Efficiently Finding Simple Schedules in Gaussian Half-Duplex Relay Line Networks

Yahya H. Ezzeldin†, Martina Cardone†, Christina Fragouli†, Daniela Tuninetti∗
† UCLA, Los Angeles, CA 90095, USA, Email: {yahya.ezzeldin, martina.cardone, christina.fragouli}@ucla.edu
∗ University of Illinois at Chicago, Chicago, IL 60607, USA, Email: danielat@uic.edu

Abstract—The problem of operating a Gaussian Half-Duplex (HD) relay network optimally is challenging due to the exponential number of listen/transmit network states that need to be considered. Recent results have shown that, for the class of Gaussian HD networks with \( N \) relays, there always exists a simple schedule, i.e., with at most \( N + 1 \) active states, that is sufficient for approximate (i.e., up to a constant gap) capacity characterization. This paper investigates how to efficiently find such a simple schedule over line networks. Towards this end, a polynomial-time algorithm is designed and proved to output a simple schedule that achieves the approximate capacity. The key ingredient of the algorithm is to leverage similarities between network states in HD and edge coloring in a graph. It is also shown that the algorithm allows to derive a closed-form expression for the approximate capacity of the Gaussian line network that can be evaluated distributively and in linear time.

I. INTRODUCTION

Computing the capacity of a wireless relay network is a long-standing open problem. For Half-Duplex (HD) networks, where the \( N \) relays cannot simultaneously transmit and receive, such problem is more challenging due to the \( 2^N \) possible listen/transmit configuration states that need to be considered. Recently, in [1] the authors proved the conjecture posed in [2], which states that simple schedules (i.e., with at most \( N + 1 \) active states) suffice for approximate (i.e., up to a constant gap) capacity characterization. This result is promising as it implies that the network can be operated close to its capacity with a limited number of state switches. However, to the best of our knowledge, it is not clear yet if such simple schedules can be found efficiently.

The main result of this work is an algorithm design that allows to compute a simple schedule that achieves the approximate capacity of the \( N \)-relay Gaussian HD line network with complexity \( \mathcal{O}(N^2) \). The algorithm leverages similarities between network states in HD and edge coloring in a graph, by associating different colors to links that cannot be activated simultaneously. In addition, the algorithm allows to derive the approximate capacity of the Gaussian HD line network in closed form. This expression has two appealing features: (i) it can be evaluated in linear time and (ii) it can be distributively computed among the \( N \) relays. The novelty and applicability of the results presented in this paper are two-fold: (i) they shed light on how to operate a class of Gaussian HD relay networks close to the capacity with the minimum number of state switches and (ii) they represent the first approximate capacity characterization in closed form for a class of Gaussian HD relay networks with general number of relays.

Related Work. The capacity of the \( N \)-relay Gaussian HD network is not known in general. Recent results in [3], [4] showed that the capacity can be approximated to within a constant gap by the cut-set upper bound evaluated with independent inputs and a schedule, which is independent of the transmitted and received signals. In the rest of the paper, we refer to this bound as the approximate capacity. To the best of our knowledge, the tightest known gap for Gaussian HD relay networks is of \( 1.96(N+2) \) bits per channel use, independently of the channel parameters [5].

In general, the evaluation of the approximate capacity is cast as an optimization problem over \( 2^N \) listen/transmit states. As \( N \) increases, this evaluation, as well as determining an optimal schedule of listen/transmit states, becomes computationally expensive. The authors in [6] designed an iterative algorithm to determine an approximately optimal schedule when the relays use decode-and-forward. In [7], the authors proposed a ‘grouping’ technique to address the complexity of the aforementioned optimization problem. This technique allows to compute the approximate capacity in polynomial-time for certain classes of Gaussian HD relay networks that include the line network as special case. While the results in [6] and [7] show that the approximate capacity can be efficiently obtained for special network topologies by solving a linear program, it is not clear how to construct (in polynomial time) a schedule that achieves the approximate capacity. Differently, in this work we design a polynomial-time algorithm that outputs a simple schedule, which achieves the approximate capacity of Gaussian HD line networks and allows to compute this quantity in closed form.

