Distributed Cross-Layer Optimization in Wireless Networks: A Second-Order Approach

Jia Liu[†] Cathy H. Xia[*] Ness B. Shroff[†] Hanif D. Sherali[‡]
[†]Department of Electrical and Computer Engineering, The Ohio State University
[*] Department of Integrated Systems Engineering, The Ohio State University
[‡]Grado Department of Industrial and Systems Engineering, Virginia Tech

Abstract

Due to the rapidly growing scale and heterogeneity of wireless networks, the design of distributed cross-layer optimization algorithms have received significant interest from the networking research community. So far, the standard distributed cross-layer approach in the literature is based on first-order Lagrangian dual decomposition and the subgradient method, which suffers a slow convergence rate. In this paper, we make the first known attempt to develop a distributed Newton's method, which is *second-order* and enjoys a *quadratic* convergence rate. However, due to interference in wireless networks, the Hessian matrix of the cross-layer problem has an *non-separable* structure. As a result, developing a distributed second-order algorithm is far more challenging than its counterpart for wireline networks. Our main results in this paper are two-fold: i) For a special network setting where all links mutually interfere, we derive decentralized *closed-form expressions* to compute the Hessian inverse; ii) For general wireless networks where the interference relationships are arbitrary, we propose a distributed iterative matrix splitting scheme for the Hessian inverse. These results successfully lead to a new theoretical framework for cross-layer optimization in wireless networks. More importantly, our work contributes to an exciting second-order paradigm shift in wireless networks optimization theory.

1 Introduction

The proliferation of mobile communication devices (e.g., smartphones, tablets, etc.) has been accompanied by a rapid growth in scale and heterogeneity of wireless networks. As a result, distributed cross-layer algorithms have received significant interest from the wireless networking research community in recent years. In the literature, the standard approach for distributed optimization in wireless networks is based on the Lagrangian dual decomposition framework and the subgradient method (LD-SG), which is primarily due to its elegant cross-layer implementations (see, e.g., [1] and references therein). The LD-SG framework is also intimately linked to the celebrated throughput-optimal

"back-pressure" algorithm [2], which has led to a large number of routing and scheduling schemes for wireless networks (see, e.g., [3–6]). However, despite its theoretical and engineering appeals, the performance of LD-SG is not satisfactory in practice. Being a first-order approach in nature (search directions are based on the first-order supports of the dual function), the subgradient method suffers from a slow convergence rate and is sensitive to step-size choices [7]. Due to these limitations, in this paper, we consider designing a distributed Newton's method for cross-layer optimization in wireless networks. The fundamental philosophy of this approach is that, being a *second-order* method, a distributed Newton's algorithm exploits both the gradient and Hessian information in determining search directions. Hence, an appropriately designed distributed Newton's method would also enjoy the powerful *quadratic rate of convergence* as in classical Newton type methods [7, 8].

However, developing second-order distributed algorithms for wireless networks is highly challenging and, to our knowledge, results in this area remain elusive. Due to a very different problem structure in wireless networks, techniques used for developing distributed second-order algorithms in wireline networks [9–11] cannot be directly applied (see Section 2 for more detailed discussions). Generally speaking, in a distributed second-order algorithm, computing the primal and dual search directions typically requires decomposing the inverses of the Hessian matrix and a weighted Laplacian matrix (weighted by the Hessian inverse), and then distributing each piece to each network entity (i.e., a node or a link). Unfortunately, unlike wireline networks for which the Hessian is (block) diagonal (see [11]), the Hessian's structure is *non-separable* due to the inherent interference in wireless networks. What is worse is that, not only are both the Hessian and weighted Laplacian inversions cumbersome in large-scale wireless networks, the obtained inverses also have no sparsity structure in general. Hence, distributed computations of the Hessian and weighted Laplacian inversion problems in wireless networks are far more difficult than their counterparts in wireline networks.

The key contribution of this paper is that we successfully develop a series of new second-order techniques to overcome all of the above difficulties in wireless networks. Hence, our work can be viewed as the first building block towards the development of an analytical foundation for cross-layer design that provides second-order convergence speed. The main technical contributions of this paper are as follows:

- We first consider a special network setting where every two links mutually interfere and cannot transmit simultaneously. This special setting is a relevant model for many interesting collision-based network architectures within a common interference domain (e.g., CSMA networks, cellular uplinks/downlinks, dense ad hoc networks, etc.). In this case, by exploiting the special "arrow-head" sparsity structure in the Hessian (cf. Eq. (18)), we prove that the inverse of the Hessian matrix is also an "arrow-head" matrix and can be computed in *closed-form*, thus significantly reducing the computational complexity. More importantly, the derived closed-form expression of

each entry in the Hessian inverse naturally leads to a distributed implementation.

- We next consider general wireless networks where the interference relationships among links are arbitrary. In the general case, since the Hessian inverse is non-sparse and deriving its closed-form expressions is intractable, we propose to iteratively compute the Hessian inverse and the weighted Laplacian inverse by a new *double matrix-splitting* technique. This double matrix-splitting scheme can be parameterized for convergence speed tuning. More importantly, this double matrix-splitting scheme can be implemented in a distributed fashion.

- We offer interesting insights and networking interpretations for our proposed distributed algorithms, as well as the connections with and differences from first-order approaches. This further advances our understanding of second-order approaches in wireless network optimization theory.

To the best of our knowledge, this paper is the first work that develops a distributed Newton's method for cross-layer optimization in wireless networks. Collectively, our results serve as an important first step in providing a cross-layer solution for wireless networks using second-order techniques.

The remainder of this paper is organized as follows. In Section 2, we review related work in the literature, putting our work in a comparative perspective. Section 3 introduces the network model and problem formulation. Section 4 develops the principal components of our distributed Newton's method. Section 5 presents some relevant numerical results, and Section 6 concludes this paper.

2 Related Work

Since distributed second-order methods for wireless networks have not been investigated in the literature, the works being surveyed herein are for wireline networks only. Historically, second-order methods for network optimization (both centralized and distributed) date back to the 1980s, including, e.g., a centralized projected Newton's method for multi-commodity flow problems [12] and a distributed conjugate gradient direction method for solving pure minimum cost flow routing problems [13]. These early attempts all employed gradient projections to identify feasible search directions. In contrast, most of the recent works in this area [9–11, 14–16] are based on the interior-point approach [17] due to its superior efficiency in both theory and practice. One of the first applications of an interior-point based second-order method was developed for the pure flow control problem (with fixed routing) [14], where Zymnis *et al.* proposed a centralized truncated-Newton's primal-dual algorithm. Bickson *et al.* [15, 16] also studied the same problem and designed a distributed algorithm based on the Gaussian belief propagation technique to avoid direct Hessian inversion [18]. Alternatively, Wei *et al.* [10] approached the same distributed flow control problem and computed the Hessian inverse based on an iterative matrix-splitting scheme. A distributed Newton's method was also developed

for the pure minimum cost routing problem (with fixed source rates) by Jadbabaie *et al.* in [9], where they proposed a consensus-based local averaging scheme to iteratively compute the Hessian inverse and established its convergence using spectral graph theory [19]. Finally, in our previous work [11], we showed that, through a suitable reformulation that exposes a block diagonal structure in the Hessian matrix, a distributed Newton's method can be developed for the more complex joint multi-path routing and flow control problem by generalizing the matrix-splitting idea in [11]. However, we point out that none of the aforementioned techniques can be directly applied to wireless networks due to a completely different Hessian matrix structure (cf. Eq. (18)), and our development of the distributed Newton's method for wireless networks cross-layer optimization is completely new.

3 Network Model and Problem Formulation

We first introduce our notation used in this paper. We use boldface to denote matrices and vectors. For a matrix $\mathbf{A}$, $\mathbf{A}^T$ denotes the transpose of $\mathbf{A}$. $\text{Diag}\{\mathbf{A}_1, \ldots, \mathbf{A}_N\}$ represents the block diagonal matrix with matrices $\mathbf{A}_1, \ldots, \mathbf{A}_N$ on its main diagonal. Also, $\text{diag}\{\mathbf{A}\}$ represents the vector containing the main diagonal entries of $\mathbf{A}$. We let $(\mathbf{A})_{ij}$ represent the entry in the i-th row and j-th column of matrix $\mathbf{A}$ and let $(\mathbf{v})_m$ represent the m-th entry of vector $\mathbf{v}$. We let $\mathbf{I}_K$ denote the K-dimensional identity matrix, and let $\mathbf{1}_K$ and $\mathbf{0}_K$ denote the K-dimensional vectors whose elements are all ones and zeros. We let $\mathbf{e}_K^{(k)}$ denote the k-th vector in the natural basis of $\mathbb{R}^K$ (i.e., the k-th entry is "1" and other entries are "0").

Network layer model. In this paper, a wireless network is represented by a directed graph, denoted by $\mathcal{G} = \{\mathcal{N}, \mathcal{L}\}$, where $\mathcal{N}$ and $\mathcal{L}$ are the sets of nodes and links, respectively. We assume that $\mathcal{G}$ is connected. The cardinalities of the sets $\mathcal{N}$ and $\mathcal{L}$ are $|\mathcal{N}| = N$ and $|\mathcal{L}| = L$, respectively. We use the so-called *node-arc incidence matrix* (NAIM) [20] $\mathbf{A} \in \mathbb{R}^{N \times L}$ to represent the network topology of $\mathcal{G}$. Let $\text{Tx}(l)$ and $\text{Rx}(l)$ denote the transmitting and receiving nodes of link l, respectively. The entries of $\mathbf{A}$ are thus defined as follows:

$$(\mathbf{A})_{nl} = \begin{cases} 1, & \text{if } n = \text{Tx}(l), \\ -1, & \text{if } n = \text{Rx}(l), \\ 0, & \text{otherwise.} \end{cases} \tag{1}$$

In the network, different source nodes send independent data to their intended destination nodes, potentially through multi-path and multi-hop routing. Suppose that there are F sessions in the network, representing F different commodities. We denote the source and destination nodes of session f, $1 \leq f \leq F$, as $\text{Src}(f)$ and $\text{Dst}(f)$, respectively. The source flow rate of session f is denoted by a scalar $s_f \in \mathbb{R}_+$. For session f, we use a *source–destination vector* vector $\mathbf{b}_f \in \mathbb{R}^N$ to represent the

supply–demand relationship of session f. More specifically, the entries in $\mathbf{b}_f$ are defined as follows:

$$(\mathbf{b}_f)_n = \begin{cases} 1, & \text{if } n = \text{Src}(f), \\ -1, & \text{if } n = \text{Dst}(f), \\ 0, & \text{otherwise.} \end{cases} \tag{2}$$

For every link l, we let $x_l^{(f)} \geq 0$ represent the flow amount of session f on link l. We assume that the network is a flow-balanced system, i.e., the following flow-balanced constraints hold at each node:

$$\begin{aligned} \sum_{l \in \mathcal{O}(n)} x_l^{(f)} - \sum_{l \in \mathcal{I}(n)} x_l^{(f)} &= s_f, && \text{if } n = \text{Src}(f), \\ \sum_{l \in \mathcal{O}(n)} x_l^{(f)} - \sum_{l \in \mathcal{I}(n)} x_l^{(f)} &= 0, && \text{if } n \neq \text{Src}(f), \text{Dst}(f), \\ \sum_{l \in \mathcal{O}(n)} x_l^{(f)} - \sum_{l \in \mathcal{I}(n)} x_l^{(f)} &= -s_f, && \text{if } n = \text{Dst}(f), \end{aligned}$$

where $\mathcal{O}(n)$ and $\mathcal{I}(n)$ represent the sets of outgoing and incoming links at node n, respectively. We let $\mathbf{x}^{(f)} \triangleq [x_1^{(f)}, \ldots, x_L^{(f)}]^T \in \mathbb{R}_+^L$ denote the *routing vector* for session f across all links. Using the notation $\mathbf{A}$, $\mathbf{b}_f$, and $\mathbf{x}^{(f)}$, the above flow-balanced constraints can be compactly written as:

$$\mathbf{A}\mathbf{x}^{(f)} - s_f \mathbf{b}_f = \mathbf{0}, \quad \forall f = 1, 2, \ldots, F. \tag{3}$$

Note that in (3), $\mathbf{A}$ is not of full row rank (because all columns sum up to zero). To eliminate the redundant rows in $\mathbf{A}$, we let $\mathbf{A}^{(f)} \in \mathbb{R}^{(N-1)\times L}$ be obtained by deleting from $\mathbf{A}$ the row corresponding to the node $\text{Dst}(f)$. It is easy to verify that $\mathbf{A}^{(f)}$ is of full row rank [20]. Also, we let $\widetilde{\mathbf{b}}^{(f)} \in \mathbb{R}^{N-1}$ be obtained by deleting from $\mathbf{b}_f$ the entry corresponding to the node $\text{Dst}(f)$. Accordingly, we rewrite (3) as:

$$\mathbf{A}^{(f)}\mathbf{x}^{(f)} - s_f \widetilde{\mathbf{b}}^{(f)} = \mathbf{0}, \quad \forall f = 1, 2, \ldots, F. \tag{4}$$

Link layer model. In this paper, we adopt the following collision-based interference model at the link layer: In a given time instant, due to the shared nature of wireless media, only a subset of links can be activated simultaneously without interfering with each other. To model this, we let $\mathcal{C} \triangleq \{\mathbf{c}^{(1)}, \ldots, \mathbf{c}^{(I)}\}$ denote the set of all possible interference-free link rate vectors, where $\mathbf{c}^{(i)} \triangleq [C_1^{(i)}, \ldots, C_L^{(i)}]^T \in \mathbb{R}_+^L$. In $\mathbf{c}^{(i)}$, if $C_{l_1}^{(i)} > 0$ and $C_{l_2}^{(i)} > 0$ for some $l_1, l_2 \in \mathcal{L}$, then it implies that links l_1 and l_2 do not interfere with each other and are both activated. Under this model, selecting a link rate vector from $\mathcal{C}$ in each time instant is equivalent to activating a subset of interference-free links. For simplicity, in this paper, we do not consider channel variations (i.e., $C_l^{(i)}$ is time-varying due to fading and/or mobility of the nodes) when selecting the subset of interference-free links. Such "opportunism" of exploiting channel state information (CSI) will be left for our future study.

Now, we let $\Lambda \triangleq \text{Co}(\mathcal{C}) \subset \mathbb{R}_+^L$ denote the link capacity region under the interference-free restriction, where $\text{Co}(\cdot)$ represents the convex-hull operation. Then, a necessary condition for the network to be stable is that the flow routing vectors satisfy $\sum_{f=1}^{F} \mathbf{x}^{(f)} \in \Lambda$. Further, it is well-known that the convex-hull operation in Λ can be achieved through a standard time-sharing argument [1, 2]. Thus, we let t_i represent the fraction of time during which link rate vector $\mathbf{c}^{(i)}$ is selected, where t_i satisfies $0 \leq t_i \leq 1$ and $\sum_{i=1}^{I} t_i = 1$. With time-sharing, the capacity of link l under the interference-free restriction can be computed as $C_l = \sum_{i=1}^{I} t_i C_l^{(i)}$. Then, the aforementioned stability condition can be written explicitly as:

$$\sum_{f=1}^{F} x_l^{(f)} \leq \sum_{i=1}^{I} t_i C_l^{(i)}, \quad \forall l = 1, \ldots, L, \tag{5}$$

where t_i, $\forall i$, are decision variables. We remark that (5) is a necessary condition on the feasibility of average flow rates, and can be viewed as a relaxation of the instantaneous link capacity constraint where time-sharing is not allowed (i.e., replacing t_i by $\varphi_i^\tau \in \{0, 1\}$ in each time instant τ, where $\varphi_i^\tau = 1$ if $\mathbf{c}^{(i)}$ is selected in τ or 0 otherwise). Hence, the solutions obtained via (5) may be infeasible under the instantaneous link capacity constraint. But to enable the development of second-order methods while drawing useful insights for future scheduling schemes design, we choose to work with the constraints in (5) in this paper. We note, however, that the solution obtained via (5) is indeed achievable under time-sharing.

