We recall the definition of a sample-path traffic arrival pattern and a causal sample-path optimal scheduling policy for a wireless networks as defined in [3], [4], [5].

Sample-path traffic arrival: Let $\mathbf{A}(t)|t \in \{0, 1, 2, \dots\}$ be a stochastic process, where $\mathbf{A}(t)$ is a random vector (representing traffic arrivals at nodes in the given network) on the probability space (Ω, \mathcal{F}, P) . For any fixed sample point $\omega \in \Omega$, the function $\mathbf{A}_\omega(t) : t \rightarrow \mathbf{A}(t)$ is called a sample-path of the stochastic process. In other words, considering traffic arrivals as a stochastic process, any sample traffic arrival pattern constitutes a sample-path of the stochastic process.

Sample-path optimal scheduling policy: A sample-path optimal scheduling policy for a wireless network is one for which at each time slot, and for any sample-path traffic arrival pattern, the sum of the queue lengths of all the nodes in the network is minimum among all policies. Further, a causal sample-path optimal scheduling policy is a sample-path optimal scheduling policy that is also causal, i.e., the scheduling decision at any given time slot is independent of future traffic arrivals.

We first provide some notations, definitions, and results that will be used in the sample-path optimality proofs of all the classes of trees. Most of these notations and definitions are similar to that defined in [3], and in [6] to prove sample-path optimality in tree structures under the one-hop interference model. Our proofs use some of the basic structure used in the sample-path optimality proof for tandem networks in [3]. The reason is that the structure of the proofs developed in [3] neither depends on the topology nor on the interference model. Following the same structure, we first identify a relationship between the location of a packet, and the time for it to reach the sink. We then prove optimality in the absence of arrivals, and finally prove optimality in the presence of arrivals. It is important to note that while the structure is similar, with a different topology as well as a different interference model, the details of the proof are quite different.

- 1) Activation Set: A set of links that can be simultaneously activated such that no two links interfere with each other according to the K -hop interference model.
- 2) Activation Vector: A binary indicator vector i with one element for each link (which is not zero if and only if the link belongs to the activation set).
- 3) S : Set of all possible activation vectors.
- 4) $A_i(t)$: Set of exogenous packet arrivals to node i at slot t .
- 5) $\mathbf{A}(t)$: Vector of arrivals at all nodes during slot t .
- 6) $X_i(t)$: Length of the queue of packets at node i by the end of slot t . $X_i(t) \geq 0 \forall i$.
- 7) $\mathbf{X}(t)$: $\mathbf{X}(t) = (X_i(t), i = 1, \dots, N)$ is the vector of queue lengths at all nodes at the end of slot t .
- 8) \mathbf{X} : The queue length process $\{\mathbf{X}(t)\}_{t=1}^\infty$.
- 9) $\mathbf{I}(t)$: Indicator vector denoting the set of links activated at time slot t . A link is activated only if the corresponding node has packets to send.
- 10) π_i : The stationary policy that schedules link activations at each time slot for Class i trees, $i = I, II, III, IV, V, VI$.
- 11) $g_i(\mathbf{X}(t))$: The activation vector corresponding to policy π_i for Class i trees, $i = I, II, III, IV, V, VI$.

For the convergencing problem, the queue length vector

evolves as $\mathbf{X}(t+1) = \mathbf{X}(t) + \mathbf{R}\mathbf{I}(t+1) + \mathbf{A}(t+1)$, where \mathbf{R} is an $N \times N$ matrix with elements

$$r_{ij} = \begin{cases} 1, & j \text{ is a child of } i \\ -1, & i = j \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

Definition: Let X, Y be the queue length processes when the initial queue length vectors are $\mathbf{X}(0) = \mathbf{x}$, $\mathbf{Y}(0) = \mathbf{y}$ respectively, there are no exogenous arrivals, and policy π_i schedules link activations for Class i trees. We say that the vectors \mathbf{x} and \mathbf{y} are related with the partial ordering \prec and we write $\mathbf{x} \prec \mathbf{y}$ if for all $t = 0, 1, \dots$, we have $s(\mathbf{X}(t)) \leq s(\mathbf{Y}(t))$, where $s(\mathbf{x}) = \sum_{i \in V} x_i$ is the total number of packets in the system when the state is \mathbf{x} .