Paper Organization. Section II describes the \( N \)-relay Gaussian HD line network and presents known capacity results. Section III discusses our main results and their implications. Section IV simplifies the approximate capacity expression for Gaussian HD line networks. Section V designs an algorithm for finding a simple schedule for a Gaussian HD line network that achieves the approximate capacity. Section VI concludes the paper. Some of the proofs are delegated to the Appendix.

II. SYSTEM MODEL

We consider the \( N \)-relay Gaussian HD line network \( \mathcal{L} \) where a source node (node 0) wishes to communicate to
a destination node (node \(N + 1\)) through a route of \(N\) relays where each relay is operating in HD. The input/output relationship for the line network \(L\) is

\[
Y_i = (1 - S_i) h_{i,i-1} X_{i-1} S_{i-1} + Z_i, \quad \forall i \in [1 : N + 1],
\]

where: (i) \(X_i\) (respectively, \(Y_i\)) denotes the channel input (respectively, output) at the \(i\)-th node; (ii) \(S_i\) is the binary random variable which represents the state of node \(i\), i.e., if \(S_i = 0\) then node \(i\) is receiving, while if \(S_i = 1\) then node \(i\) is transmitting; (iii) \(Z_i\) indicates the additive white Gaussian noise at node \(i\), where the noises are assumed to be independent and identically distributed as \(CN(0,1)\); (iv) \(h_{i,j}\) denotes the channel complex coefficient from node \(j\) to node \(i\) and \(h_{i,j} = 0\) whenever \(j \neq i - 1\); the channel gains are assumed to be constant for the whole transmission duration and hence known to all nodes; (v) \(\lambda\) determines the fraction of time that the network operates in each of the states \(s \in [0:1]^N\), i.e., \(\lambda_s = \Pr(S_i = s_i, \forall i \in [1 : N])\); (vi) \(\Lambda = \{ \lambda \in \mathbb{R}^N : \lambda \geq 0, \|\lambda\|_1 = 1\}\) is the set of all possible schedules; (iii) \(\mathcal{R}_s\) (respectively, \(\mathcal{T}_s\)) represents the set of indices of relays receiving (respectively, transmitting) in the relaying state \(s \in [0:1]^N\); (iv) \(\mathcal{A}' = [1:N] \backslash \mathcal{A}\).

We can equivalently write the expression in (2) as

\[
C_L = \max_{\lambda \in \Lambda} \min_{\mathcal{A} \subseteq [1:N]} \sum_{s \in [0:1]^N} \lambda_s \sum_{i \in \mathcal{N} + 1 \cup \mathcal{A}} \ell_i, \quad \ell_i = \log (1 + |h_{i,i-1}|^2), \quad \forall i \in [1 : N + 1].
\]

where: (i) the schedule \(\lambda \in \mathbb{R}^N\) determines the fraction of time the network operates in each of the states \(s \in [0:1]^N\), i.e., \(\lambda_s = \Pr(S_i = s_i, \forall i \in [1 : N])\); (ii) \(\Lambda = \{ \lambda : \lambda \in \mathbb{R}^N, \lambda \geq 0, \|\lambda\|_1 = 1\}\) is the set of all possible schedules; (iii) \(\mathcal{R}_s\) (respectively, \(\mathcal{T}_s\)) represents the set of indices of relays receiving (respectively, transmitting) in the relaying state \(s \in [0:1]^N\); (iv) \(\mathcal{A}' = [1:N] \backslash \mathcal{A}\).

We can equivalently write the expression in (2) as

\[
C_L = \max_{\lambda \in \Lambda} \min_{\mathcal{A} \subseteq [1:N]} \sum_{s \in [0:1]^N} \lambda_s \sum_{i \in \mathcal{N} + 1 \cup \mathcal{A}} \ell_{i,s},
\]

where \(\ell_{i,s} = \begin{cases} \ell_i & \text{if } i \in \mathcal{R}_s \cup \{N + 1\} \text{ and } i - 1 \in \mathcal{T}_s \cup \{0\} \\ 0 & \text{otherwise} \end{cases}\).