Problem formulation. We associate a utility function $U_f(s_f) : \mathbb{R}_+ \to \mathbb{R}$ with each session f. The overall network utility is given by $\sum_{f=1}^{F} U_f(s_f)$. We also assume that the utility functions U_f are strictly concave, monotonically increasing, twice continuously differentiable, and reversely self-concordant (see [8] for the definition of self-concordance). Our objective is to maximize the overall network utility. Putting together the models described earlier, we can formulate the cross-layer optimization (CLO) problem as follows:

CLO:

$$\begin{aligned}
\text{Maximize} \quad & \sum_{f=1}^{F} U_f(s_f) \\
\text{subject to} \quad & \mathbf{A}^{(f)}\mathbf{x}^{(f)} - s_f \widetilde{\mathbf{b}}^{(f)} = \mathbf{0}, && \forall f = 1, \ldots, F, \\
& \sum_{f=1}^{F} x_l^{(f)} \leq \sum_{i=1}^{I} t_i C_l^{(i)}, && \forall l = 1, \ldots, L, \\
& \sum_{i=1}^{I} t_i = 1, \\
& x_l^{(f)} \geq 0, \ \forall f, l; \quad s_f \geq 0, \ \forall f; \quad t_i \geq 0, \ \forall i.
\end{aligned}$$

Note that Problem CLO is a convex program and can be solved in the Lagrangian dual domain with a zero duality gap [7,8]. Moreover, due to the separable structure of the dual function, Problem CLO can be solved distributedly by the dual decomposition and subgradient optimization (LD-SG) framework (see [1] or [21, Appendix A] for an overview). However, as mentioned earlier, the convergence performance of LD-SG is unsatisfactory. In what follows, we will investigate a new distributed second-order method to solve Problem CLO.

4 A Distributed Newton's Method

In this section, we first reformulate Problem CLO to facilitate the second-order design of our distributed Newton's method in Section 4.1. Then, we investigate its Hessian matrix structure in Section 4.2. The distributed computations of the primal Newton directions and the dual variables are presented in Sections 4.3 and 4.4, respectively.

4.1 Problem Reformulation and Interior-Point Based Distributed Newton's Method

We start by reformulating Problem CLO using the interior-point framework. Following the standard interior-point approach [17], we apply a logarithmic barrier function to all inequality constraints and then accommodate them in the objective function. As a result, the augmented objective function (to be minimized) can be written as follows:

$$f_\mu(\mathbf{y}) = -\mu \sum_{f=1}^{F} U_f(s_f) - \sum_{l=1}^{L} \log\left(\sum_{i=1}^{I} t_i C_l^{(i)} - \sum_{f=1}^{F} x_l^{(f)} \right) - \sum_{f=1}^{F} \log(s_f) - \sum_{l=1}^{L}\sum_{f=1}^{F} \log(x_l^{(f)}) - \sum_{i=1}^{I} \log(t_i), \quad (6)$$

where $\mathbf{y} \triangleq \left[s_1 \cdots s_F \middle| x_1^{(1)} \cdots x_1^{(F)} \middle| \cdots \middle| x_L^{(1)} \cdots x_L^{(F)} \middle| t_1 \cdots t_I\right]^T$ groups all variables. In (6), $\mu > 0$ is a parameter that is used to track the central path in the interior-point method as $\mu \to \infty$ [8]. Moreover, we let

$$\mathbf{M} \triangleq \left[\begin{array}{c|ccc|c} \mathbf{B} & \mathbf{A}_1 & \cdots & \mathbf{A}_L & \\ \hline & & & & \mathbf{1}_I^T \end{array}\right] \in \mathbb{R}^{[(N-1)F+1]\times[(L+1)F+I]},$$

where $\mathbf{B}$ and $\mathbf{A}_l$ are defined as $\mathbf{B} \triangleq \text{Diag}\{\widetilde{\mathbf{b}}^{(1)}, \ldots, \widetilde{\mathbf{b}}^{(F)}\}$, and $\mathbf{A}_l \triangleq \text{Diag}\{-\mathbf{a}_l^{(1)}, \ldots, -\mathbf{a}_l^{(F)}\}$, and where in the definition of $\mathbf{A}_l$, the vector $\mathbf{a}_l^{(f)}$ is the l-th column in the matrix $\mathbf{A}^{(f)}$ in Problem CLO (i.e., $\mathbf{A}^{(f)} = \left[\mathbf{a}_1^{(f)}, \mathbf{a}_2^{(f)}, \ldots, \mathbf{a}_L^{(f)}\right]$). Then, we can reformulate Problem CLO as follows:

$$\begin{aligned} \textbf{R-CLO: Minimize} \quad & f_\mu(\mathbf{y}) \\ \text{subject to} \quad & \mathbf{M}\mathbf{y} = \mathbf{e}_{\bar{n}}^{(\bar{n})}, \end{aligned} \quad (7)$$

where $\bar{n} = (N-1)F + 1$. In $f_\mu(\mathbf{y})$, note that as $\mu \to \infty$, the original objective function of Problem CLO dominates the barrier functions, and hence the solution of Problem R-CLO approaches that of Problem CLO asymptotically. Further, since μ can be increased exponentially (e.g., letting $\mu_k = 2^k$), it suffices to focus on a second-order solution to the $f_\mu(\mathbf{y})$ problem in order to achieve a second-order convergence speed.

Now, we solve $f_\mu(\mathbf{y})$ by applying the (centralized) Newton's method, which is a second-order algorithm. Starting from an initial feasible solution $\mathbf{y}^0$, the centralized Newton's method iteratively searches for an optimal solution as follows:

$$\mathbf{y}^{k+1} = \mathbf{y}^k + \pi^k \Delta \mathbf{y}^k, \tag{8}$$

where $\pi^k > 0$ is a positive step-size. In (8), $\Delta \mathbf{y}^k$ denotes the primal Newton direction, which is the solution to the following linear equation system obtained by deriving the Karush-Kuhn-Tucker (KKT) system of the second-order approximation of $f_\mu(\mathbf{y})$ [7, 8]:

$$\begin{bmatrix} \mathbf{H}_k & \mathbf{M}^T \\ \mathbf{M} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \Delta \mathbf{y}^k \\ \mathbf{w}^k \end{bmatrix} = - \begin{bmatrix} \nabla f_\mu(\mathbf{y}^k) \\ \mathbf{0} \end{bmatrix}, \tag{9}$$

where $\mathbf{H}_k \triangleq \nabla^2 f_\mu(\mathbf{y}^k) \in \mathbb{R}^{(L+1)F \times (L+1)F}$ is the Hessian matrix of $f_\mu(\mathbf{y})$ at $\mathbf{y}^k$, and the vector $\mathbf{w}^k \in \mathbb{R}^{(N-1)F+1}$ contains the dual variables for the constraint $\mathbf{M}\mathbf{y} = \mathbf{e}_{\bar{n}}^{(\bar{n})}$ at the k-th iteration. Here, the entries in $\mathbf{w}^k$ are arranged as $[(\mathbf{w}_k^{(1)})^T, \ldots, (\mathbf{w}_k^{(F)})^T, w_k]^T$, where w_k is the dual variable associated with the time-sharing constraint $\sum_{i=1}^{I} t_i = 1$, and $\mathbf{w}_k^{(f)}$ is in the form of

$$\mathbf{w}_k^{(f)} \triangleq \left[w_1^{(f)}, \ldots, w_{\mathrm{Dst}(f)-1}^{(f)}, w_{\mathrm{Dst}(f)+1}^{(f)}, \ldots, w_N^{(f)}\right]^T. \tag{10}$$

Note that in (10), we have dropped the iteration index k within $[\cdot]$ for notational simplicity. For the same reason, in the rest of the paper, the iteration index k will be dropped whenever such an omission does not cause confusion. Also, we let $w_{\mathrm{Dst}(f)}^{(f)} \equiv 0$, for all f. It can be readily verified that the coefficient matrix of the linear equation in (9) is nonsingular. Therefore, the primal Newton direction $\Delta \mathbf{y}^k$ and the dual variables $\mathbf{w}^k$ can be uniquely determined by solving (9). However, solving for $\Delta \mathbf{y}^k$ and $\mathbf{w}^k$ simultaneously via (9) requires global information and is difficult to be decentralized.

The first key step towards designing a distributed Newton's method is to solve (9) in an *alternative* fashion as follows:

$$\Delta \mathbf{y}^k = -\mathbf{H}_k^{-1}(\nabla f_\mu(\mathbf{y}^k) + \mathbf{M}^T \mathbf{w}^k), \tag{11}$$

$$\mathbf{w}^k = (\mathbf{M}\mathbf{H}_k^{-1}\mathbf{M}^T)^{-1}(-\mathbf{M}\mathbf{H}_k^{-1}\nabla f_\mu(\mathbf{y}^k)). \tag{12}$$

Hence, given $\mathbf{y}^k$, we can first compute $\mathbf{w}^k$ from (12). With $\mathbf{w}^k$, we can solve for $\Delta \mathbf{y}^k$ from (11). Then, $\Delta \mathbf{y}^k$ can be used in (8) (along with an appropriate step-size π^k) to determine the next primal

feasible solution $\mathbf{y}^{k+1}$. However, as we shall see later, computing $\mathbf{H}_k^{-1}$ and $(\mathbf{M}\mathbf{H}_k^{-1}\mathbf{M}^T)^{-1}$ (which is the Laplacian matrix [19] weighted by $\mathbf{H}_k^{-1}$) remains difficult due to the *non-separable* structure of $\mathbf{H}_k$ and requires global information. This is in stark contrast to those optimization problems for wireline networks [9–11], where the Hessian matrices are (block) diagonal and their distributed inversion computations are much easier.

4.2 The Structure of the Hessian Matrix

To see the coupled and non-separable structure of $\mathbf{H}_k$, we evaluate the first and second partial derivatives of $f_\mu(\mathbf{y})$, for which the non-zero ones are:

$$\frac{\partial f_\mu}{\partial s_f} = -\mu U_f'(s_f) - \frac{1}{s_f}, \qquad \frac{\partial^2 f_\mu}{\partial s_f^2} = -\mu U_f''(s_f) + \frac{1}{s_f^2},$$

$$\frac{\partial f_\mu}{\partial x_l^{(f)}} = \frac{1}{\delta_l} - \frac{1}{x_l^{(f)}}, \qquad \frac{\partial^2 f_\mu}{\partial (x_l^{(f)})^2} = \frac{1}{\delta_l^2} + \frac{1}{(x_l^{(f)})^2},$$

$$\frac{\partial^2 f_\mu}{\partial x_l^{(f_1)} \partial x_l^{(f_2)}} = \frac{1}{\delta_l^2} \qquad \frac{\partial f_\mu}{\partial t_i} = -\sum_{l=1}^{L} \left(\frac{C_l^{(i)}}{\delta_l} \right) - \frac{1}{t_i},$$

$$\frac{\partial^2 f_\mu}{\partial t_i^2} = \sum_{l=1}^{L} \frac{(C_l^{(i)})^2}{\delta_l^2} + \frac{1}{t_i^2}, \qquad \frac{\partial^2 f_\mu}{\partial t_{i_1} \partial t_{i_2}} = \sum_{l=1}^{L} \frac{C_l^{(i_1)} C_l^{(i_2)}}{\delta_l^2},$$

$$\frac{\partial^2 f_\mu}{\partial x_l^{(f)} \partial t_i} = -\frac{C_l^{(i)}}{\delta_l^2},$$

where $\delta_l \triangleq \sum_{i=1}^{I} t_i C_l^{(i)} - \sum_{f=1}^{F} x_l^{(f)}$ represents the *unused link capacity* of link l. For convenience, we use a vector

$$\mathbf{c}_l \triangleq [C_l^{(1)}, \ldots, C_l^{(I)}]^T \in \mathbb{R}^I \tag{13}$$

to group the capacity values of the l-th link in each of the I link rate vectors. We further define the following matrices:

$$\mathbf{S} \triangleq \mathrm{Diag}\left\{ -\mu U_1''(s_1) + \frac{1}{s_1^2}, \ldots, -\mu U_F''(s_F) + \frac{1}{s_F^2} \right\}, \tag{14}$$

$$\mathbf{X}_l \triangleq \mathrm{Diag}\left\{ \frac{1}{(x_l^{(1)})^2}, \ldots, \frac{1}{(x_l^{(F)})^2} \right\} + \frac{1}{\delta_l^2} \mathbf{1}_F \mathbf{1}_F^T, \quad \forall l, \tag{15}$$

$$\mathbf{C}_l \triangleq -\frac{1}{\delta_l^2} \mathbf{1}_F \mathbf{c}_l^T, \quad \forall l, \tag{16}$$

$$\mathbf{T} \triangleq \mathrm{Diag}\left\{ \frac{1}{t_1^2}, \ldots, \frac{1}{t_I^2} \right\} + \sum_{l=1}^{L} \frac{1}{\delta_l^2} \mathbf{c}_l \mathbf{c}_l^T. \tag{17}$$

Then, it can be verified that the Hessian matrix $\mathbf{H}_k$ has the following "arrow-head" structure:

$$\mathbf{H}_k = \left[\begin{array}{c:ccc} \mathbf{S} & & & \\ \hdashline & \mathbf{X}_1 & & \mathbf{C}_1 \\ & & \ddots & \vdots \\ & & & \mathbf{X}_L \quad \mathbf{C}_L \\ & \mathbf{C}_1^T & \cdots & \mathbf{C}_L^T \quad \mathbf{T} \end{array}\right]. \tag{18}$$

Remark 1. *It is insightful to compare the structure of* $\mathbf{H}_k$ *with that in [11]. Due to the absence of the time-sharing component, the Hessian in [11, Section V-C] is block diagonal and exactly the same as the principal submatrix* $\mathrm{Diag}\{\mathbf{S}, \mathbf{X}_1, \ldots, \mathbf{X}_L\}$ *in (18). In this paper, however, the coupling between the routing* x*-variables and the time-sharing* t*-variables yields two non-zero "bands" consisting of the* $\mathbf{C}_l$*-matrices. Thus, the block diagonal structure is destroyed in (18) after incorporating the time-sharing component. Hence,* $\mathbf{H}_k^{-1}$ *needs not be sparse even though* $\mathbf{H}_k$ *itself has certain sparsity structure. Also, due to the "arrow-head" structure in (18), finding closed-form expressions for* $\mathbf{H}_k^{-1}$ *is intractable in general. Fortunately, as will be shown in later sections, the arrow-head structure still provides some unique features that can be exploited to iteratively and distributedly compute* $\mathbf{H}_k^{-1}$*. Furthermore, under an interesting special case, this arrow-head structure leads to closed-form expressions for* $\mathbf{H}_k^{-1}$ *and reveals many interesting networking insights.*

4.3 Distributed Computation of the Primal Newton Direction

In Section 4.3.1, we first consider a special collision-based network setting in which each pair of links in the network are mutually interfered. It turns out that in this special network setting, we are able to derive a closed-form expression for $\mathbf{H}_k^{-1}$, which further leads to a *fully distributed* computational scheme for the primal Newton directions. Next, in Section 4.3.2, we propose a matrix-splitting based technique for computing the primal Newton direction for general wireless network settings. This iterative approach circumvents the difficulty of directly computing $\mathbf{H}_k^{-1}$ and also leads to a fully distributed computational scheme.