To each state \mathbf{x} we define the *departure times* $t_i^{\mathbf{x}}$, $i = 1, \dots, s(\mathbf{x})$ and the *positions* $d_i^{\mathbf{x}}$, $i = 1, \dots, s(\mathbf{x})$ as follows.

Definition: Assume that the system is initially in state \mathbf{x} , there are no exogenous arrivals, and policy π_j schedules link activations for the given Class j tree. Let $\{\mathbf{X}(t)\}_{t=1}^{\infty}$ be the corresponding queue length process. The time $t_i^{\mathbf{x}}$ is defined as

$$t_i^{\mathbf{x}} = \min\{t : t > 0, s(\mathbf{X}(t)) \leq s(\mathbf{x}) - i\}, \quad i = 1, \dots, s(\mathbf{x}), \quad (2)$$

and the position $d_i^{\mathbf{x}}$ is defined as the distance (number of hops) from the sink at which the i^{th} packet to exit the system lies.

Since these definitions were previously defined for tandem networks in [3], we provide an example to show how they extend to a Class IV tree under a 1-hop interference model. Consider the Class IV tree in Figure 11 with sink 0. Suppose that in state \mathbf{x} , node A has two packets, node B has one packet, and node C has three packets. In the equivalent linear network, node 1 would have two packets, and node 2 would have four packets. According to the definitions above, $d_i^{\mathbf{x}} = 1$ for $i \in \{1, 2\}$, and $d_i^{\mathbf{x}} = 2$ for $i \in \{3, 4, 5, 6\}$. Also, we have $t_1^{\mathbf{x}} = 1$, $t_2^{\mathbf{x}} = 2$, $t_3^{\mathbf{x}} = 4$, $t_4^{\mathbf{x}} = 4$, $t_5^{\mathbf{x}} = 8$, and $t_6^{\mathbf{x}} = 10$. Irrespective of the way we order and schedule the packets in nodes B and C , $d_i^{\mathbf{x}}$, and $t_i^{\mathbf{x}}$ will remain the same for state \mathbf{x} for $i \in \{1, 2, 3, 4, 5, 6\}$. Ordering the packets in B and C is equivalent to ordering the packets in node 2 in the equivalent linear network. For simplicity, we can order the packets from the left-most node to the right-most node among nodes equidistant from the sink in order to obtain a unique index and schedule for each packet. So the definitions of $d_i^{\mathbf{x}}$ and $t_i^{\mathbf{x}}$ extend to the six classes of trees defined in this paper.

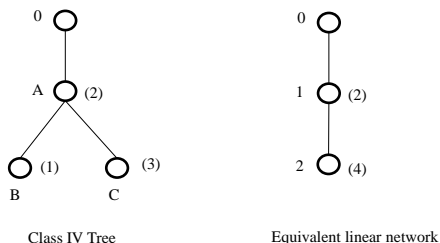


Fig. 11. Example of Class IV tree and its corresponding equivalent linear network

We now recall Lemma 3.2 in [3].

Lemma 1. For any two vectors \mathbf{x} and \mathbf{y} , we have $\mathbf{x} \prec \mathbf{y}$ if and only if

$$t_i^{\mathbf{x}} \leq t_{i+k}^{\mathbf{y}}, \quad i = 1, \dots, s(\mathbf{x}), \quad (3)$$

where $k = s(\mathbf{y}) - s(\mathbf{x})$.

This lemma states that $\mathbf{x} \prec \mathbf{y}$ if and only if, for any i , the time by which the i^{th} packet in state \mathbf{x} leaves the system is no greater than the time by which the $(i+k)^{\text{th}}$ packet in state \mathbf{y} leaves the system. The proof of this lemma can be found in [3].

We now prove the optimality of policy π_{IV} for Class IV trees. We first develop a relationship between $t_i^{\mathbf{x}}$ and $d_i^{\mathbf{x}}$ when the network is scheduled according to policy π_{IV} at each time slot and there are no packet arrivals in the system.