Similarly, we denote with \(C_L^\lambda\), the HD rate achieved by the line network \(L\) when operated with the fixed schedule \(\lambda\), i.e.,

\[
C_L^\lambda = \min_{\mathcal{A} \subseteq [1:N]} \sum_{s \in [0:1]^N} \lambda_s \sum_{i \in \mathcal{N} + 1 \cup \mathcal{A}} \ell_{i,s}.
\]

Note that for all possible schedules \(\lambda\), \(C_L^\lambda \leq C_L\).

**Definition 1.** We say that a schedule \(\lambda\) is simple if the number of active states (i.e., states \(s\) such that \(\lambda_s \neq 0\)) is less than or equal to \(N + 1\). In other words, we have \(\|\lambda\|_0 \leq N + 1\).

In [9], it was observed that information can be conveyed by randomly switching the relay between transmit and receive modes. However, this only improves the capacity by a constant, at most 1 bit per relay.

**III. MAIN RESULTS AND DISCUSSION**

Our main result, stated in the theorem below, is two-fold: (i) it designs a polynomial-time algorithm that outputs a simple schedule optimal for approximate capacity and (ii) it provides a closed-form expression for the approximate capacity of the HD line network that can be evaluated in linear time.

**Theorem 1.** For the \(N\)-relay Gaussian HD line network \(L\) described in (1), a simple schedule optimal for approximate capacity can be obtained in \(O(N^2)\) time and the approximate capacity \(C_L\) in (3) is given by

\[
C_L = \min_{\lambda \in \Lambda} \sum_{s \in [0:1]^N} \lambda_s \sum_{i \in \mathcal{N} + 1 \cup \mathcal{A}} \ell_i.
\]

**Converse.** It is not hard to argue that the right-hand side of (5) is an upper bound on \(C_L\). This can be seen by assuming that, for a given \(i \in [1 : N]\), node \(i - 1\) perfectly cooperates with node 0 and similarly node \(i + 1\) perfectly cooperates with node \(N + 1\). Clearly, the capacity of this new line network is an upper bound on \(C_L\) and it has an approximate capacity equal to \(\sum_{i=1}^{N+1} \ell_i\). Since this is true for all \(i \in [1 : N]\), then \(C_L\) is less than or equal to the right-hand side of (5). The heart of the proof is thus to prove the achievability of (5).

Before we delve into the proof of the achievability in Theorem 1, we highlight the following remarks to motivate the need to search for a simple schedule for the line network.

**Remark 1.** Are two active states sufficient for approximate capacity characterization? Consider a line network with one relay. For this network, the schedule that achieves the approximate capacity has only two active states, which activate the links alternatively. Intuitively, one might think that this would extend to general number of relays. For example, for a network with \(N = 3\), can we achieve the approximate capacity by only considering the listen/transmit states \(s_1 = 010, s_2 = 101\)\

**Remark 2.** Can we a priori limit our search over a polynomial number of states? For the Full-Duplex (FD) line network, we can a priori limit our search for the minimum cut over \(N + 1\) cuts (instead of \(2^N\)). This reduction in the number of cuts is also possible for the HD line network as we prove in the next section. This fact raises the question whether we can also a priori reduce the search space for the active states to a polynomial set (instead of \(2^N\)). This is not possible as we state in the theorem below, which is proved in [10].

**Theorem 2.** With only the knowledge that relays are arranged in a line, the cardinality of the smallest search space of states over which a schedule optimal for approximate capacity can be found is \(\Omega(2^{N/3})\).
Remark 3. Theorem 1 has two appealing consequences:

1) The capacity of the line network with \( N \) relays can be computed in linear time in \( N \). This improves on the result in [7], where the approximate capacity can be found in polynomial time (but not linear in the worst case) by solving a linear program with \( O(N) \) variables.