4.3.1 Closed-Form Expressions for the Primal Newton Direction: A Special Network Setting

We now consider a special collision-based network setting where each pair of links are mutually interfered. This special setting is interesting in that it is a relevant model for many collision-based protocols within one common interference domain (e.g., CSMA networks, cellular uplinks/downlinks, etc.). In this network setting, due to the pairwise mutual interference relationship, it is clear that there can only be L feasible schedules (i.e., $I = L$), each of which has only one active link. In this

case, without loss of generality, we can redefine $\mathbf{c}_l$ in (13) as $\mathbf{c}_l = [0 \cdots 1 \cdots 0]^T \in \mathbb{R}^L$, where the non-zero entry "1" appears at the l-th position[1]. With this new notion of $\mathbf{c}_l$, we have

$$\mathbf{T} = \text{Diag}\left\{\frac{1}{t_1^2}, \ldots, \frac{1}{t_L^2}\right\} + \sum_{l=1}^{L} \frac{1}{\delta_l^2}\mathbf{c}_l\mathbf{c}_l^T = \text{Diag}\left\{\left(\frac{1}{t_1^2} + \frac{1}{\delta_1^2}\right), \ldots, \left(\frac{1}{t_1^2} + \frac{1}{\delta_1^2}\right)\right\}. \tag{19}$$

We point out that the diagonal structure of $\mathbf{T}$ in (19) will play a critical role in deriving the closed-form expressions for $\mathbf{H}_k^{-1}$ and the primal Newton direction.

The key idea to compute $\mathbf{H}_k^{-1}$ is to rewrite $\mathbf{H}_k$ in a *decomposition structure* to enable the application of the Sherman–Morrison–Woodbury (SMW) matrix inversion formula [7]. For this purpose, we define two new vectors as follows:

$$\widetilde{\mathbf{e}}_l \triangleq [\mathbf{0}_F^T, \ldots, \mathbf{1}_F^T, \ldots, \mathbf{0}_F^T, \mathbf{0}_L^T]^T \in \mathbb{R}^{(L+1)F+L},$$
$$\widetilde{\mathbf{c}}_l \triangleq [\mathbf{0}_F^T, \ldots, \mathbf{0}_F^T, \mathbf{c}_l^T]^T \in \mathbb{R}^{(L+1)F+L}.$$

Then, it can be readily verified that the "arrow head" structure of $\mathbf{H}_k$ in (18) can be decomposed into a block diagonal matrix with a rank-$2L$ update:

$$\mathbf{H}_k = \begin{bmatrix} \mathbf{S} & & & & \\ & \mathbf{X}_1 & & & \\ & & \ddots & & \\ & & & \mathbf{X}_L & \\ & & & & \mathbf{T} \end{bmatrix} + \sum_{l=1}^{L}\left(-\frac{1}{\delta_l^2}\right)\widetilde{\mathbf{e}}_l\widetilde{\mathbf{c}}_l^T + \sum_{l=1}^{L}\left(-\frac{1}{\delta_l^2}\right)\widetilde{\mathbf{c}}_l\widetilde{\mathbf{e}}_l^T, \tag{20}$$

where the rank-L updates $\sum_{l=1}^{L}\left(-\frac{1}{\delta_l^2}\right)\widetilde{\mathbf{e}}_l\widetilde{\mathbf{c}}_l^T$ and $\sum_{l=1}^{L}\left(-\frac{1}{\delta_l^2}\right)\widetilde{\mathbf{c}}_l\widetilde{\mathbf{e}}_l^T$ yield the vertical and horizontal bands in (18), respectively. For convenience, we define the following matrices:

$$\mathbf{D} \triangleq \text{Diag}\left\{\mathbf{S}, \mathbf{X}_1, \ldots, \mathbf{X}_L, \mathbf{T}\right\},$$
$$\mathbf{U} \triangleq \left[\frac{\widetilde{\mathbf{e}}_1}{-\delta_1}, \cdots, \frac{\widetilde{\mathbf{e}}_L}{-\delta_L}, \frac{\widetilde{\mathbf{c}}_1}{-\delta_1}, \cdots, \frac{\widetilde{\mathbf{c}}_L}{-\delta_L}\right],$$
$$\mathbf{V} \triangleq \left[\frac{\widetilde{\mathbf{c}}_1}{\delta_1}, \cdots, \frac{\widetilde{\mathbf{c}}_L}{\delta_L}, \frac{\widetilde{\mathbf{e}}_1}{\delta_1}, \cdots, \frac{\widetilde{\mathbf{e}}_L}{\delta_L},\right]^T.$$

Then, Eq. (20) can be compactly written as $\mathbf{H}_k = \mathbf{D} + \mathbf{UV}$. From the SMW matrix inversion formula, we have

$$\mathbf{H}_k^{-1} = (\mathbf{D} + \mathbf{UV})^{-1} = \mathbf{D}^{-1} - \mathbf{D}^{-1}\mathbf{U}(\mathbf{I} + \mathbf{VD}^{-1}\mathbf{U})^{-1}\mathbf{VD}^{-1}. \tag{21}$$

Now, we consider the computation of each term in the right-hand-side (RHS) of (21). First, thanks to the *block diagonal structure* of $\mathbf{D}$, we have that $\mathbf{D}^{-1}$ is simply $\text{Diag}\left\{\mathbf{S}^{-1}, \mathbf{X}_1^{-1}, \ldots, \mathbf{X}_L^{-1}, \mathbf{T}^{-1}\right\}$.

[1]For simplicity, we have assumed that each active link has a unit link capacity. We note that the extension to cases with arbitrary positive link capacity values is straightforward, but at the expense of more complex notation.

Due to the similar structure of $\mathbf{S}$ and $\mathbf{X}_l$ as in [11], we immediately obtain the results given below for computing $\mathbf{S}^{-1}$ and $\mathbf{X}_l^{-1}$. Lemma 4.1 is a direct consequence of the diagonal structure of $\mathbf{S}$. The proof of Lemma 4.2 is based on an observation of the decomposable structure of $\mathbf{X}_l$ and the use of the SMW matrix inversion formula. We refer readers to [22] for the proof details.

Lemma 4.1 (Distributedness of computing $\mathbf{S}^{-1}$)**.** *The matrix $\mathbf{S}^{-1}$ can be computed in a distributed fashion (source node-wise) as* $\mathbf{S}^{-1} = \text{Diag}\{\frac{1}{-\mu U_1''(s_1)+1/s_1^2}, \ldots, \frac{1}{-\mu U_F''(s_F)+1/s_F^2}\}$.

Lemma 4.2 (Distributed closed-form expression for $\mathbf{X}_l^{-1}$)**.** *The entries of $\mathbf{X}_l^{-1}$ can be computed distributedly (link-wise) in closed-form as follows:*

$$(\mathbf{X}_l^{-1})_{f_1 f_2} = \begin{cases} \left(x_l^{(f_1)}\right)^2 \left(1 - \frac{\left(x_l^{(f_1)}\right)^2}{\|\widehat{\mathbf{x}}_l\|^2}\right), & \text{if } 1 \le f_1 = f_2 \le F, \\ -\frac{\left(x_l^{(f_1)} x_l^{(f_2)}\right)^2}{\|\widehat{\mathbf{x}}_l\|^2}, & \text{if } 1 \le f_1 \ne f_2 \le F. \end{cases}$$

Similar to Lemma 4.1, due to the diagonal structure of $\mathbf{T}$, the following result is obvious:

Lemma 4.3 (Distributed closed-form expression for $\mathbf{T}^{-1}$)**.** *The entries of $\mathbf{T}^{-1}$ can be computed distributed (link-wise) in closed-form as:* $\mathbf{T}^{-1} = \text{Diag}\left\{\frac{t_1^2\delta_1^2}{t_1^2+\delta_1^2}, \ldots, \frac{t_L^2\delta_L^2}{t_L^2+\delta_L^2}\right\}$.

Next, consider the inverse of $\mathbf{I} + \mathbf{V}\mathbf{D}^{-1}\mathbf{U}$ in (21), which is in the form known as the Schur complement [23]. By exploiting the special structure in $\mathbf{I} + \mathbf{V}\mathbf{D}^{-1}\mathbf{U}$, we establish the following important result:

Theorem 4.4. *The inverse of $(\mathbf{I} + \mathbf{V}\mathbf{D}^{-1}\mathbf{U})$ has the following block-wise structure:*

$$(\mathbf{I} + \mathbf{V}\mathbf{D}^{-1}\mathbf{U})^{-1} = \left[\begin{array}{c|c} \mathbf{Q}_1 & \mathbf{Q}_2 \\ \hline \mathbf{Q}_3 & \mathbf{Q}_1 \end{array}\right], \tag{22}$$

where $\mathbf{Q}_i$, $i = 1, \ldots, 3$ are all diagonal matrices:

$$\mathbf{Q}_1 = \text{Diag}\left\{\frac{\delta_l^2(t_l^2+\delta_l^2)}{\delta_l^2(t_l^2+\delta_l^2) - t_l^2(\mathbf{1}^T\mathbf{X}_l^{-1}\mathbf{1})},\ l = 1, \ldots, L\right\} \tag{23}$$

$$\mathbf{Q}_2 = \text{Diag}\left\{\frac{t_l^2\delta_l^2}{\delta_l^2(t_l^2+\delta_l^2) - t_l^2(\mathbf{1}^T\mathbf{X}_l^{-1}\mathbf{1})},\ l = 1, \ldots, L\right\} \tag{24}$$

$$\mathbf{Q}_3 = \text{Diag}\left\{\frac{(t_l^2+\delta_l^2)(\mathbf{1}^T\mathbf{X}_l^{-1}\mathbf{1})}{\delta_l^2(t_l^2+\delta_l^2) - t_l^2(\mathbf{1}^T\mathbf{X}_l^{-1}\mathbf{1})},\ l = 1, \ldots, L\right\}. \tag{25}$$

The proof of Theorem 4.4 is based on the blockwise matrix inversion theorem [23], with the application of the sparsity structure of vectors $\widetilde{\mathbf{e}}_l$, $\widetilde{\mathbf{c}}_l$ and the diagonal structure of $\mathbf{T}$. The proof details are relegated to Appendix B. We point out that, for notational convenience, we do not expand $\mathbf{1}^T\mathbf{X}_l^{-1}\mathbf{1}$ in (23)–(25). This term can be easily calculated in closed-form and distributedly

(link-wise) by noting that $\mathbf{1}^T\mathbf{X}_l^{-1}\mathbf{1} = \sum_{f_1=1}^{F}\sum_{f_1=1}^{F}(\mathbf{X}_l^{-1})_{f_1f_2}$ and that $(\mathbf{X}_l^{-1})_{f_1f_2}$ can be calculated in closed-form (c.f. Lemma 4.2).

By applying Theorem 4.4 in the SMW formula and after some algebraic derivations, we derive the following structural and closed-form expression for $\mathbf{H}_k^{-1}$. To this end, we define the following quantities:

$$R_{l,1} = \frac{t_l^2\delta_l^4/\|\hat{\mathbf{x}}_l\|^2}{\delta_l^2(t_l^2+\delta_l^2) - t_l^2(\mathbf{1}^T\mathbf{X}_l^{-1}\mathbf{1})}, \tag{26}$$

$$R_{l,2} = \frac{t_l^2\delta_l^2}{t_l^2+\delta_l^2} + \frac{t_l^4\delta_l^2(\mathbf{1}^T\mathbf{X}_l^{-1}\mathbf{1})/(t_l^2+\delta_l^2)}{\delta_l^2(t_l^2+\delta_l^2) - t_l^2(\mathbf{1}^T\mathbf{X}_l^{-1}\mathbf{1})}. \tag{27}$$

Theorem 4.5. *$\mathbf{H}_k^{-1}$ has the same "arrow head" structure as in $\mathbf{H}_k$, i.e.,*

$$\mathbf{H}_k^{-1} = \left[\begin{array}{c:ccc} \widehat{\mathbf{S}} & & & \\ \hdashline & \widehat{\mathbf{X}}_1 & & \widehat{\mathbf{C}}_1 \\ & & \ddots & \vdots \\ & & \widehat{\mathbf{X}}_L & \widehat{\mathbf{C}}_L \\ & \widehat{\mathbf{C}}_1^T & \cdots \quad \widehat{\mathbf{C}}_L^T & \widehat{\mathbf{T}} \end{array}\right], \tag{28}$$

where $\widehat{\mathbf{S}}$, $\widehat{\mathbf{X}}_l$, $\widehat{\mathbf{C}}_l$, and $\widehat{\mathbf{T}}$ can be expressed in closed-form as follows:

$$\widehat{\mathbf{S}} = \mathrm{Diag}\left\{\frac{1}{-\mu U_1''(s_1)+1/s_1^2}, \ldots, \frac{1}{-\mu U_F''(s_F)+1/s_F^2}\right\}, \tag{29}$$

$$(\widehat{\mathbf{X}}_l)_{f_1f_2} = \begin{cases} \left(x_l^{(f_1)}\right)^2\left[1-(1-R_{l,1})\frac{\left(x_l^{(f_1)}\right)^2}{\|\hat{\mathbf{x}}_l\|^2}\right], & \text{if } 1 \le f_1 = f_2 \le F, \\ -(1-R_{l,1})\frac{\left(x_l^{(f_1)}x_l^{(f_2)}\right)^2}{\|\hat{\mathbf{x}}_l\|^2}, & \text{if } 1 \le f_1 \ne f_2 \le F, \end{cases} \tag{30}$$

$$(\widehat{\mathbf{C}}_l)_{fl_1} = \begin{cases} R_{l,1}(x_l^{(f)})^2, & \text{if } l_1 = l,\ f\text{=}1,\ldots,F, \\ 0, & \text{otherwise}, \end{cases} \tag{31}$$

$$\widehat{\mathbf{T}} = \mathrm{Diag}\left\{R_{l,2},\ l = 1,\ldots,L\right\}. \tag{32}$$

We relegate the proof of Theorem 4.5 to Appendix C. Finally, by applying Theorem 4.5, the primal Newton direction can be computed distributedly as stated in the following theorem:

Theorem 4.6. *In the special wireless network setting, given dual variables $\widetilde{\mathbf{w}}$, the Newton direction Δs_f, $\Delta x_l^{(f)}$, Δt_l for each source rate s_f, link flow rate $x_l^{(f)}$, and link time-sharing variable t_l can be*

computed using local information at each source node s and link l, respectively, as follows:

$$\Delta s_f = \frac{s_f\big(\mu s_f U_f'(s_f) + 1 - s_f w^{(f)}_{\mathrm{Src}(f)}\big)}{1 - \mu s_f^2 U_f''(s_f)}, \quad \forall f, \tag{33}$$

$$\Delta x_l^{(f)} = \big(x_l^{(f)}\big)^2 \left[\left(1 - (1-R_{l,1})\frac{(x_l^{(f)})^2}{\|\widehat{\mathbf{x}}_l\|^2}\right)\left(\frac{1}{x_l^{(f)}} - \frac{1}{\delta_l} + w^{(f)}_{\mathrm{Tx}(l)} - w^{(f)}_{\mathrm{Rx}(l)}\right) + R_{l,1}(x_l^{(f)})^2\left(\frac{1}{t_l} - \frac{1}{\delta_l} - w\right)\right.$$
$$\left. - \sum_{f'=1, f'\neq f}^{F} (1-R_{l,1})\frac{(x_l^{(f')})^2}{\|\widehat{\mathbf{x}}_l\|^2}\left(\frac{1}{x_l^{(f')}} - \frac{1}{\delta_l} + \widetilde{w}^{(f')}_{\mathrm{Tx}(l)} - \widetilde{w}^{(f')}_{\mathrm{Rx}(l)}\right)\right], \quad l, f. \tag{34}$$

$$\Delta t_l = R_{l,1}\left[\sum_{f=1}^{F}(x_l^{(f)})^2\left(\frac{1}{x_l^{(f)}} - \frac{1}{\delta_l} + w^{(f)}_{\mathrm{Tx}(l)} - w^{(f)}_{\mathrm{Rx}(l)}\right)\right] + R_{l,2}\left(\frac{1}{t_l} - \frac{1}{\delta_l} - w\right). \tag{35}$$

The basic idea of proving Theorem 4.6 is to apply Theorem 4.5 and exploit the second-order properities of $\mathbf{a}_l^{(f)}$ and $\mathbf{b}^{(f)}$ (cf. [11, Section V-B]) to simplify the result. We relegate the proof details to Appendix D.