Lemma 2. For Class IV trees, for all states \mathbf{x} , $t_i^{\mathbf{x}}$ is defined as follows.

$$t_i^{\mathbf{x}} = \begin{cases} d_i^{\mathbf{x}} & i = 1 \\ i & d_i^{\mathbf{x}} = 1 \\ t_{i-1}^{\mathbf{x}} + d_i^{\mathbf{x}} & i > 1, 2 \leq d_i^{\mathbf{x}} \leq K \\ \max(d_i^{\mathbf{x}}, t_{i-1}^{\mathbf{x}} + K + 1) & i > 1, d_i^{\mathbf{x}} > K \end{cases} \quad (4)$$

Proof: Consider the system operated under policy π_{IV} , with initial state \mathbf{x} and there are no arrivals in the system. Since the closest packet to the sink gets priority, the first packet gets forwarded to the sink at each slot. Hence, $t_1^{\mathbf{x}} = d_1^{\mathbf{x}}$. If $d_i^{\mathbf{x}} = 1$, there are at least i packets at distance 1 to the sink. Therefore, the i^{th} packet will reach the destination at the end of slot i . If $i > 1$ and $d_i^{\mathbf{x}} > 1$ and $d_i^{\mathbf{x}} \leq K$, the i^{th} packet is located at a node that is $d_i^{\mathbf{x}}$ hops away from the sink. Since no two links can simultaneously transmit in the system, at the slot at which the $(i-1)^{\text{th}}$ packet reaches the sink, the i^{th} packet will still remain at the same node. Therefore, $d_i^{\mathbf{x}}$ slots after $t_{i-1}^{\mathbf{x}}$, the i^{th} packet will reach the sink.

Suppose that $i > 1$ and $d_i^{\mathbf{x}} > K$. The proof for this case is similar to that of Lemma 3.1 in [3]. However, since our proof is for the K -hop interference model, and Lemma 3.1 only considers the 1-hop interference model, we provide the details below. Consider the following cases.

Case 1: $d_i^{\mathbf{x}} - t_{i-1}^{\mathbf{x}} \geq K + 1$.

At any slot $t < t_{i-1}^{\mathbf{x}}$, the packet $i-1$ should reside in a node j in the original tree such that $d(0, j) \leq t_{i-1}^{\mathbf{x}} - t$ because it should reach the destination in $t_{i-1}^{\mathbf{x}} - t$ slots, and cannot be forwarded faster than one hop during each slot. Also, at time t , the packet i should reside in a node m such that $d(0, m) \geq d_i^{\mathbf{x}} - t$ since it cannot move faster towards the destination than one hop per slot. Therefore we have $d(0, m) \geq d_i^{\mathbf{x}} - t \geq t_{i-1}^{\mathbf{x}} - t + K + 1 \geq d(0, j) + K + 1$. This implies that packet $i-1$ will be, at each slot t , at least $K+1$ nodes closer to the destination than packet i in both the original tree as well as the equivalent linear network. Therefore packet i will be the first packet in its queue, and all the nodes in the tree that are one hop closer to the destination than the node at which packet i currently is have no packets in their respective queues. Therefore, packet i will be forwarded by one node towards the destination at each slot. Hence, packet i will reach the destination by the end of slot $d_i^{\mathbf{x}}$, i.e., $t_i^{\mathbf{x}} = d_i^{\mathbf{x}}$.

Case 2: $d_i^x - t_{i-1}^x \leq K$.

If $i > 1$, $d_i^x > K$, then $t_i^x \geq t_{i-1}^x + K + 1$. This is because any packet which is not residing in a node at depth $\leq K$ from the sink at $t = 0$, can reach one of these nodes only when there are no packets left to schedule in any of these nodes, since these nodes would be activated otherwise (as they are closer to the root of the tree). This is only true because the tree is a Class IV tree. This is not true for Class V and Class VI trees. Hence, during the slot at which $i-1$ leaves the system, packet i will be in node $K+1$ in the equivalent linear network (corresponding to one of the nodes at depth $K+1$ in the original tree) or further away from the destination, and therefore it requires at least $K+1$ additional slots in order to reach the destination.