2) The approximate capacity in Theorem 1 can be computed in a distributive way as follows. Each relay \( i \in [1:N] \) computes the quantity

\[
m_i = \min \left\{ \frac{\ell_i \ell_{i+1}}{\ell_i + \ell_{i+1}}, m_{i-1} \right\},
\]

where \( m_0 = \infty \), and sends it to relay \( i + 1 \). With this, at the end we have \( m_N = C_L \). In other words, for approximate capacity computation, it is only required that each relay knows the capacity of the incoming and outgoing links.

IV. FUNDAMENTAL CUTS IN HD LINE NETWORKS

In this section, we prove that in a Gaussian HD line network, we can compute the approximate capacity \( C_L \) in (3) by considering only \( N + 1 \) cuts, which are the same that one would need to consider if the network was operating in FD.

For the Gaussian line network \( L \), when all the \( N \) relays operate in FD, the FD capacity is given by

\[
C_L^{FD} = \min_{A \subseteq [1:N]} \sum_{i \in \{N+1\} \cup A, 1 \leq i \leq (0) \cup A^c} \ell_i = \min_{i \in [1:N+1]} \{ \ell_i \}
\]

that is, without explicit knowledge of the values of \( \ell_i \) or their ordering, the number of cuts over which we need to optimize (see \( C_L^{FD} \) in (7)) is \( N + 1 \). We refer to these cuts as fundamental. Let \( \mathcal{F} \) denote the set of these fundamental cuts (which are of the form \( A = [i : N], i \in [1 : N] \) or \( A = \emptyset \)), then for any cut \( A \) of the network, there exists a fundamental cut \( F(A) \in \mathcal{F} \) such that:

\[
\sum_{i \in \{N+1\} \cup F(A), 1 \leq i \leq (0) \cup F(A)^c} \ell_i \leq \sum_{i \in \{N+1\} \cup A, 1 \leq i \leq (0) \cup A^c} \ell_i.
\]

Furthermore, the function \( F(\cdot) \) does not depend on the values of \( \ell_i \). We next prove that the fundamental cuts in HD equal those in (7) for FD. Consider a fixed schedule \( \lambda \). Then, by using (8) for the inner summation in (4), for each \( s \in [0 : 1]^N \) we have

\[
\sum_{i \in \{N+1\} \cup F(A), 1 \leq i \leq (0) \cup F(A)^c} \ell_{i,s} \leq \sum_{i \in \{N+1\} \cup A, 1 \leq i \leq (0) \cup A^c} \ell_{i,s}.
\]

Thus, we can simplify (4) as

\[
C^\lambda = \min_{A \subseteq [1:N]} \sum_{s \in [0 : 1]^N} \lambda_s \sum_{i \in \{N+1\} \cup A, 1 \leq i \leq (0) \cup A^c} \ell_{i,s} = \min_{A \subseteq [1:N]} \sum_{s \in [0 : 1]^N} \lambda_s \sum_{i \in \{N+1\} \cup A, 1 \leq i \leq (0) \cup A^c} \ell_{i,s} = \min_{i \in [1:N+1]} \left( \sum_{s \in S_i} \lambda_s \right) \ell_i,
\]

where

\[
S_i = \{ s \in [0 : 1]^N | i \in \{N+1\} \cup \mathcal{R}_s, i-1 \in \{0\} \cup \mathcal{T}_s \}.
\]

The set \( S_i \subseteq [0 : 1]^N \) represents the collection of states that activate the \( i \)-th link. For illustration consider a network with \( N = 3 \). We have

\[
S_1 = \{000, 001, 010, 011\}, \quad S_2 = \{100, 101\}, \quad S_3 = \{010, 110\}, \quad S_4 = \{001, 011, 101, 111\}.
\]

Using the same arguments as in (9), we can similarly simplify the expression of \( C_L \) in (3). Thus, the result presented in this section explicitly provides the \( N + 1 \) cuts (out of the \( 2^N \) possible ones) over which it is sufficient to minimize in order to obtain \( C_L \) in (3).