Remark 2. *An important remark for Theorem 4.6 is in order. Theorem 4.6 not only provides a closed-form expression for a distributed computation of the primal Newton direction, but also offers an interesting network interpretation. Here, we can think of the difference of the dual variables $(w^{(f)}_{\mathrm{Tx}(l)} - w^{(f)}_{\mathrm{Rx}(l)})$ in (34) as "the queue length difference" in the back-pressure algorithm, although $w_n^{(f)}$ itself cannot be exactly interpreted as queue length (since it can be positive or negative). Note that in (34), it can be shown that $\left(1 - (1-R_{l,1})\frac{(x_l^{(f)})^2}{\|\widehat{\mathbf{x}}_l\|^2}\right)$ and $\sum_{f'=1,f'\neq f}^{F}(1-R_{l,1})\frac{(x_l^{(f')})^2}{\|\widehat{\mathbf{x}}_l\|^2}$ are all positive. Hence, if the positive $(w^{(f)}_{\mathrm{Tx}(l)} - w^{(f)}_{\mathrm{Rx}(l)})$–values outweigh the negative ones, i.e., the "pressure" on the transmitter side of link l is greater than the "pressure" on the receiver side, then $x_l^{(f)}$ will be increased in the next iteration. Note also that, unlike first-order methods, the decision to increase or decrease $x_l^{(f)}$ considers not only the "pressure difference" of flow f, but also the "pressure difference" from other flows at link l (via an appropriate weighting scheme as evident in (34)).*

4.3.2 A Matrix-Splitting Based Iterative Scheme for Computing the Primal Newton Direction: General Network Settings

In Section 4.3.1, we have used an SMW-based approach to derive a closed-form expression for $\mathbf{H}_k^{-1}$, which, through (11), further led to a fully distributed computational scheme for the primal Newton direction. It is worth pointing out that, for general wireless network settings, the SMW-based approach usually fails. This is because in general wireless networks, each pair of links do not necessarily mutually interfere with each other. As a result, the $\mathbf{c}_l$-vectors may not be orthogonal to each other. Thus, in the Schur complement $(\mathbf{I} + \mathbf{V}\mathbf{D}^{-1}\mathbf{U})$, the submatrix block $\mathbf{K}_1$ (cf. Appendix B) becomes a dense matrix, which yields little special structure to exploit, and makes the computation

of $(\mathbf{I}+\mathbf{V}\mathbf{D}^{-1}\mathbf{U})^{-1}$ equally difficult comparing to that of $\mathbf{H}_k^{-1}$ itself. To circumvent the trouble of computing $\mathbf{H}_k^{-1}$ directly, we now propose a matrix-splitting based technique to compute the primal Newton direction.

A brief overview of matrix-splitting. Simply speaking, matrix-splitting is a generic framework for solving linear equation systems iteratively [24]. Consider a consistent linear equation system $\mathbf{F}\mathbf{z}=\mathbf{d}$, where $\mathbf{F}\in\mathbb{R}^{n\times n}$ is nonsingular and $\mathbf{z},\mathbf{d}\in\mathbb{R}^n$. Split $\mathbf{F}$ as $\mathbf{F}=\mathbf{F}_1-\mathbf{F}_2$, where $\mathbf{F}_1$ is also nonsingular. Clearly, there are multiple choices in splitting $\mathbf{F}$. A good splitting strategy, however, is to choose an $\mathbf{F}_1$ that is easier to invert than $\mathbf{F}$. Then, it follows that $\mathbf{z}=(\mathbf{F}_1^{-1}\mathbf{F}_2)\mathbf{z}+\mathbf{F}_1^{-1}\mathbf{d}$. Now, let $\mathbf{z}^0$ be an arbitrary starting vector and consider the following iterative scheme [24]:

$$\mathbf{z}^{k+1}=(\mathbf{F}_1^{-1}\mathbf{F}_2)\mathbf{z}^k+\mathbf{F}_1^{-1}\mathbf{d},\quad k\geq 0. \tag{36}$$

It can be shown that the iterative scheme in (36) is convergent to the unique solution $\mathbf{z}=\mathbf{F}^{-1}\mathbf{b}$ if and only if $\rho(\mathbf{F}_1^{-1}\mathbf{F}_2)<1$, where $\rho(\cdot)$ represents the spectral radius of a matrix. The following result provides a sufficient condition for $\rho(\mathbf{F}_1^{-1}\mathbf{F}_2)<1$ (see [10, 24] for more details).

Lemma 4.7. *Suppose that* $\mathbf{F}$ *is a real symmetric matrix. If both matrices* $\mathbf{F}_1+\mathbf{F}_2$ *and* $\mathbf{F}_1-\mathbf{F}_2$ *are positive definite, then* $\rho(\mathbf{F}_1^{-1}\mathbf{F}_2)<1$.

Lemma 4.7 suggests that checking the convergence of a given matrix splitting scheme can be translated into checking the positive definiteness of matrices. To this end, the following result provides a convenient sufficient condition for checking positive definiteness by using diagonal dominance:

Lemma 4.8. *If a symmetric matrix* $\mathbf{Q}$ *is strictly diagonally dominant, i.e.,* $|(\mathbf{Q})_{ii}|>\sum_{j\neq i}|(\mathbf{Q})_{ij}|$, *and if* $(\mathbf{Q})_{ii}>0$ *for all* i, *then* $\mathbf{Q}$ *is positive definite.*

Lemma 4.8 is an immediate consequence of the well-known Gershgorin circle theorem (see [23, Corollary 7.2.3] for more details).

Matrix-splitting scheme for computing the primal Newton direction. Now, we are ready to use the matrix splitting scheme in (36) to compute $\mathbf{w}^k$. But before deriving the details of the matrix splitting scheme, we would like to point out that due to the block diagonal structure between the $\mathbf{S}$-block and the rest of $\mathbf{H}_k^{-1}$ (cf. 18), we are still able to compute the primal Newton direction Δs_f in *closed-form* by (33) even in general wireless network settings. As a result, the matrix-splitting scheme only needs to compute $\Delta x_l^{(f)}$ and Δt_i. We let $\overline{\mathbf{H}}_k$ denote the submatrix block obtained by removing $\mathbf{S}$ and its associated rows and columns from $\mathbf{H}_k$, i.e.,

$$\overline{\mathbf{H}}_k=\begin{bmatrix}\mathbf{X}_1 & & & \mathbf{C}_1\\ & \ddots & & \mathbf{C}_2\\ & & \mathbf{X}_L & \mathbf{C}_L\\ \mathbf{C}_1^T & \cdots & \mathbf{C}_L^T & \mathbf{T}\end{bmatrix}. \tag{37}$$

We further define the following matrices:

$$\mathbf{\Lambda}_k = \mathrm{Diag}\left\{\mathrm{diag}\left\{\overline{\mathbf{H}}_k\right\}\right\}, \tag{38}$$

$$\mathbf{\Omega}_k = \overline{\mathbf{H}}_k - \mathbf{\Lambda}_k. \tag{39}$$

Further, let $\overline{\mathbf{\Omega}}_k$ be a diagonal matrix with diagonal entries given by $(\overline{\mathbf{\Omega}}_k)_{ii} = \sum_j |(\mathbf{\Omega}_k)_{ij}|$, i.e., the 1-norm of the non-diagonal row entries in $\mathbf{\Omega}_k$. Then, we have the following result:

Lemma 4.9. *Let $\overline{\mathbf{H}}_k$ be split as $\overline{\mathbf{H}}_k = (\mathbf{\Lambda}_k + \alpha\overline{\mathbf{\Omega}}_k) - (\alpha\overline{\mathbf{\Omega}}_k - \mathbf{\Omega}_k)$, where $\alpha > \frac{1}{2}$ is a parameter for tuning convergence speed. Then, the following sequence $\{\Delta\mathbf{y}_m^k\}$, $m = 1, 2, \ldots$, generated by*

$$\Delta\mathbf{y}_{m+1}^k = (\mathbf{\Lambda}_k + \alpha\overline{\mathbf{\Omega}}_k)^{-1}(\alpha\overline{\mathbf{\Omega}}_k - \mathbf{\Omega}_k)\Delta\mathbf{y}_m^k + (\mathbf{\Lambda}_k + \alpha\overline{\mathbf{\Omega}}_k)^{-1}(-\nabla f_\mu(\mathbf{y}^k) - \mathbf{M}^T\mathbf{w}^k) \tag{40}$$

converges to $\Delta\mathbf{y}^k$ in (11) as $m \to \infty$.

The key to proving Lemma 4.9 is to verify that both the sum and difference of the two components in the splitting scheme are strictly diagonally dominant. We relegate the proof details to Appendix E.

Remark 3. *Several remarks regarding the matrix splitting scheme in Lemma 4.9 are in order. First, Lemma 4.9 is inspired by, and is also a generalization of, the matrix splitting scheme in [10]. The basic idea of both splitting schemes is to construct a diagonal nonsingular matrix ($\mathbf{\Lambda}_k + \alpha\overline{\mathbf{\Omega}}_k$ in our case) for which the inverse can be easily computed by each link. However, our matrix splitting scheme differs from that in [10] in the following aspects: i) Since $(\mathbf{\Lambda}_k + \alpha\overline{\mathbf{\Omega}}_k)$ is not element-wise non-negative (c.f. [10]), the matrix $\overline{\mathbf{\Omega}}_k$ in this work requires a different definition compared to that in [10], which also leads to a more subtle proof; ii) Our splitting scheme is parameterized by α, which enables convergence speed tuning in (40), while the scheme in [10] can be viewed as a special case of our scheme with $\alpha = 1$.*

We now make several comments on the choice of the parameter α. From (36), it is clear that the approximation error of the sequence $\{\Delta\mathbf{y}_m^k\}_{m=1,2,\ldots}$ decreases at a rate of $O(\rho(\mathbf{F}_1^{-1}\mathbf{F}_2)^m)$. Hence, a smaller value of $\rho(\mathbf{F}_1^{-1}\mathbf{F}_2)$ implies a faster convergence speed of the iterative scheme. To this end, we have the following result for the selection of α:

Lemma 4.10. *Consider two alternative matrix splitting schemes with parameters α_1 and α_2, respectively, satisfying $\frac{1}{2} < \alpha_1 \leq \alpha_2$. Let ρ_{α_1} and ρ_{α_2} be their spectral radii, respectively. Then, $\rho_{\alpha_1} \leq \rho_{\alpha_2}$.*

The proof of Lemma 4.10 is based on the comparison theorem [24, Theorem 2.3] and we relegate the proof details to Appendix F. Lemma 4.10 suggests that, in order to make the matrix splitting scheme converge faster, we should choose a smaller α, i.e., we can let $\alpha = \frac{1}{2} + \epsilon$, where $\epsilon > 0$ is small. It is worth pointing out that $\alpha > \frac{1}{2}$ is only a sufficient condition for positive definiteness, and hence

convergence. In practical implementation, we could choose $\alpha \leq \frac{1}{2}$ for even faster convergence speed as long as $\rho((\mathbf{\Lambda}_k + \alpha\overline{\mathbf{\Omega}}_k)^{-1}(\alpha\overline{\mathbf{\Omega}}_k - \mathbf{\Omega}_k)) < 1$ holds.

Next, we show that the matrix splitting scheme in Lemma 4.9 can indeed be implemented in a distributed fashion to compute the primal Newton direction, where the result is stated in the following theorem:

Theorem 4.11. *In general wireless settings, let $\Delta x_l^{(f)}(m)$, and $\Delta t_i(m)$ denote the the values of $\Delta x_l^{(f)}$ and Δt_i in the m-th iteration, respectively. Given dual variables $\mathbf{w}^k$, the Newton directions $\Delta x_l^{(f)}$ and Δt_i can be iteratively computed using local information at each node s and link l, respectively, as follows:*

$$\Delta x_l^{(f)}(m+1) = \frac{1}{P_{l,1}^{(f)}}\left[P_{l,2}^{(f)} + \left(\frac{1}{x_l^{(f)}} - \frac{1}{\delta_l} + w_{\mathrm{Tx}(l)}^{(f)} - w_{\mathrm{Rx}(l)}^{(f)}\right)\right], \tag{41}$$

$$\Delta t_l(m+1) = \frac{1}{Q_{i,1}}\left[Q_{i,2} + \left(\frac{1}{t_i} + \sum_{l=1}^{L}\left(\frac{C_l^{(i)}}{\delta_l}\right)\right) - w\right], \tag{42}$$

where $P_{l,1}^{(f)}$, $P_{l,2}^{(f)}$, $Q_{i,1}$, and $Q_{i,2}$ are, respectively, defined as:

$$P_{l,1}^{(f)} = \frac{1}{\delta_l^2}\left[1 + \alpha\left((F-1) + \sum_{i=1}^{I} C_l^{(i)}\right)\right] + \frac{1}{(x_l^{(f)})^2}, \tag{43}$$

$$P_{l,2}^{(f)} = \frac{1}{\delta_l^2}\left[\alpha\left(F - 1 + \sum_{i=1}^{I} C_l^{(i)}\right)\Delta x_l^{(f)}(m) - \sum_{f'=1,\neq f}^{F} \Delta x_l^{(f')}(m) + \sum_{i=1}^{I} C_l^{(i)}\Delta t_i(m)\right], \tag{44}$$

$$Q_{i,1} = \frac{1}{t_i^2} + \sum_{l=1}^{L}\frac{1}{\delta_l^2}\left[(C_l^{(i)})^2 + \alpha C_l^{(i)}\left(F + \sum_{i'=1,\neq i}^{I} C_l^{(i')}\right)\right], \tag{45}$$

$$Q_{i,2} = \frac{1}{\delta_l^2}\left[\sum_{l=1}^{L} C_l^{(i)}\left(\sum_{f=1}^{F}\left(\Delta x_l^{(f)}(m) + \alpha F\right) + \sum_{i'=1,\neq i}^{I} C_l^{(i')}(\alpha\Delta t_i(m) - \Delta t_{i'}(m))\right)\right]. \tag{46}$$

Theorem 4.11 can be proved by using the "arrow head" structure of $\overline{\overline{\mathbf{H}}}_k$ and the second-order properties of $\mathbf{a}_l$ and $\widetilde{\mathbf{b}}^{(f)}$ (cf. [11, Section V-B]) to compute the element-wise expansion of (40). We relegate the proof details to Appendix G.