We now show that $t_i^x = t_{i-1}^x + K + 1$. If packet i is forwarded towards the destination by one node at each slot then it will reach the destination by slot d_i^x . However, this is impossible since $d_i^x - t_{i-1}^x \leq K$, and we need $t_i^x \geq t_{i-1}^x + K + 1$. This means that at some slot, packet i is not forwarded from its node (say node k). Suppose that packet $i-1$ was residing at node j during this slot. Then we must have $d(0, j) \leq d(0, k) + K$, i.e., in the equivalent linear network packet $i-1$ is either in the same node with i or in a node that is at most K hops in front of i towards the destination. Therefore, at the slot at which $i-1$ was not forwarded and at all subsequent slots until the time packet $i-1$ leaves the system, packets i and $i-1$ cannot be in two nodes m, n such that $d(0, m) - d(0, n) > K + 1$. Therefore, $K + 1$ slots after the time packet $i-1$ reaches node 0, packet i also reaches node 0. Thus, $t_i^x = t_{i-1}^x + K + 1$. ■

We now show that the partial ordering defined earlier propagates in time if there are no exogenous arrivals in the network. Specifically, we show that if we deviate from the optimal scheduling policy during any given slot, and follow the optimal policy for all the following slots until all packets exit the system, the partial ordering is preserved.

Lemma 3. *If we have $\mathbf{x} \prec \mathbf{y}$, and \mathbf{i} is an arbitrary activation vector, then for $\mathbf{u} = \mathbf{x} + \mathbf{R}g_{IV}(\mathbf{x})$ and $\mathbf{z} = \mathbf{y} + \mathbf{R}\mathbf{i}$, we have $\mathbf{u} \prec \mathbf{z}$.*

Proof: We show that for all $i = 1, \dots, s(\mathbf{u})$, we have $t_i^u \leq t_{i+s(\mathbf{z})-s(\mathbf{u})}^z$. Hence, from Lemma 1, we can conclude that $\mathbf{u} \prec \mathbf{z}$. In order to prove this result, we show that the following relations hold.

- $s(\mathbf{u}) = s(\mathbf{x}) \Rightarrow t_i^u = t_i^x - 1$
- $s(\mathbf{u}) = s(\mathbf{x}) - 1 \Rightarrow t_i^u = t_{i+1}^x - 1$
- $s(\mathbf{z}) = s(\mathbf{y}) \Rightarrow t_i^z \geq t_i^y - 1$
- $s(\mathbf{z}) = s(\mathbf{y}) - 1 \Rightarrow t_i^z = t_{i+1}^y - 1$

Let $s(\mathbf{y}) - s(\mathbf{x}) = k$. We consider the following cases.

Case 1: $s(\mathbf{u}) = s(\mathbf{x})$ and $s(\mathbf{z}) = s(\mathbf{y})$.

In this case, we need to show that $t_i^u \leq t_{i+k}^z$, $\forall i = 1, \dots, s(\mathbf{u})$.

Since \mathbf{u} results from applying policy π_{IV} , from the definition of departure times, it immediately follows that

$$t_i^u = t_i^x - 1. \quad (5)$$

We now show that for any packet i , $t_i^z \geq t_i^y - 1$ by induction.

For $i = 1$, $t_1^z = d_1^z$, and $t_1^y = d_1^y$. Further, $d_1^z \geq d_1^y - 1$ since a packet can at most go one hop closer to the sink during a

slot. Hence, $t_1^z \geq d_1^y - 1 = t_1^y - 1$.

Assume that the result is true for some packet $i \geq 1$.

Consider packet $i + 1$.

If $d_{i+1}^z = 1$, then $t_{i+1}^z = i + 1$. Also, $d_{i+1}^z = 1$ implies that $d_{i+1}^y = 1$ or $d_{i+1}^y = 2$. If $d_{i+1}^y = 1$, $t_{i+1}^y = i + 1$. On the other hand, if $d_{i+1}^y = 2$, then there are i packets that are one-hop away from the sink (since $d_{i+1}^z = 1$). Therefore, in this case, $t_{i+1}^y = i + 2$. Thus, in either case, $t_{i+1}^z \geq t_{i+1}^y - 1$.