V. FINDING A SIMPLE SCHEDULE OPTIMAL FOR APPROXIMATE CAPACITY

In this section, we design a polynomial-time algorithm that finds a simple schedule, which achieves the approximate capacity of the \( N \)-relay Gaussian HD line network. The algorithm leverages similarities between network states in HD and edge coloring in a graph. In particular, an edge coloring assigns colors to edges in a graph such that no two adjacent edges are colored with the same color. Similarly in HD, a network state cannot be a receiver and a transmitter simultaneously. Thus, if we assign one color to all activated links (viewed as edges) in a network state, this does not violate the rules of edge coloring in a graph. In what follows, we first explain how the algorithm makes use of these similarities assuming that the link capacities \( \ell_i \) are all integers and later in the section, we show how the algorithm extends to rational and real values of \( \ell_i \).

A. An Algorithm for Networks with Integer Link Capacities

We highlight the algorithm procedure in the following main steps and provide intuitions for each step. As a running example to illustrate the different steps we consider the line network \( L \) with \( N = 3 \) relays described in (6) with \( R = 1 \).

Step 1. Let \( M \) be a common multiple of the link capacities \( \ell_i \). For the line network \( L \) we construct an associated graph \( G_L \) where: (i) the set of nodes is the same as in the network \( L \) and (ii) each link with capacity \( \ell_i \) in \( L \) is replaced by \( n_i \) parallel edges, where

\[
n_i = \frac{M}{\ell_i}, \quad \forall i \in [1 : N + 1].
\]

Clearly, computing \( M \) and \( n_i \) requires \( O(N) \) operations. The main motivation behind this step is that from (9), it is not difficult to see that a good schedule would try to assign more weights \( \lambda_s \) to a link with a smaller capacity. Hence, if we treat edge colors as equally weighted, a link with a smaller capacity should get more colors. Thus, the main idea above is to assign \( n_i \) adjacent edges inversely proportional to \( \ell_i \).

Running Example. We have

\[
M = 6 \quad \text{and} \quad n_1 = 3, \quad n_2 = 3, \quad n_3 = 2, \quad n_4 = 6.
\]
Step 2. In this step, our goal is to edge color the graph \(G_{E_3}\). By noting that \(G_{E_3}\) is a bipartite graph, we know that an optimal coloring can be performed with \(\Delta\) colors, where \(\Delta\) is the maximum node degree and is equal to
\[
\Delta = \max_{i \in [1:N]} \{n_i + n_{i+1}\}. \tag{11}
\]
In particular, we define our coloring by the interval of colors \(C_i \subset [1: \Delta]\) that are assigned to the \(n_i\) edges that connect node \(i-1\) to node \(i\) such that \(|C_i| = n_i\). Specifically, we assign \(C_i\) for \(i \in [1:N+1]\)
\[
C_i = \begin{cases} [1:n_i], & i \text{ even} \\ [\Delta-n_{i+1}+1: \Delta], & i \text{ odd}. \end{cases} \tag{12}
\]
The complexity of this step is \(O(N)\), since for each \(i \in [1:N+1]\) we only compute two numbers that define the interval \(C_i\), that is the two limit points \(C_i^{(l)}\) and \(C_i^{(r)}\) of the interval, i.e., \(C_i = [C_i^{(l)} : C_i^{(r)}]\).

Running example. Since we have \(\Delta = 8\), then the assigned color intervals are
\[
C_1 = [6:8], \ C_2 = [1:3], \ C_3 = [7:8], \ C_4 = [1:6].
\]

Step 3. From the previous step, we have \(\Delta\) colors each of which corresponds to a network state running for \(1/\Delta\) fraction of time. However, some of these colors can represent the same operation states. For instance, in our running example, the colors 7 and 8 appear in both \(C_1\) and \(C_3\) (and nowhere else). Therefore, we can group the time fractions of colors 7 and 8 together, since they operate the network in the same way. To perform this color grouping, we run an iterative algorithm over the color intervals \(C_i, i \in [1:N+1]\) constructed in the previous step, which outputs a schedule for the network. The algorithm pseudocode is shown in Algorithm 1 and can be summarized as follows:

1) We first find a descendingly ordered set of colors \(p_u\) at which the network state changes. The network state changes whenever an interval \(C_i\) begins or ends. Therefore in this step, we sort the unique elements of an array \(p\) that contains the \(C_i^{(l)}\) and \(C_i^{(r)} + 1\) for all \(i \in [1:N+1]\). Since there are \(N+1\) different \(C_i\) then this operation takes \(O(N \log N)\) time. At this point, it is worth noting that for odd \(i\), we have \(C_i^{(r)} = \Delta\) while for even \(i\), we have \(C_i^{(l)} = 1\). Thus, the descendingly ordered array \(p_u\) has at most \(N+2\) unique elements.