Remark 4. *There are several remarks pertaining to Theorem 4.11. First, it can be seen from (43), (44), (45), and (46) that all the information needed to update $\Delta x_l^{(f)}$ are locally available at link l. This confirms that the matrix splitting scheme for the primal Newton direction $\Delta x_l^{(f)}$ can be distributedly implemented. Second, similar to the special network setting, the term $w_{\mathrm{Tx}(l)}^{(f)} - w_{\mathrm{Rx}(l)}^{(f)}$ is also involved in (41). Thus, the "back-pressure" type of network interpretation in the special case (see Remark 2) also applies to general wireless network settings. Lastly, from (45) and (46), it can be seen that we*

need to collect all primal Newton direction information in the current iteration (i.e., $\Delta x_l^{(f)}(m)$ and $\Delta t_i(m)$) in order to compute the time-sharing update direction $\Delta t_i(m+1)$. Later in Section 4.5, we will discuss the mechanism to facilitate this information sharing.

4.4 Distributed Computation of the Dual Variables

Recall that the dual variables can be computed by solving the linear equation system in (12), i.e.,

$$\mathbf{w}^k = (\mathbf{M}\mathbf{H}_k^{-1}\mathbf{M}^T)^{-1}(-\mathbf{M}\mathbf{H}_k^{-1}\nabla f(\mathbf{y}^k)).$$

However, due to their complex structures, computing $(\mathbf{M}\mathbf{H}_k^{-1}\mathbf{M}^T)^{-1}$ and $(-\mathbf{M}\mathbf{H}_k^{-1}\nabla f(\mathbf{y}^k))$ also require global information and are difficult to implement in a distributed fashion. In what follows, we will again propose to use the matrix-splitting technique in a *embedded* fashion to compute $(\mathbf{M}\mathbf{H}_k^{-1}\mathbf{M}^T)^{-1}$ and $(-\mathbf{M}\mathbf{H}_k^{-1}\nabla f(\mathbf{y}^k))$ for general wireless network settings and show that they can also be implemented in a *distributed* fashion. Then, for the special network setting in Section 4.3.1, we will show that the matrix-splitting scheme for computing $(\mathbf{M}\mathbf{H}_k^{-1}\mathbf{M}^T)^{-1}$ and $(-\mathbf{M}\mathbf{H}_k^{-1}\nabla f(\mathbf{y}^k))$ can further be significantly simplified based on the closed-form structure of $\mathbf{H}_k^{-1}$.

4.4.1 General Wireless Network Settings

In general wireless network settings, the closed-form expression for $\mathbf{H}_k^{-1}$ is not available. Thus, the matrix-splitting technique in Section 4.3.2 cannot be applied directly and we need to first compute $\mathbf{M}\mathbf{H}_k^{-1}\mathbf{M}^T$ as well as $-\mathbf{M}\mathbf{H}_k^{-1}\nabla f(\mathbf{y}^k)$. We will first focus on $\mathbf{M}\mathbf{H}_k^{-1}\mathbf{M}^T$, and then extend the obtained results to $-\mathbf{M}\mathbf{H}_k^{-1}\nabla f_\mu(\mathbf{y}^k)$ due to their similar structures. Note that $\mathbf{M}\mathbf{H}_k^{-1}\mathbf{M}^T$ can be decomposed as:

$$\mathbf{M}\mathbf{H}_k^{-1}\mathbf{M}^T = \widetilde{\mathbf{B}}_0\mathbf{S}^{-1}\widetilde{\mathbf{B}}_0^T + \widehat{\mathbf{M}}\overline{\mathbf{H}}_k^{-1}\widehat{\mathbf{M}}^T, \tag{47}$$

where $\widetilde{\mathbf{B}}_0$ and $\widehat{\mathbf{M}}$ is defined as

$$\widetilde{\mathbf{B}}_0 \triangleq [\widetilde{\mathbf{B}}^T, \mathbf{0}_F]^T, \quad \text{and} \quad \widehat{\mathbf{M}} \triangleq \left[\begin{array}{ccc:c} \mathbf{A}_1 & \cdots & \mathbf{A}_L & \\ \hdashline & & & \mathbf{1}_I^T \end{array}\right]. \tag{48}$$

Due to the diagonal structure of $\widetilde{\mathbf{B}}_0$ and $\mathbf{S}^{-1}$, the term $\widetilde{\mathbf{B}}_0\mathbf{S}^{-1}\widetilde{\mathbf{B}}_0^T$ in (47) can be computed in a distributed fashion at each source node. The term $\widehat{\mathbf{M}}\overline{\mathbf{H}}_k^{-1}\widehat{\mathbf{M}}^T$ is more involved due to the lack of a closed-form expression of $\overline{\mathbf{H}}_k^{-1}$. Fortunately, a closer look at $\widehat{\mathbf{M}}\overline{\mathbf{H}}_k^{-1}\widehat{\mathbf{M}}^T$ reveals that it can be further separated into two steps: i) compute $\mathbf{Z}_k = \overline{\mathbf{H}}_k^{-1}\widehat{\mathbf{M}}^T$ first; and then ii) compute $\widehat{\mathbf{M}}\mathbf{Z}_k$. Further, we note that $\mathbf{Z}_k = \overline{\mathbf{H}}_k^{-1}\widehat{\mathbf{M}}^T$ can be written as follows:

$$\mathbf{z}_j = \overline{\mathbf{H}}_k^{-1}\hat{\mathbf{m}}_j, \quad j = 1, \ldots, (N-1)F+1, \tag{49}$$

where $\mathbf{z}_j$ and $\hat{\mathbf{m}}_j$ represent the j-th columns in $\mathbf{Z}_k$ and $\widehat{\mathbf{M}}$, respectively. It is important to recognize that (49) is in the same form as in the primal Newton direction update (cf. (11)). Therefore, $\mathbf{z}_j$ can be computed using the same matrix-splitting technique as described in Section 4.3.2. We formally state this result as follows:

Proposition 4.12. *Partition the entries of $\mathbf{z}_j$ into the following blocks corresponding to the arrangement of $\widetilde{\mathbf{y}}$, i.e.,*

$$\mathbf{z}_j = \left[z_j^{(x_1^{(1)})} \cdots z_j^{(x_1^{(F)})} \middle| \cdots\cdots \middle| z_j^{(x_L^{(1)})} \cdots z_j^{(x_L^{(F)})} \middle| z_j^{(t_1)} \cdots z_j^{(t_I)} \right]^T,$$

Let $z_{j,m}^{(x_l^{(f)})}$ and $z_{j,m}^{(t_i)}$ denote the m-th iteration values of $z_j^{(x_l^{(f)})}$ and $z_j^{(t_i)}$, respectively. Then, $\mathbf{z}_j$ can be iteratively computed by using the following matrix-splitting scheme:

$$z_{j,m+1}^{(x_l^{(f)})} = \frac{1}{P_{l,1}^{(f)}} \left[P_{l,3}^{(f)}(j) + P_{l,4}^{(f)}(j) \right], \quad \forall l, f, \tag{50}$$

$$z_{j,m+1}^{t_i} = \frac{1}{Q_{i,1}} \left[Q_{i,3}(j) + Q_4(j) \right], \quad \forall i, \tag{51}$$

where $P_{l,1}^{(f)}$ and $Q_{i,1}$ are as defined in (43) and (45), respectively; and where $P_{l,3}^{(f)}$, $P_{l,4}^{(f)}$, $Q_{i,3}$, and $Q_{i,4}$ can be computed as:

$$P_{l,3}^{(f)}(j) = \frac{1}{\delta_l^2} \left[\alpha \left(F - 1 + \sum_{i=1}^{I} C_l^{(i)} \right) z_{j,m}^{(x_l^{(f)})} - \sum_{f'=1,\neq f}^{F} z_{j,m}^{(x_l^{(f')})} + \sum_{i=1}^{I} C_l^{(i)} z_{j,m}^{(t_i)} \right], \tag{52}$$

$$P_{l,4}^{(f)}(j) = \begin{cases} 1 & \text{if } \mathrm{Tx}(l) = j - (f-1)(N-1), \\ -1 & \text{if } \mathrm{Rx}(l) = j - (f-1)(N-1), \\ 0 & \text{otherwise}, \end{cases} \tag{53}$$

$$Q_{i,3}(j) = \frac{1}{\delta_l^2} \left[\sum_{l=1}^{L} C_l^{(i)} \left(\sum_{f=1}^{F} \left(z_{j,m}^{x_l^{(f)}} + \alpha F \right) + \sum_{i'=1,\neq i}^{I} C_l^{(i')} (\alpha z_{j,m}^{t_i} - z_{j,m}^{t_{i'}}) \right) \right], \tag{54}$$

$$Q_4(j) = \begin{cases} 1 & \text{if } j = F(N-1) + 1, \\ 0 & \text{otherwise}. \end{cases} \tag{55}$$

The proof of Proposition 4.12 is similar to that of Theorem 4.11 and by noting the special sparsity structure of $\widehat{\mathbf{M}}$. We regelate the proof details to Appendix H. In the second step, i.e., $\widehat{\mathbf{M}}\overline{\mathbf{H}}_k^{-1}\widehat{\mathbf{M}}^T = \widehat{\mathbf{M}}\mathbf{Z}_k$, by exploiting the special sparsity structure of $\widehat{\mathbf{M}}$, we obtain the following result, and we relegate the proof details to Appendix I:

Proposition 4.13. *Let $(\widehat{\mathbf{M}}\mathbf{Z}_k)_{j_1j_2}$ be the entry in the j_1-th row and j_2-column of $\widehat{\mathbf{M}}\mathbf{Z}_k$. Then, $(\widehat{\mathbf{M}}\mathbf{Z}_k)_{j_1j_2}$ can be distributedly computed as*

$$(\widehat{\mathbf{M}}\mathbf{Z}_k)_{j_1j_2} = \begin{cases} \sum_{l\in\mathcal{O}(n)} z_{j_2}^{(x_l^{(f)})} - \sum_{l\in\mathcal{I}(n)} z_{j_2}^{(x_l^{(f)})}, & \text{if } j_1 = (f-1)(N-1)+n,\ n \neq \beta_f(\mathrm{Src}(f)), \\ & j_2 = (f'-1)(N-1)+n', \\ z_{j_2}^{(s_f)} + \sum_{l\in\mathcal{O}(n)} z_{j_2}^{(x_l^{(f)})} - \sum_{l\in\mathcal{I}(n)} z_{j_2}^{(x_l^{(f)})} & \text{if } j_1 = (f-1)(N-1)+n,\ n \neq \beta_f(\mathrm{Src}(f)), \\ & j_2 = (f'-1)(N-1)+n', \\ \sum_{i=1}^{I} z_{j_2}^{(t_i)}, & \text{if } j_1 = (N-1)F+1,\ j_2 = 1,\ldots,(N-1)F+1. \end{cases} \tag{56}$$

Remark 5. *First, from (50) and (51), it can be seen that since the $\mathbf{z}_j$-variable can be obtained by using the same matrix-splitting scheme as in (41) and (42), they can be computed along with the primal Newton directions in (41) and (42) by sharing $P_{l,1}^{(f)}$ and $Q_{i,1}$, thus saving significant computing resources.*

Next, we consider the procedure to compute the term $-\mathbf{M}\mathbf{H}_k^{-1}\nabla f_\mu(\mathbf{y}^k)$. Similarly, it is easy to recognize that it can be decomposed into

$$-\mathbf{M}\mathbf{H}_k^{-1}\nabla f_\mu(\mathbf{y}^k) = \widetilde{\mathbf{B}}_0\mathbf{S}^{-1}(-\nabla_{\mathbf{s}} f_\mu(\mathbf{y}^k)) + \widehat{\mathbf{M}}\overline{\mathbf{H}}_k^{-1}(-\nabla_{\mathbf{x},\mathbf{t}} f_\mu(\mathbf{y}^k)),$$

where $\nabla_{\mathbf{s}} f_\mu(\mathbf{y}^k)$ and $\nabla_{\mathbf{x},\mathbf{t}} f_\mu(\mathbf{y}^k)$ represent the partial derivatives with respect to the s_f-variables and the remaining variables, respectively. Due to the diagonal structure of $\widetilde{\mathbf{B}}_0$ and $\mathbf{S}^{-1}$, the vector $\widetilde{\mathbf{B}}_0\mathbf{S}^{-1}(-\nabla_{\mathbf{s}} f_\mu(\mathbf{y}^k))$ can be distributedly computed at each source node. For computing $\widehat{\mathbf{M}}\overline{\mathbf{H}}_k^{-1}(-\nabla_{\mathbf{x},\mathbf{t}} f_\mu(\mathbf{y}^k))$, we can also separate this into two parts as: i) compute $\mathbf{g} = \overline{\mathbf{H}}_k^{-1}(-\nabla_{\mathbf{x},\mathbf{t}} f_\mu(\mathbf{y}^k))$ and ii) compute $\widehat{\mathbf{M}}\mathbf{g}$. Therefore, we have the following result:

Proposition 4.14. *Partition the entries of $\mathbf{g}$ into the following blocks corresponding to the arrangement of $\widetilde{\mathbf{y}}$, i.e.,*

$$\mathbf{g} = \left[g^{(x_1^{(1)})}\cdots g^{(x_1^{(F)})}\middle|\cdots\cdots\middle|g^{(x_L^{(1)})}\cdots g^{(x_L^{(F)})}\middle|g^{(t_1)}\cdots g^{(t_I)}\right]^T,$$