If $2 \leq d_{i+1}^z \leq K$, $t_{i+1}^z = t_i^z + d_{i+1}^z$. If $d_{i+1}^z = d_{i+1}^y$, then this packet was not scheduled. Hence, $t_{i+1}^y = t_i^y + d_{i+1}^y$. Since $t_i^z \geq t_i^y - 1$ by the induction hypothesis, it follows that $t_{i+1}^z \geq t_{i+1}^y - 1$. On the other hand, if $d_{i+1}^z = d_{i+1}^y - 1$, then no other packet closer to the sink could have been scheduled because of the K -hop interference model. Hence, $t_i^z = t_i^y$. Hence, we obtain $t_{i+1}^z = t_{i+1}^y - 1$.

If $d_{i+1}^z > K$, then $t_{i+1}^z = \max(d_{i+1}^z, t_i^z + K + 1)$. Since $d_{i+1}^z \geq d_{i+1}^y - 1$, and by the induction hypothesis, we have $t_{i+1}^z - 1 = \max(d_{i+1}^y, t_i^y + K + 1) - 1 \leq \max(d_{i+1}^z, t_i^z + K + 1) = t_{i+1}^z$.

Hence, the result holds by induction for any packet i .

Since $t_i^u = t_i^x - 1 \leq t_{i+k}^y - 1 \leq t_{i+k}^z$, it follows that $\mathbf{u} \prec \mathbf{z}$ in this case.

Case 2: $s(\mathbf{u}) = s(\mathbf{x}) - 1$, $s(\mathbf{z}) = s(\mathbf{y})$.

In this case, we need to show that $t_i^u \leq t_{i+k+1}^z$, $\forall i = 1, \dots, s(\mathbf{u})$.

Since one packet exits the system according to policy π_{IV} , the $(i+1)^{th}$ packet in the previous slot now becomes the i^{th} packet. Therefore, $t_i^u = t_{i+1}^x - 1$.

For state \mathbf{z} , the situation is identical to that of Case 1. Therefore, $t_i^z \geq t_i^y - 1$. Therefore, it follows that $t_i^u \leq t_{i+k+1}^z$, $\forall i = 1, \dots, s(\mathbf{u})$.

Case 3: $s(\mathbf{u}) = s(\mathbf{x}) - 1$, $s(\mathbf{z}) = s(\mathbf{y}) - 1$.

In this case, we need to show that $t_i^u \leq t_{i+k}^z$, $\forall i = 1, \dots, s(\mathbf{u})$.

From Case 2 for \mathbf{u} , we have $t_i^u = t_{i+1}^x - 1$.

For state \mathbf{z} , we now show by induction that $t_i^z \geq t_{i+1}^y - 1$.

$i = 1$: We have $t_1^z = d_1^z \leq d_2^y \leq t_2^y$. If $t_2^y = d_2^y$, then $t_1^z = d_1^z \geq d_2^y - 1 = t_2^y - 1$. Therefore, the result holds in this case. On the other hand, if $t_2^y > d_2^y$, then the second packet was at a node at distance $\leq K + 1$ from the sink at the previous slot (otherwise, $t_2^y = d_2^y$). Since the first packet was scheduled, i.e., the closest packet to the sink was scheduled, the second packet in state \mathbf{y} cannot be scheduled, and the time for all packets in the system to reach the sink decreases by 1. In this case, $t_1^z = t_2^y - 1$. Therefore, $t_1^z \geq t_2^y - 1$.

Assume that it holds for some i by the induction hypothesis.

$i + 1$: If $d_{i+1}^z = 1$, then since a packet in node 1 was scheduled in the previous slot, $d_{i+2}^y = 1$. Hence, $t_{i+1}^z = i + 1$ and $t_{i+2}^y = i + 2$. Therefore, $t_{i+1}^z = t_{i+2}^y - 1$.

If $2 \leq d_{i+1}^z \leq K$, since node 1 was scheduled, this packet could not have been scheduled in the previous slot. However, since the closest packet was scheduled, the time for this packet to reach the sink decreases by 1. Therefore, $t_{i+1}^z = t_{i+2}^y - 1$.

If $d_{i+1}^z > K$, then $d_{i+1}^z \geq d_{i+2}^y - 1$, since it moves one hop closer to the sink if scheduled, or stays at the same distance, otherwise. Hence, by this relation and the induction

hypothesis, we have $t_{i+2}^y - 1 = \max(d_{i+2}^y, t_{i+1}^y + K + 1) - 1 \leq \max(d_{i+1}^z, t_i^z + K + 1) = t_{i+1}^z$.