Running example. We have \(p_u = \{9, 7, 6, 4, 1\}\).

2) Next, we go through the array \(p_u\) to construct the group of colors \(I_j\). Each set \(I_j\) represents a network state and all colors in a set \(I_j\) operate the network in the same way. To do this we compute the endpoints \(I_j^{(l)}\), \(I_j^{(r)}\) for each \(I_j\). The fraction of time the network operates in the state represented by \(I_j\) is stored in the vector \(w\) and is calculated as \(|I_j|/\Delta\).

3) For each \(I_j\) constructed, the algorithm performs a loop of \(N\) iterations in order to determine the state of each node in this particular network configuration state and records this in a row of the matrix \(\Lambda\). Thus, the active states are represented by rows of \(\Lambda\). The algorithm finally outputs the variables \(\Lambda\) and \(w\). The complexity of steps 2 and 3 is \(O(N^2)\).

Running example. Algorithm 1 outputs
\[
I_1 = [7:8], \ \Lambda(1,:) = 010, \ w(1) = 2/8, \\
I_2 = [6:7], \ \Lambda(2,:) = 001, \ w(2) = 1/8, \\
I_3 = [4:5], \ \Lambda(3,:) = 111, \ w(3) = 2/8, \\
I_4 = [1:3], \ \Lambda(4,:) = 101, \ w(4) = 3/8.
\]

Finally, we note that since \(p_u\) has at most \(N+2\) terms, then the number of states output by the algorithm is at most \(N+1\), i.e., the schedule is simple as also stated in Theorem 1.

Algorithm 1 Grouping Edge Colors

Input \(N, \Delta, (C_i^{(l)}, C_i^{(r)}), \forall i \in [1:N+1]\)
Output \(\Lambda, w\)
for each \(i \in [1:N+1]\) do
\(p(i) \leftarrow C_i^{(l)}, \ p(i+N+1) \leftarrow C_i^{(r)} + 1\)
end for
\(p_u \leftarrow \text{UniqueDescSort}(p)\)
for each \(j \in [1:length(p_u)]\) do
\(I_j^{(l)} \leftarrow p(u(j)), \ I_j^{(r)} \leftarrow p(u(j)+1)\)
\(w(j) \leftarrow (I_j^{(r)} - I_j^{(l)} + 1)/\Delta, \ \text{state} \leftarrow 1\)
end for
for each \(i \in [1:N+1]\) do
if \(I_j^{(l)} : I_j^{(r)} \subseteq C_i\) then
\(\text{state} \leftarrow 0\)
if \(i < N + 1\) then \(\Lambda(j,i) \leftarrow 0\)
if \(i > 1\) then \(\Lambda(j,i-1) \leftarrow 1\)
end if
else if \(i < N + 1\) then
\(\Lambda(j,i) \leftarrow \text{state}\)
end else