Let $g_m^{(x_l^{(f)})}$ and $g_m^{(t_i)}$ denote the m-th iteration values of $g^{(x_l^{(f)})}$ and $g^{(t_i)}$, respectively. Then, $\mathbf{g}$ can be iteratively computed by using the following matrix-splitting scheme:

$$g_{m+1}^{(x_l^{(f)})} = \frac{1}{P_{l,1}^{(f)}}\left[P_{l,5}^{(f)} + \left(\frac{1}{x_l^{(f)}} - \frac{1}{\delta_l}\right)\right], \quad \forall l, f, \tag{57}$$

$$g_{m+1}^{t_i} = \frac{1}{Q_{i,1}}\left[Q_{i,5} + \left(\frac{1}{t_i} + \sum_{l=1}^{L}\left(\frac{C_l^{(i)}}{\delta_l}\right)\right)\right], \quad \forall i, \tag{58}$$

where $P_{l,1}^{(f)}$ *and* $Q_{i,1}$ *are as defined in (43) and (45), respectively; and where* $P_{l,5}^{(f)}$ *and* $Q_{i,5}$ *are computed as:*

$$P_{l,5}^{(f)} = \frac{1}{\delta_l^2}\left[\alpha\left(F-1+\sum_{i=1}^{I} C_l^{(i)}\right) g_m^{(x_l^{(f)})} - \sum_{f'=1,\neq f}^{F} g_m^{(x_l^{(f')})} + \sum_{i=1}^{I} C_l^{(i)} g_m^{(t_i)}\right], \tag{59}$$

$$Q_{i,5} = \frac{1}{\delta_l^2}\left[\sum_{l=1}^{L} C_l^{(i)}\left(\sum_{f=1}^{F}\left(g_m^{x_l^{(f)}} + \alpha F\right) + \sum_{i'=1,\neq i}^{I} C_l^{(i')}(\alpha g_m^{t_i} - g_m^{t_{i'}})\right)\right]. \tag{60}$$

Proposition 4.15. *Let* $(\widehat{\mathbf{M}}\mathbf{g})_j$ *be the entry in the* j*-th entry of* $\widehat{\mathbf{M}}\mathbf{g}$*. Then,* $(\widehat{\mathbf{M}}\mathbf{g})_j$ *can be distributedly computed as*

$$(\widehat{\mathbf{M}}\mathbf{g})_j = \begin{cases} \sum_{l\in\mathcal{O}(n)} g^{(x_l^{(f)})} - \sum_{l\in\mathcal{I}(n)} g^{(x_l^{(f)})}, & \text{if } j=(f-1)(N-1)+n,\ n\neq\beta_f(\mathrm{Src}(f)), \\ g^{(s_f)} + \sum_{l\in\mathcal{O}(n)} g^{(x_l^{(f)})} - \sum_{l\in\mathcal{I}(n)} g^{(x_l^{(f)})} & \text{if } j=(f-1)(N-1)+n,\ n=\beta_f(\mathrm{Src}(f)), \\ \sum_{i=1}^{I} g^{(t_i)}, & \text{if } j=(N-1)F+1. \end{cases} \tag{61}$$

The proofs of Propositions 4.14 and 4.15 follow the same reasoning as that for Propositions 4.12 and 4.13. Thus, we omit their proofs for brevity. By putting together Propositions 4.12–4.15, we can distributedly compute $\widehat{\mathbf{M}}\overline{\mathbf{H}}_k^{-1}\widehat{\mathbf{M}}^T$ and $(-\widehat{\mathbf{M}}\overline{\mathbf{H}}_k^{-1}\nabla_{\mathbf{x},\mathbf{t}} f_\mu(\mathbf{y}^k))$ without the explicit knowledge of $\mathbf{H}_k^{-1}$.

Finally, with the obtained $\widehat{\mathbf{M}}\overline{\mathbf{H}}_k^{-1}\widehat{\mathbf{M}}^T$ and $(-\widehat{\mathbf{M}}\overline{\mathbf{H}}_k^{-1}\nabla f_{\mathbf{x}}(\mathbf{y}^k))$, we can again use matrix-splitting to compute the dual variables $\mathbf{w}^k$. Similar to the computation of the primal Newton direction for the general network setting, we define the following matrices:

$$\mathbf{\Pi}_k = \mathrm{Diag}\left\{\mathrm{diag}\left\{\widehat{\mathbf{M}}\overline{\mathbf{H}}_k^{-1}\widehat{\mathbf{M}}^T\right\}\right\}, \tag{62}$$

$$\mathbf{\Psi}_k = \widehat{\mathbf{M}}\overline{\mathbf{H}}_k^{-1}\widehat{\mathbf{M}}^T - \mathbf{\Pi}_k. \tag{63}$$

Also, let $\overline{\mathbf{\Psi}}$ be the diagonal matrix with diagonal entries given as $(\overline{\mathbf{\Psi}})_{ii} = \sum_j |(\overline{\mathbf{\Psi}})_{ij}|$. Then, adopting the same approach as in previous discussions, we have the following:

Proposition 4.16. *Let* $\widehat{\mathbf{M}}\overline{\mathbf{H}}_k^{-1}\widehat{\mathbf{M}}^T$ *be split as* $\widehat{\mathbf{M}}\overline{\mathbf{H}}_k^{-1}\widehat{\mathbf{M}}^T = (\mathbf{\Pi}_k + \alpha\overline{\mathbf{\Psi}}_k) - (\alpha\overline{\mathbf{\Psi}}_k - \mathbf{\Psi}_k)$*, where* $\alpha > \frac{1}{2}$ *is a parameter for tuning convergence speed. Then, the following sequence* $\{\mathbf{w}_m^k\}$*,* $m = 1, 2, \ldots$*, generated by*

$$\mathbf{w}_{m+1}^k = (\mathbf{\Pi}_k + \alpha\overline{\mathbf{\Psi}}_k)^{-1}(\alpha\overline{\mathbf{\Psi}}_k - \mathbf{\Psi}_k)\mathbf{w}_m^k + (\mathbf{\Pi}_k + \alpha\overline{\mathbf{\Psi}}_k)^{-1}(-\mathbf{M}\mathbf{H}_k^{-1}\nabla f_\mu(\mathbf{y}^k)) \tag{64}$$

converges to $\mathbf{w}^k$ *in (12) as* $m \to \infty$*.*

Note that since $(\mathbf{\Pi}_k + \alpha\overline{\mathbf{\Psi}}_k)$ is diagonal and due to the *node-based local structure* in $-\mathbf{M}\mathbf{H}_k^{-1}\nabla f_\mu(\mathbf{y}^k)$ (cf. (61)), it can be easily inverted in a distributed fashion at each node using only local information.

4.4.2 The Special Wireless Network Setting

It is worth pointing out that, for the special wireless network setting in Section 4.3.1, there is no need to use the matrix-splitting scheme to compute $\mathbf{M}\mathbf{H}_k^{-1}\mathbf{M}^T$ and $(-\mathbf{M}\mathbf{H}_k^{-1}\nabla f(\mathbf{y}^k))$. Thanks to the closed-form expression of $\mathbf{H}_k^{-1}$ given by Theorem 4.5, the computation of $\mathbf{M}\mathbf{H}_k^{-1}\mathbf{M}^T$ and $(-\mathbf{M}\mathbf{H}_k^{-1}\nabla f(\mathbf{y}^k))$ can be significantly simplified as follows:

$$\mathbf{M}\mathbf{H}_k^{-1}\mathbf{M}^T = \left[\begin{array}{c|c} \widetilde{\mathbf{B}}\widehat{\mathbf{S}}\widetilde{\mathbf{B}}^T + \sum_{l=1}^L \mathbf{A}_l\widehat{\mathbf{X}}_l\mathbf{A}_l^T & \left(\sum_{l=1}^L \mathbf{A}_l\widehat{\mathbf{C}}_l\right)\mathbf{1} \\ \hline \mathbf{1}^T\left(\sum_{l=1}^L \widehat{\mathbf{C}}_l^T\mathbf{A}_l^T\right) & \mathbf{1}^T\widehat{\mathbf{T}}\mathbf{1} \end{array}\right], \tag{65}$$

$$\mathbf{M}\mathbf{H}_k^{-1}\nabla f_\mu(\mathbf{y}^k) = -\left[\begin{array}{c} \widetilde{\mathbf{B}}\widehat{\mathbf{S}}\nabla_{\mathbf{s}} f_\mu(\mathbf{y}^k) + \sum_{l=1}^L \mathbf{A}_l(\widehat{\mathbf{X}}_l\nabla_{\mathbf{x}_l} f_\mu(\mathbf{y}^k) + \widehat{\mathbf{C}}_l\nabla_{\mathbf{t}} f_\mu(\mathbf{y}^k)) \\ \hline \mathbf{1}^T\left(\sum_{l=1}^L \widehat{\mathbf{C}}_l^T\nabla_{\mathbf{x}_l} f_\mu(\mathbf{y}^k) + \widehat{\mathbf{T}}\nabla_{\mathbf{t}} f_\mu(\mathbf{y}^k)\right) \end{array}\right], \tag{66}$$

where $\nabla_{\mathbf{s}} f_\mu(\mathbf{y}^k)$, $\nabla_{\mathbf{x}_l} f_\mu(\mathbf{y}^k)$, and $\nabla_{\mathbf{t}} f_\mu(\mathbf{y}^k)$ are gradient components with respect to $\mathbf{s} \triangleq [s_1, \ldots, s_F]^T$, $\mathbf{x}_l \triangleq [x_l^{(1)}, \ldots, x_l^{(F)}]^T$, and $\mathbf{t} \triangleq [t_1, \ldots, t_L]^T$, respectively. We note that in (65), the block $\widetilde{\mathbf{B}}\widehat{\mathbf{S}}\widetilde{\mathbf{B}}^T + \sum_{l=1}^L \mathbf{A}_l\widehat{\mathbf{X}}_l\mathbf{A}_l^T$ is the same as in [11, Eq. (19)]. Therefore, the same structural property in [11, Theorem 7] also appears in this block. The block $\left(\sum_{l=1}^L \mathbf{A}_l\widehat{\mathbf{C}}_l\right)\mathbf{1}$ is an $(N-1)\times F$-dimensional vector, which, due to the special sparsity structure of $\mathbf{A}_l$, can be distributedly computed. The block $\mathbf{1}^T\left(\sum_{l=1}^L \widehat{\mathbf{C}}_l^T\mathbf{A}_l^T\right)$ is the transpose of $\left(\sum_{l=1}^L \mathbf{A}_l\widehat{\mathbf{C}}_l\right)\mathbf{1}$. The block $\mathbf{1}^T\widehat{\mathbf{T}}\mathbf{1}$ is a scalar and, due to the diagonal structure of $\widehat{\mathbf{T}}$, can be easily computed. Based on these structural properties and the matrix-splitting scheme in Proposition 4.16, we can explicitly derive the dual variable update. To this end, we define two types of link sets as follows:

$$\Phi(n) \triangleq \mathcal{I}(n) \cup \mathcal{O}(n), \qquad \Psi(n,f) \triangleq \{l \in \mathcal{I}(n) \cup \mathcal{O}(n) : \mathrm{Tx}(l) = \mathrm{Dst}(f) \text{ or } \mathrm{Rx}(l) = \mathrm{Dst}(f)\}.$$

Let $\mathbb{1}_S(a)$ denote the set indicator function, which takes value 1 if $a \in S$ and 0 otherwise. Then, with some algebraic manipulations, we obtain the following result:

Theorem 4.17. *Given a primal solution $\mathbf{y}^k$, the update of the dual variable $w_n^{(f)}$ can be computed using local information at each node. More specifically, $w_n^{(f)}$ can be written as*

$$w_n^{(f)}(m+1) = \frac{1}{U_n^f(m)}(V_{n,1}^{(f)}(m) + V_{n,2}^{(f)}(m) - W_n^f(m)), \tag{67}$$

$$w(m+1) = \frac{1}{\rho(m)}[\sigma(m) - \tau(m)], \tag{68}$$

where $U_n^{(f)}(m)$, $V_{n,1}^{(f)}(m)$, $V_{n,2}^{(f)}(m)$, $W_n^{(f)}(m)$, $\rho(m)$, $\sigma(m)$, *and* $\tau(m)$ *are, respectively, defined as*

$$U_n^{(f)}(m) \triangleq \begin{cases} \sum_{l\in\Phi(n)}[1+\alpha(1-\mathbb{1}_{\Psi(n,f)}(l))](x_l^{(f)})^2\Big(1-\frac{R_{l,1}(x_l^{(f)})^2}{\|\widehat{\mathbf{x}}_l\|^2}\Big)+ \\ \quad \sum_{f'=1,\neq f}^{F}\Big(\sum_{l\in\Psi(n,f')}\frac{\alpha R_{l,1}(x_l^{(f)}x_l^{(f')})^2}{\|\widehat{\mathbf{x}}_l\|^2}\Big)+\alpha\sum_{l\in\Phi(n)}|R_{l,2}|(x_l^{(f)})^2 & \text{if } n\neq \mathrm{Src}(f), \\ \sum_{l\in\Phi(n)}[1+\alpha(1-\mathbb{1}_{\Psi(n,f)}(l))](x_l^{(f)})^2\Big(1-\frac{R_{l,1}(x_l^{(f)})^2}{\|\widehat{\mathbf{x}}_l\|^2}\Big)+ \\ \quad \sum_{f'=1,\neq f}^{F}\Big(\sum_{l\in\Psi(n,f')}\frac{\alpha R_{l,1}(x_l^{(f)}x_l^{(f')})^2}{\|\widehat{\mathbf{x}}_l\|^2}\Big)+\alpha\sum_{l\in\Phi(n)}|R_{l,2}|(x_l^{(f)})^2 \\ \quad +\frac{1}{-\mu U_f''(s_f)+\frac{1}{(s_f)^2}} & \text{if } n=\mathrm{Src}(f), \end{cases} \tag{69}$$

$$V_{n,1}^{(f)}(m) \triangleq \sum_{l\in\mathcal{I}(n)}(x_l^{(f)})^2\Big(1-\frac{R_{l,1}(x_l^{(f)})^2}{\|\widehat{\mathbf{x}}_l\|^2}\Big)(\widetilde{w}_{\mathrm{Tx}(l)}^{(f)}-\alpha\widetilde{w}_{\mathrm{Rx}(l)}^{(f)})+$$
$$\sum_{l\in\mathcal{O}(n)\setminus\Psi(n,f)}(x_l^{(f)})^2\Big(1-\frac{R_{l,1}(x_l^{(f)})^2}{\|\widehat{\mathbf{x}}_l\|^2}\Big)(\widetilde{w}_{\mathrm{Rx}(l)}^{(f)}-\alpha\widetilde{w}_{\mathrm{Tx}(l)}^{(f)})-$$
$$\sum_{f'=1,\neq f}^{F}\Big(\sum_{l\in\Psi(n,f')}\frac{\alpha R_{l,1}(x_l^{(f)}x_l^{(f')})^2}{\|\widehat{\mathbf{x}}_l\|^2}\Big)\widetilde{w}_n^f+\alpha\sum_{l\in\Phi(n)}|R_{l,2}|(x_l^{(f)})^2\tilde{w}, \tag{70}$$

$$V_{n,2}^{(f)}(m) \triangleq \sum_{f'=1,\neq f}^{F}\Big(\Big(\sum_{l\in\mathcal{O}(n)}\frac{R_{l,2}(x_l^{(f)}x_l^{(f')})^2}{\|\widehat{\mathbf{x}}_l\|^2}-\sum_{l\in\mathcal{I}(n)}\frac{R_{l,2}(x_l^{(f)}x_l^{(f')})^2}{\|\widehat{\mathbf{x}}_l\|^2}\Big)(\widetilde{w}_{\mathrm{Tx}(l)}^{(f')}-\widetilde{w}_{\mathrm{Rx}(l)}^{(f')})\Big) \tag{71}$$
$$-\alpha\sum_{l\in\Phi(n)}R_{l,2}(x_l^{(f)})^2, \tag{72}$$