Thus, by induction, $t_i^z \geq t_{i+1}^y - 1$ for all packets i .

Therefore, $t_{i+k}^z \geq t_{i+k+1}^y - 1 \geq t_{i+1}^x - 1 = t_i^u$.

Hence, it follows that $t_i^u \leq t_{i+k}^z$ for this case.

Case 4: $s(\mathbf{u}) = s(\mathbf{x})$, $s(\mathbf{z}) = s(\mathbf{y}) - 1$.

In this case, we need to show that $\forall i, t_i^u \leq t_{i+k-1}^z$.

The case for \mathbf{u} is identical to that in Case 1, and the case for \mathbf{z} is identical to that in Case 3. Therefore, $t_i^u = t_i^x - 1$, and $t_{i+k-1}^z \geq t_{i+k}^y - 1$. Since $t_{i+k}^y - 1 \geq t_i^x - 1$, it follows that $t_i^u \leq t_{i+k-1}^z$. Thus, we have shown that $\mathbf{u} \prec \mathbf{z}$. ■

We now show that the ordering \prec between two states is preserved after a packet arrives at any network node. To be precise, let \mathbf{e}_j be the vector which has all its elements equal to zero except for the element j which is 1. Then we have the following.

Lemma 4. *If we have $\mathbf{x} \prec \mathbf{y}$, then for all $j \in V$, we also have $\mathbf{x} + \mathbf{e}_j \prec \mathbf{y} + \mathbf{e}_j$.*

Proof: Due to space limitations, we have provided the proof as supplementary material (downloadable from <http://ieeexplore.ieee.org>). ■

We now prove Theorem 7 for Class IV trees. We note that the proof is identical to Theorem 3.1 in [3]. For the reader's convenience, we repeat the proof.

Proof of Theorem 7 for Class IV Trees:

Proof: For $t = 0$, we have $\mathbf{X}^I(0) = \mathbf{X}(0)$, and hence $\mathbf{X}^I(t) \prec \mathbf{X}(t)$ at $t = 0$. Assume that $\mathbf{X}^I(t) \prec \mathbf{X}(t)$ is true for some t . We show that it holds for $t + 1$ as well. Let $\mathbf{I}(t + 1)$ be the activation vector under some policy π at $t + 1$. Then from Lemma 3 we have

$$(\mathbf{X}^I(t) + \mathbf{R}g_I(\mathbf{X}^I(t))) \prec \mathbf{X}(t) + \mathbf{R}\mathbf{I}(t + 1). \quad (6)$$

Further, the arrival vector $\mathbf{A}(t + 1)$ can be written as $\sum_{i \in V} A_i(t + 1)\mathbf{e}_i$.

Hence from Lemma 4 and the relation 6 we can see that

$$\begin{aligned} \mathbf{X}^I(t + 1) &= \mathbf{X}^I(t) + \mathbf{R}g_I(\mathbf{X}^I(t)) + \sum_{i \in V} A_i(t + 1)\mathbf{e}_i \\ &\prec \mathbf{X}(t) + \mathbf{R}\mathbf{I}(t + 1) + \sum_{i \in V} A_i(t + 1)\mathbf{e}_i \\ &= \mathbf{X}(t + 1). \end{aligned} \quad (7)$$

REFERENCES

- [1] S. Hariharan and N. B. Shroff, "On sample-path optimal dynamic scheduling for sum-queue minimization in trees under the k -hop interference model," in *IEEE INFOCOM*, 2012.
- [2] V. J. Venkataramanan and X. Lin, "Low-complexity scheduling algorithm for sum-queue minimization in wireless convergecast," in *IEEE INFOCOM*, 2011.
- [3] L. Tassioulas and A. Ephremides, "Dynamic scheduling for minimum delay in tandem and parallel constrained queue networks," *Annals of Operations Research*, vol. 48, no. 4, pp. 333–355, 1994.
- [4] T. Ji, E. Athanasopoulou, and R. Srikant, "Optimal scheduling policies in small generalized switches," in *IEEE INFOCOM*, 2009.