B. Rate Achieved by the Schedule

In the previous subsection we developed an algorithm that outputs a simple schedule for the \(N\)-relay Gaussian HD line network \(L\). We now derive the rate that the constructed schedule achieves. Let \(\lambda\) be the schedule output by Algorithm 1, then we have the following relation \(\forall i \in [1:N+1]\)
\[
\sum_{s \in S_i} \lambda_s = \sum_{j \in [1:K]} \sum_{\lambda_{(j,:)} \in S_i} w(j) = \sum_{j \in [1:K]} \left(\frac{|I_j|}{\Delta} = \frac{n_i}{\Delta}\right) \tag{13}
\]
where \(K\) is the number of network configuration states output by Algorithm 1 and \(S_i\) is the set of states which make the link of capacity \(\ell_i\) active and is defined in (9). From (9), (10) and (13), we can simplify the achievable rate \(C_L\) as
\[
C_L = \min_{i \in [1:N+1]} \left(\sum_{s \in S_i} \lambda_s\right) \ell_i^{(l)} \left(\frac{n_i}{\Delta} \right) \frac{1}{\ell_i} \frac{n_i}{\Delta} = M. \tag{14}
\]
From the definition of maximum degree \(\Delta\) in (11), we obtain
\[
\frac{\Delta}{M} = \frac{1}{M} \max_{i \in [1:N]} \{n_i + n_{i+1}\} = \max_{i \in [1:N]} \left\{\frac{1}{\ell_i} + \frac{1}{\ell_i+1}\right\}. \tag{15}
\]
Finally, by substituting (15) into (14) we obtain that the simple schedule \(\lambda\) constructed from our proposed polynomial-time algorithm achieves the right-hand side of (5). This concludes the proof of Theorem 1 when the link capacities are integers.
C. Extension to Real Link Valued Capacities

The extension of the algorithm in Section V-A to networks with rational link capacities is straightforward, by simply multiplying each of the link capacities with a common multiple of the denominators. In this subsection, we focus on how the algorithm can be used to find a schedule for the line network $\mathcal{L}$ with real link capacities such that the rate achieved by this schedule is at most a constant gap $\varepsilon$ away from $C_e$.

For a fixed $\varepsilon > 0$, let $\mathcal{L}_{q,e}$ be a line network with rational link capacities $q_i$, such that

$$\forall i \in [1 : N + 1], \quad q_i \in \mathbb{Q}, \quad \ell_i - \varepsilon \leq q_i \leq \ell_i.$$  \hfill (16)

Such $\mathcal{L}_{q,e}$ always exists (but it is not unique) since the set of rationals $\mathbb{Q}$ is dense in $\mathbb{R}$. We now relate $C_{q,e}$ and $C_e$ by appealing to the lemma below, which we prove in Appendix A.

**Lemma 3.** Let $\mathcal{L}$ be a line network with real link capacities $\ell_i$, then $\forall \varepsilon > 0$, we have

$$C_{L_{q,e}} \leq C_{L_q} \leq C_L \leq C_{L_{q,e}} + \varepsilon,$$  \hfill (17)

where the line network $\mathcal{L}_{q,e}$ is constructed as in (16) and $\lambda^{q,e}$ is an optimal schedule of the line network $\mathcal{L}_{q,e}$.

The result in Lemma 3 has the following implications:
1) The statement of Lemma 3 directly implies that $|C_L - C_{L_{q,e}}| \leq \varepsilon$. Since this holds for all $\varepsilon > 0$, then we can use the algorithm in Section V-A on $\mathcal{L}_{q,e}$ to get a simple schedule $\lambda^{q,e}$ that achieves a rate $C_{L_{q,e}}$ that is at most $\varepsilon$ away from $C_e$.

2) From Lemma 3, we have that $\lim_{\varepsilon \to 0} C_{L_{q,e}} \leq C_L \leq \lim_{\varepsilon \to 0} [C_{L_{q,e}} + \varepsilon]$.

Additionally, from Section V-B, we know that for the network $\mathcal{L}_{q,e}$ with rational link capacities $q_i$, we have

$$\lim_{\varepsilon \to 0} C_{L_{q,e}} = \lim_{\varepsilon \to 0} \min_{\varepsilon > 0} \frac{q_i q_{i+1}}{q_i + q_{i+1}} = \min_{j \in [1 : N]} \left\{ \frac{\ell_j \ell_{j+1}}{\ell_j + \ell_{j+1}} \right\}.$$  \hfill (18)

These two observations imply that, for any line network $\mathcal{L}$ with real link capacities, we have

$$C_L = \min_{j \in [1 : N]} \left\{ \frac{\ell_j \ell_{j+1}}{\ell_j + \ell_{j+1}} \right\}.$$  \hfill (19)

This concludes the proof of Theorem 1 for real link capacities.