$$W_n^{(f)}(m) \triangleq \begin{cases} \sum_{l\in\mathcal{O}(n)}(x_l^{(f)}-\frac{(x_l^{(f)})^2}{\delta_l})-\sum_{l\in\mathcal{I}(n)}\frac{R_{l,2}}{\|\hat{x}_l\|^2}(x_l^{(f)})^2\sum_{f'=1}^{F}(x_l^{(f')}-\frac{(x_l^{(f')})^2}{\delta_l}) \\ \quad -(\sum_{l\in\mathcal{O}(n)}R_{l,2}(x_l^{(f)})^2(\frac{1}{\delta_l}+\frac{1}{t_l})-\sum_{l\in\mathcal{I}(n)}R_{l,2}(x_l^{(f)})^2(\frac{1}{\delta_l}+\frac{1}{t_l})), & \text{if } n\neq\mathrm{Src}(f), \\ \sum_{l\in\mathcal{O}(n)}(x_l^{(f)}-\frac{(x_l^{(f)})^2}{\delta_l})-\sum_{l\in\mathcal{I}(n)}\frac{R_{l,2}}{\|\hat{x}_l\|^2}(x_l^{(f)})^2\sum_{f'=1}^{F}(x_l^{(f')}-\frac{(x_l^{(f')})^2}{\delta_l}) \\ \quad +\frac{s_f(1+\mu s_f U_f'(s_f))}{\mu s_f^2 U_f''(s_f)-1} \\ \quad -(\sum_{l\in\mathcal{O}(n)}R_{l,2}(x_l^{(f)})^2(\frac{1}{\delta_l}+\frac{1}{t_l})-\sum_{l\in\mathcal{I}(n)}R_{l,2}(x_l^{(f)})^2(\frac{1}{\delta_l}+\frac{1}{t_l})), & \text{if } n=\mathrm{Src}(f), \end{cases} \tag{73}$$

$$\rho(m) \triangleq \sum_{l=1}^{L}|R_{l,3}|+\alpha\Big[\sum_{f=1}^{F}\sum_{\substack{n=1,\\ n\neq\mathrm{Dst}(f)}}^{N}|R_{l,2}|(x_l^{(f)})^2\Big], \tag{74}$$

$$\sigma(m) \triangleq \sum_{l=1}^{L}R_{l,3}+\sum_{f=1}^{F}\sum_{\substack{n=1\\ n\neq\mathrm{Dst}(f)}}^{N}(R_{l,2}w_n^{(f)}+\alpha|R_{l,2}|\tilde{w})(x_l^{(f)})^2, \tag{75}$$

$$\tau(m) \triangleq \sum_{l=1}^{L} \left[R_{l,3} \left(\frac{1}{t_l} + \frac{1}{\delta_l} \right) + \sum_{f=1}^{F} R_{l,2} \left(x_l^{(f)} - \frac{(x_l^{(f)})^2}{\delta_l} \right) \right]. \tag{76}$$

Theorem 4.17 can be proved by following the same approach as in [11, Theorem 10] with some modifications to incorporate the augmented blocks in (65) and (66). We relegate the proof details to Appendix J.

Remark 6. *There are several observations worth noting in regard to Theorem 4.17. First, from (69), (70), (71), and (73), we can see that all the information needed to update $w_n^{(f)}$ are either locally available at node n or at links incident to node n. This again illustrates that the splitting scheme can be distributedly implemented. Second, it can be verified that $V_{n,1}^{(f)}$ involves "same-session" as well as "cross-session" quadratic terms of $x_l^{(f)}$ coming into and going out of node n. This is starkly different from the dual update scheme in the subgradient method in that all flow-related quantities here are of second-order and weighted by $w_{\mathrm{Tx}(l)}^{(f)} + w_{\mathrm{Rx}(l)}^{(f)}$ (when $\alpha = 1$), which can be loosely interpreted as "total queue length" on both sides of l (see Remark 2). $V_{n,2}^{(f)}$ involves a "back-pressure" weighting mechanism ($w_{\mathrm{Tx}(l)}^{(f)} - w_{\mathrm{Rx}(l)}^{(f)}$). But unlike $V_{n,1}^{(f)}$, the second-order quantities in $V_{n,2}^{(f)}$ are only related to "cross-session" flow products $x_l^{(f)} x_l^{(f')}$. Third, although the dual update scheme within a second-order method is more complex at each node, the more rapid convergence rate of a second-order method, with its accompanying less information exchange, can outweigh this local computational cost increase.*

4.5 Implementation of the Distributed Newton's Method

So far, we have derived the major components of our proposed distributed computational scheme for obtaining the primal Newton direction and performing the dual variable updates, which are the key elements of our proposed distributed Newton's method. There are, however, several open issues that remain to be studied for practical implementation, namely, the scale of information exchange, stopping criterion, step-size selection, etc.

4.5.1 Information Exchange Scale Analysis

We first consider the required information exchange in the primal Newton direction update. From Theorem 4.6, we can see that to compute Δs_f, we need s_f and $w_{\mathrm{Src}(f)}^{(f)}$. Since s_f is available at $\mathrm{Src}(f)$, we can see from Proposition 4.16 and Theorem 4.17 that $w_{\mathrm{Src}(f)}^{(f)}$ can also be computed at $\mathrm{Src}(f)$. Hence, there is no need for any information exchange in computing Δs_f.

For $\Delta x_l^{(f)}$, it can be seen from (34) that we need $x_l^{f'}$, $f' = 1, \ldots, F$, $w_{\mathrm{Tx}(l)}^{(f')}$, and $w_{\mathrm{Rx}(l)}^{(f')}$. Clearly, $x_l^{f'}$, $f' = 1, \ldots, F$, are available at link l. From Proposition 4.16 and Theorem 4.17, it can be verified

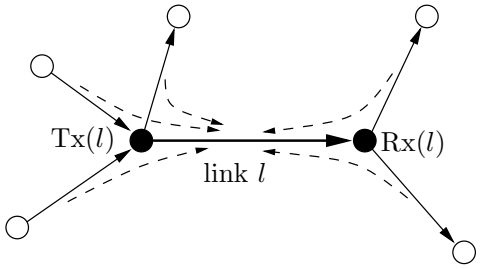


Figure 1: Information exchange for computing $\Delta x_l^{(f)}$, which only requires exchanging information from links one-hop away from link l.

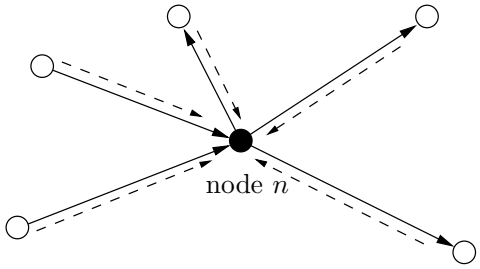


Figure 2: In the special network setting, information exchange for computing $\widetilde{w}_n^{(f)}$, which only requires exchanging information from nodes one-hop away from node n.

that $w_{\mathrm{Tx}(l)}^{(f')}$ and $w_{\mathrm{Rx}(l)}^{(f')}$ can also be computed using flow and dual information with respect to links that share $\mathrm{Tx}(l)$ and $\mathrm{Rx}(l)$. This implies that computing $\Delta x_l^{(f)}$ only requires exchanging information *one-hop* away from link l, as shown in Fig. 1.

Next, we analyze the information exchange scale in updating dual variables. We first consider the special case of Section 4.4.2. From Theorem 4.17, we can see that to compute $w_n^{(f)}$, we need $x_l^{(f')}$, $w_{\mathrm{Tx}(l)}^{(f')}$, and $w_{\mathrm{Rx}(l)}^{(f')}$, where $l \in \Phi(n)$, $f' = 1, \ldots, F$. It is clear that $x_l^{(f')}$, $l \in \Phi(n)$, is readily available at node n. On the other hand, $w_{\mathrm{Tx}(l)}^{(f')}$ and $w_{\mathrm{Rx}(l)}^{(f')}$ are either available at node n itself or are available at nodes one-hop away from node n. This implies that computing $w_n^{(f)}$ only requires exchanging information from nodes *one-hop* away from node n, as shown in Fig. 2.

The analysis of the general case is slightly more involved since (64) cannot be written explicitly in terms of the $\mathbf{x}$-variables and $\mathbf{w}$-variables. From (64), we can see that to compute $\mathbf{\Pi}_k$, $\mathbf{\Psi}_k$, and $\overline{\mathbf{\Psi}}_k$, we need to know each entry of $\mathbf{M}\mathbf{H}_k^{-1}\mathbf{M}^T$, which can be obtained from Proposition 4.13. It can further be seen from Proposition 4.13 that for the entry in the row corresponding to node n and flow f and the column corresponding to node n' and f', information exchange between nodes n and n' is necessary. This implies that, in general network settings, the largest number of hops for information exchange is the diameter of the network graph. Clearly, the increase of exchange hops is the price to pay for optimizing wireless networks with more complicated interference relationships.

Two interesting remarks are in order. First, in the special network setting, although Problem CLO is more complex than the pure flow control problem in [10], the information exchange required for the distributed Newton's algorithm for Problem CLO turns out to be more decentralized than that in [10]. More specifically, the information exchange for Problem CLO is from entities at most one-hop away, while in the pure flow control problem in [10], each source node needs to send information to all the links on its predefined route. This somewhat surprising result can be loosely explained by the fact that by allowing multi-path routing, the routing decision is automatically determined by the

node "pressure" as described in Remark 2 at each node, thus alleviating the burden of exchanging information along the fixed routes. Second, we can see that our distributed Newton's method requires a similar scale of information exchange to that in the subgradient method (cf. Appendix A).

4.5.2 Initialization of the Algorithm

Another open question in the implementation is how to initialize the algorithm. One simple solution is as follows. Each $\mathrm{Src}(f)$ can choose an initial value ϵ_f and equally distribute ϵ_f along all of its outgoing links. The time-sharing variables can be initialized as $t_i = \frac{1}{I}$, $\forall i$. Also, if each intermediate node has multiple outgoing links, then the sum of its incoming traffic will be equally distributed along each outgoing link as well. Clearly, if ϵ_f, $\forall f$, are small enough, then the constraint $\sum_{f=1}^{F} x_l^{(f)} \leq C_l$ can be satisfied. The initial values of dual variables can be chosen arbitrarily since they are unrestricted (e.g., a simple choice is to set $w_n^{(f)} = 1$ if $n \neq \mathrm{Dst}(f)$ and $w_n^{(f)} = 0$ if $n = \mathrm{Dst}(f)$).

4.5.3 Stopping Criterion

Since the Newton's method enjoys a quadratic rate of convergence (under certain regularity conditions [7]), a simple stopping rule would be to let all sources and links run the algorithm for a fixed amount of time. If the time duration is long enough for a given maximum sized network, then due to the rapid convergence speed, by the time the clock expires, it is with high probability that the algorithm will have converged to a near-optimal solution.

A more sophisticated way to stop the algorithm can be based on the so-called Newton decrement [8]. In Newton's methods, for a given primal vector $\mathbf{y}^k$, the Newton decrement is defined as [8]

$$\lambda(\mathbf{y}^k) = \sqrt{(\Delta \mathbf{y}^k)^T \mathbf{H}_k \Delta \mathbf{y}^k}, \tag{77}$$

which measures the decrease in the objective function value at each iteration. Thus, we can use $\lambda(\mathbf{y}^k) \leq \epsilon$ as a stopping criterion, where ϵ is a predefined error tolerance. The following result shows that $\lambda(\mathbf{y}^k)$ can also be computed in a distributed fashion. Again, for ease of notation, we omit the iteration index k.

Proposition 4.18. *The Newton decrement $\lambda(\mathbf{y})$ can be computed as*

$$\lambda(\mathbf{y}) = \Bigg[\sum_{f=1}^{F} (\Delta s_f)^2 \Big(-\mu U_f''(s_f) + \frac{1}{(s_f)^2} \Big) + \sum_{l=1}^{L} \Bigg(\sum_{f=1}^{F} \Big(\frac{\Delta x_l^{(f)}}{x_l^{(f)}} \Big)^2 + \frac{1}{\delta_l} \Big(\sum_{f=1}^{F} \Delta x_l^{(f)} \Big)^2 \Bigg) - 2 \sum_{l=1}^{L} \frac{1}{\delta_l^2} \Big(\sum_{f=1}^{F} \Delta x_l^{(f)} \Big) \Big(\sum_{i=1}^{I} C_l^{(i)} \Delta t_i \Big) + \sum_{i=1}^{I} \Big(\frac{\Delta t_i}{t_i} \Big)^2 + \sum_{l=1}^{L} \frac{1}{\delta_l^2} \Big(\sum_{i=1}^{I} C_l^{(i)} \Delta t_i \Big)^2 \Bigg]^{\frac{1}{2}}. \tag{78}$$

We remark that, for given t_i-values, since (78) is separable with respect to each source node and each link, each source can compute the quantity $(\Delta s_f)^2\Big(-\mu U_f''(s_f)+\frac{1}{(s_f)^2}\Big)$ and each link can compute the quantity $\Big(\frac{\Delta x_l^{(f)}}{x_l^{(f)}}\Big)^2+\frac{1}{\delta_l}\Big(\sum_{f=1}^F \Delta x_l^{(f)}\Big)^2$. Therefore, $\lambda(\widetilde{\mathbf{y}}^k)$ can be computed distributedly using only local information. The proof of Proposition 4.18 is based on the decomposition structure of $\mathbf{H}_k$ and we relegate the proof details to Appendix K.

To compute the Newton decrement, we can see from (78) that each source needs s_f and Δs_f and each link needs $x_l^{(f)}$ and $\Delta x_l^{(f)}$. From earlier discussions on Δs_f, we can conclude that no information exchange is required at each source node. Also, from earlier discussions on $\Delta x_l^{(f)}$, we know that at most one-hop information exchange is required in this regard. However, to allow every source and link to compute the final value of the Newton decrement, every source and link will need to broadcast a packet containing the value of $(\Delta s_f)^2\Big(-\mu U_f''(s_f)+\frac{1}{(s_f)^2}\Big)$ and $\Big(\frac{\Delta x_l^{(f)}}{x_l^{(f)}}\Big)^2+\frac{1}{\delta_l}\Big(\sum_{f=1}^F \Delta x_l^{(f)}\Big)^2$, respectively, to the network. Thus, we can see that the more accurate termination criterion is obtained at the expense of a larger scale of information exchange across the network.

4.5.4 Step-Size Selection

As in the classical Newton's method [7,8], under certain regularity conditions, when the iterates $\{\mathbf{y}^k\}$ approach a close neighborhood of an optimal solution, using a fixed step size $\pi^k = 1$ gives us the so-called *quadratic rate of convergence*, which is very efficient. If the iterates $\{\mathbf{y}^k\}$ are far from an optimal solution (which is also called "damped Newton phase" and can be measured by $\|\nabla f_\mu(\mathbf{y}^k)\|_2$ – see [8] for more details), then some inexact line search methods, such as the Armijo rule [7] (also called "backtracking line search" in [8]) or the step-size rule in [10] can be used. Due to the inexactness of these line search methods, the theoretical convergence rate would be sub-quadratic, but still in theory and practice, the rate would be superlinear, and so, much faster than the linear rate of convergence of subgradient-type methods.