- [5] G. R. Gupta and N. B. Shroff, "Delay analysis and optimality of scheduling policies for multi-hop wireless networks," *IEEE/ACM Transactions on Networking*, vol. 19, no. 1, pp. 129–141, 2010.
- [6] S. Hariharan and N. B. Shroff, "On optimal dynamic scheduling for sum-queue minimization in trees," in *IEEE WIOPT*, 2011.
- [7] B. Miller and C. Bisdikian, *Bluetooth Revealed: The Insider's Guide to an Open Specification for Global Wireless Communications*. Prentice Hall, 2000.
- [8] B. Hajek and G. Sasaki, "Link scheduling in polynomial time," *IEEE Transactions on Information Theory*, vol. 34, no. 5, pp. 910–917, 1988.
- [9] S. Hariharan and N. B. Shroff, "On sample-path optimal dynamic scheduling for sum-queue minimization in forests," *IEEE/ACM Transactions on Networking*, vol. 22, no. 1, pp. 151–164, 2014.
- [10] C. Florens, M. Franceschetti, and R. J. McEliece, "Lower bounds on data collection time in sensory networks," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 6, pp. 1110–1120, 2004.
- [11] J. C. Bermond, L. Gargano, and A. A. Rescigno, "Gathering with minimum delay in tree sensor networks," *Lecture Notes in Computer Science*, 2008.
- [12] L. Gargano and A. A. Rescigno, "Optimally fast data gathering in sensor networks," *Lecture Notes in Computer Science*, 2006.
- [13] B. Hajek and R. G. Ogier, "Optimal dynamic routing in communication networks with continuous traffic," *Networks*, vol. 14, no. 3, pp. 457–487, 1984.
- [14] S. Zhao and X. Lin, "On the design of scheduling algorithms for end-to-end backlog minimization in multi-hop wireless networks," in *IEEE INFOCOM*, 2012.
- [15] S. Hariharan, "Communication-efficient convergecast for data fusion in wireless sensor networks," Ph.D. dissertation, The Ohio State University, 2011. [Online]. Available: http://rave.ohiolink.edu/etdc/view?acc_num=osu1317672795



Srikanth Hariharan received his Ph.D. in Electrical and Computer Engineering from the Ohio State University in 2011, M.S. in Electrical and Computer Engineering from Purdue University in 2007, and B.Tech. in Electrical Engineering from the Indian Institute of Technology Madras in 2006. He is currently with the Analysis and Optimization department in AT&T Labs. His research interests include analysis and optimization of LTE/UMTS networks, data aggregation in wireless sensor networks, multi-hop wireless scheduling, and wireless security.



Ness B. Shroff (F'07) received his Ph.D. degree from Columbia University, NY, in 1994 and joined Purdue University as an Assistant Professor. At Purdue, he became Professor of the School of Electrical and Computer Engineering in 2003 and director of CWSA in 2004, a university-wide center on wireless systems and applications. In July 2007, he joined The Ohio State University as the Ohio Eminent Scholar of Networking and Communications, a chaired Professor of ECE and CSE. He is also a guest chaired professor of Wireless Communications

and Networking in the department of Electronic Engineering at Tsinghua University. His research interests span the areas of wireless and wireline communication networks. He is especially interested in fundamental problems in the design, performance, pricing, and security of these networks.

Dr. Shroff is a past editor for IEEE/ACM Trans. on Networking and the IEEE Communications Letters and current editor of the Computer Networks Journal. He has served as the technical program co-chair and general co-chair of several major conferences and workshops, such as the IEEE INFOCOM 2003, ACM Mobihoc 2008, IEEE CCW 1999, and WICON 2008. He was also a co-organizer of the NSF workshop on Fundamental Research in Networking (2003) and the NSF Workshop on the Future of Wireless Networks (2009). Dr. Shroff is a fellow of the IEEE. He received the IEEE INFOCOM 2008 best paper award, the IEEE INFOCOM 2006 best paper award, the IEEE IWQoS 2006 best student paper award, the 2005 best paper of the year award for the Journal of Communications and Networking, the 2003 best paper of the year award for Computer Networks, and the NSF CAREER award in 1996 (his INFOCOM 2005 paper was also selected as one of two runner-up papers for the best paper award).