VI. CONCLUSION

In this work we developed a polynomial-time algorithm for finding a simple schedule (one with at most $N + 1$ active states) that achieves the approximate capacity of the $N$-relay Gaussian HD line network. We characterized the rate achieved by the constructed schedule in closed form, hence providing a closed-form expression for the approximate capacity of the Gaussian HD line network. To the best of our knowledge, this is the first work which provides a closed-form expression for the approximate capacity of an HD relay network with general number of relays and designs an efficient algorithm to find simple schedules which achieve it.

APPENDIX A

The second inequality in (17) is straightforward from the definition of $C_e$. Therefore, we need to prove the first and third inequalities. To prove the first inequality note that, from (9), we can upperbound $C_{L_{q,e}}$ as

$$C_{L_{q,e}} = \min_{\varepsilon \in [1 : N + 1]} \left( \sum_{s \in S} \lambda^{q,e}_{s} \right) q_i,$$

where the inequality in (a) follows since, from the construction in (16), $\forall i \in [1 : N + 1]$, we have $q_i \leq \ell_i$. This proves the first inequality. To prove the third inequality, we use the fact that, from the construction in (16), $\forall i \in [1 : N + 1]$, we have $q_i \geq \ell_i - \varepsilon$. This implies that for any schedule $\lambda$, we have

$$C_L \geq \min_{\varepsilon \in [1 : N + 1]} \left( \sum_{s \in S} \lambda_{s} \right) (\ell_i - \varepsilon) \leq \sum_{s \in S} \lambda_{s} (\ell_i - \varepsilon) = q_i C_{L_{q,e}},$$

where the inequality in (a) follows since, from the construction in (16), $\forall i \in [1 : N + 1]$, we have $q_i \leq \ell_i$. This proves the third inequality.

REFERENCES

[1] M. Cardone, D. Tuninetti, and R. Knopp, “The approximate optimality of simple schedules for half-duplex multi-relay networks,” in IEEE Information Theory Workshop (ITW), April 2015, pp. 1–5.

[2] S. Brahma, C. Fragouli, and A. Ozgür, “On the complexity of scheduling in half-duplex diamond networks,” IEEE Transactions on Information Theory, vol. 62, no. 5, pp. 2557–2572, May 2016.

[3] A. S. Avestimehr, S. N. Diggavi, and D. N. C. Tse, “Wireless network information flow: A deterministic approach,” IEEE Transactions on Information Theory, vol. 57, no. 4, pp. 1872–1905, April 2011.

[4] A. Ozgür and S. N. Diggavi, “Approximately achieving Gaussian relay network capacity with lattice-based QMF codes,” IEEE Transactions on Information Theory, vol. 59, no. 12, pp. 8275–8294, December 2013.

[5] M. Cardone, D. Tuninetti, R. Knopp, and U. Salim, “Gaussian half-duplex relay networks: improved constant gap and connections with the assignment problem,” IEEE Transactions on Information Theory, vol. 60, no. 6, pp. 3559 – 3575, June 2014.

[6] L. Ong, M. Motani, and S. J. Johnson, “On capacity and optimal scheduling for the half-duplex multiple-relay channel,” IEEE Transactions on Information Theory, vol. 58, no. 9, pp. 5770–5784, June 2012.

[7] R. Ekin, F. Parvaresh, I. Shomorony, and A. Avestimehr, “Computing half-duplex schedules in Gaussian relay networks via min-cut approximations,” IEEE Transactions on Information Theory, vol. 60, no. 11, pp. 7204–7220, November 2014.

[8] S. Lim, Y.-H. Kim, A. El Gamal, and S.-Y. Chung, “Noisy network coding,” IEEE Transactions on Information Theory, vol. 57, no. 5, pp. 3132–3152, May 2011.

[9] G. Kramer, “Models and theory for relay channels with receive constraints,” in 42nd Annual Allerton Conference on Communication, Control, and Computing, September 2004, pp. 1312–1321.

[10] Y. H. Erezoldin, M. Cardone, C. Fragouli, and D. Tuninetti, “Finding simple half-duplex schedules in Gaussian relay line networks,” arXiv:1701.04426, January 2017.