To conclude this section, we summarize our distributed Newton's method for Problem CLO in Algorithm 1.

5 Numerical Results

In this section, we present some numerical results to demonstrate the efficacy of our proposed distributed Newton's method. First, we examine the convergence speed of the parameterized matrix splitting scheme in computing the primal Newton directions and dual variables. We use a 10-node 3-session network as an example. For both primal Newton directions and dual variables, vary α from 0.1 to 1. In both cases, the matrix-splitting scheme is terminated when the error between the true solution of $\Delta\mathbf{y}^k$ in Eq. (11) (resp., $\mathbf{w}^k$ in Eq. (12)) and the matrix-splitting based iterative computa-

Algorithm 1 Distributed Newton's Method for Solving CLO.

Initialization:

1. Each source and link: Choose some appropriate values of s_f, $x_l^{(f)}$, $\forall f$, and t_i, $\forall i$.
2. Each node: Choose appropriate values of dual variables $w_n^{(f)}$, $\forall f$ and w.

Main Iteration:

3. *Primal Newton directions (special setting)*: Update Δs_f, $\Delta x_l^{(f)}$, and Δt_l using (33), (34), and (35) at each source node and link, respectively.
4. *Primal Newton directions (general settings)*: Update Δs_f, $\Delta x_l^{(f)}$, and Δt_l using (33), (41), and (42) at each source node and link, respectively. Meanwhile, compute and store $\mathbf{z}_j$, $\forall j$, using (50) and (51); compute and store $\mathbf{g}$ using (57) and (58).
5. *Dual updates (special setting)*: Update $w_n^{(f)}$ and w using (67) and (68)at each node.
6. *Dual updates (general settings)*: First compute $\widehat{\mathbf{M}}\overline{\mathbf{H}}_k^{-1}\widehat{\mathbf{M}}^T$ using (56) and the $\mathbf{z}$-vectors obtained from Step 4. Also, compute $(-\widehat{\mathbf{M}}\overline{\mathbf{H}}_k^{-1}\nabla_{\mathbf{x},\mathbf{t}} f_\mu(\mathbf{y}^k))$ using (61) and the $\mathbf{g}$-vector obtained from Step 4. Then, update the dual variables $w_n^{(f)}$ and w using (64).
7. Terminate the algorithm if a predefined run-time limit is reached or if the Newton decrement criterion is satisfied. Otherwise, go to Step 3 for the special setting case or to Step 4 for the general setting case.

tion scheme is less than 1×10^{-6}. The approximation errors for primal Newton directions and dual variables are shown in Fig. 3 and Fig. 4, respectively (in log scale). We can see that for all values of α, the error decreases exponentially fast. Also, the smaller the value of α, the faster the convergence speed. More specifically, when $\alpha = 0.1$, the number of iterations is approximately half of that when $\alpha = 1$ (9 vs. 15 in the primal case and 27 vs. 53 in the dual case). This confirms our theoretical analysis of the choice of α in Lemma 4.10.

To illustrate our proposed distributed Newton method, we first study a five-node multi-hop wireless network as shown in Fig. 5. In this network, five nodes are distributed in a square region of $800m \times 800m$. The network is assumed to be operating under the special network setting (i.e., all links are mutually interfered). The capacity of each link is normalized to 1. There are two sessions in the network: N5 to N4 and N1 to N3. We adopt $\log(s_f)$ as our utility function, which represents the so-called proportional fairness [25]. The optimal routing paths for session N5 $\rightarrow$ N4 and N1 $\rightarrow$ N3 are plotted in Figs. 6 and Fig. 7, respectively. In these figures, the symbols $\bigstar$ and $\blacksquare$ denote the source and destination nodes of each session, respectively. The optimal session rates for N5 $\rightarrow$ N4 and N1 $\rightarrow$ N3 are 1.0550 and 1.1175, respectively. The optimal time-sharing solution for scheduling is shown in Table 1. The convergence behavior of the distributed Newton method is illustrated in Fig. 8, which shows the objective values of the approximating and the original problems. It can be

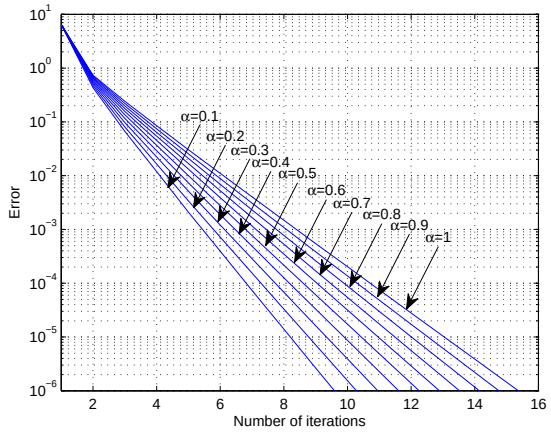


Figure 3: The error between the true solution of $\Delta\mathbf{y}^k$ in Eq. (11) and the matrix-splitting based iterative computational scheme.

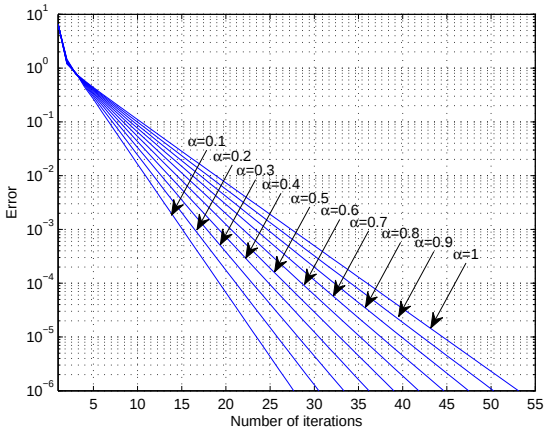


Figure 4: The error between the true solution of $\mathbf{w}^k$ in Eq. (12) and the matrix-splitting based iterative computational scheme.

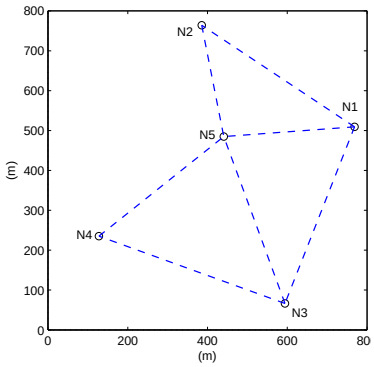


Figure 5: A five-node two-session network.

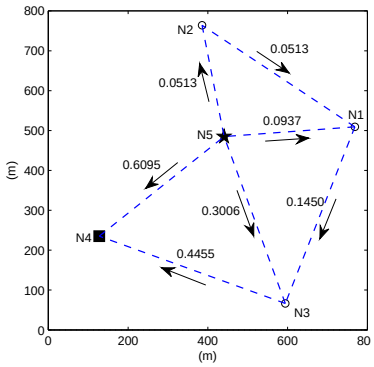


Figure 6: The optimal routing solutions for session N5 → N4.

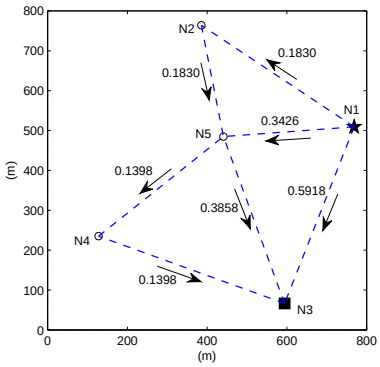


Figure 7: The optimal routing solutions for session N1 → N3.

Table 1: Optimal time-sharing solution for the network in Fig. 5.

t_1	t_2	t_3	t_4	t_5	t_6	t_7
0.0603	0.0561	0.1289	0.0523	0.0642	0.0551	0.0603
t_8	t_9	t_{10}	t_{11}	t_{12}	t_{13}	t_{14}
0.0561	0.0678	0.0547	0.0525	0.0990	0.0523	0.1403

seen that our proposed algorithm takes only 27 Newton steps to converge, which is very efficient.

To further illustrate the advantage of our proposed algorithm over first-order approaches, we randomly generated 50 network examples with 30 nodes and six sessions, and compared the number of iterations for the proposed algorithm versus the subgradient algorithm. The results are shown in

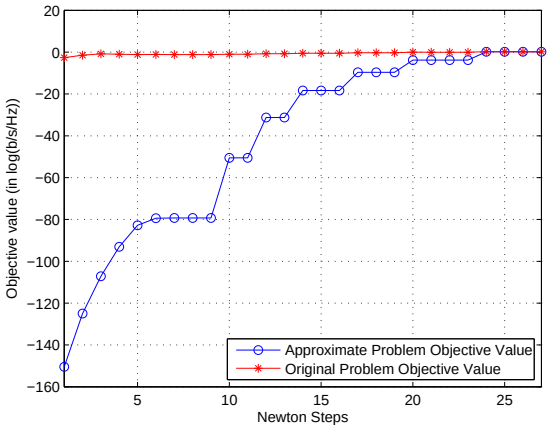


Figure 8: Convergence behavior of the proposed distributed Newton's method for the five-node network example.

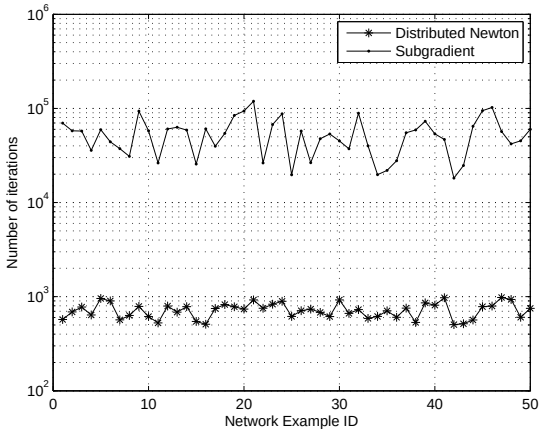


Figure 9: Convergence speed comparison between our proposed algorithm and the subgradient algorithm over 50 randomly generated network examples.

Fig. 9. For these 50 examples, the mean numbers of iterations for our distributed Newton method and the subgradient method are 720.58 and 53870.12, respectively.

6 Conclusion

In this paper, we developed a new distributed Newton's method for cross-layer optimization in wireless networks. We first considered a special network setting where all links mutually interfere with each other. In this case, we derived *closed-form expressions* for the Hessian inverse, which further yielded a distributed implementation of the Newton's method. For general wireless networks where the interference relationships are arbitrary, we proposed a *double matrix-splitting scheme* to compute the primal Newton directions and dual variables, respectively, which also led to a distributed implementation of the Newton's method. Collectively, these results serve as the first building block of a new second-order theoretical framework for cross-layer optimization in wireless networks. Distributed second-order methods for wireless networks is an important and yet under-explored area. Future research topics may include to incorporate signal to interference plus noise ratio (SINR) based interference models, to analyze the impact of inexact line searches on convergence, to design efficient scheduling schemes, and to consider stochastic traffic models.

A Dual Subgradient Method for Solving Problem CLO: An Overview

Since Problem CLO is a linearly constrained convex program and the Slater's condition [7] is satisfied, it can be equivalently solved in its dual domain because of a zero duality gap. To solve the Problem CLO in its dual domain, we first slightly modify the first constraint in Problem CLO as an inequality constraint $\mathbf{A}^{(f)}\mathbf{x}^{(f)} - s_f\widetilde{\mathbf{b}}^{(f)} \geq \mathbf{0}$. This modification does not affect the solution at optimality and can be interpreted from a network stability perspective (i.e., total service rate at each node is no less than the total arrival rate). Then, by associating a dual variable $u_n^{(f)} \geq 0$ for all n and f, and rearranging terms in the Lagrangian, it can be shown that the dual function can be written as follows:

$$\Theta(\mathbf{u}) = \sum_{f=1}^{F} \Theta_{\mathrm{FC}}\Big(u_{\mathrm{Src}(f)}^{(f)}\Big) + \sum_{l=1}^{L} \Theta_{\mathrm{SchR}}\Big(u_{\mathrm{Tx}(l)}^{(f)}, u_{\mathrm{Rx}(l)}^{(f)}\Big),$$

where $\Theta_{\mathrm{FC}}\Big(u_{\mathrm{Src}(f)}^{(f)}\Big)$ and $\Theta_{\mathrm{SchR}}\Big(u_{\mathrm{Tx}(l)}^{(f)}, u_{\mathrm{Rx}(l)}^{(f)}\Big)$ respectively correspond to the flow-control subproblem at node $\mathrm{Src}(f)$ (transport layer) and the joint routing and scheduling subproblem at each link l (network and link layers), and are given as follows:

$$\Theta_{\mathrm{FC}}\Big(u_{\mathrm{Src}(f)}^{(f)}\Big) \triangleq \max\Big\{U_f(s_f) - u_{\mathrm{Src}(f)}^{(f)} s_f \,|s_f \geq 0\Big\}, \tag{79}$$

$$\Theta_{\mathrm{SchR}}\Big(u_{\mathrm{Tx}(l)}^{(f)}, u_{\mathrm{Rx}(l)}^{(f)}\Big) \triangleq \max\left\{\sum_{f=1}^{F}\big(u_{\mathrm{Tx}(l)}^{(f)} - u_{\mathrm{Rx}(l)}^{(f)}\big)x_l^{(f)} \;\middle|\; \begin{array}{l} x_l^{(f)} \geq 0, \\ \sum_{f=1}^{F} x_l^{(f)} \leq \sum_{i=1}^{I} t_i C_l^{(i)} \;\forall f, \\ \sum_{i=1}^{I} t_i = 1, t_i \geq 0, \forall i. \end{array}\right\}. \tag{80}$$

The dual problem can be written as

$$\begin{array}{rl} \text{Minimize} & \Theta(\mathbf{u}) \\ \text{subject to} & \mathbf{u} \geq \mathbf{0}. \end{array} \tag{81}$$

Due to this separable structure, the dual function $\Theta(\mathbf{u})$ can be evaluated by computing $\Theta_{\mathrm{FC}}\Big(u_{\mathrm{Src}(f)}^{(f)}\Big)$ and $\Theta_{\mathrm{R}}\Big(u_{\mathrm{Tx}(l)}^{(f)}, u_{\mathrm{Rx}(l)}^{(f)}\Big)$ for each source node and each link, respectively. The optimal dual variables $\mathbf{u}^*$ can be computed by using the subgradient method iteratively as follows:

$$u_n^{(f)}(k+1) = \max\{u_n^{(f)}(k) - \pi^k d_n^{(f)}(k), 0\}, \quad \forall n, f, \tag{82}$$

where $\pi^k > 0$ is a step size chosen at the k-th iteration and $d_n^{(f)}(k)$ is a subgradient at the k-th iteration, which can be computed as

$$d_n^{(f)}(k) = \begin{cases} \sum_{l\in\mathcal{O}(n)} x_l^{(f)}(k) - \sum_{l\in\mathcal{I}(n)} x_l^{(f)}(k), & \text{if } n \neq \mathrm{Src}(f), \\ \sum_{l\in\mathcal{O}(n)} x_l^{(f)}(k) - s_f(k), & \text{if } n = \mathrm{Src}(f). \end{cases} \tag{83}$$

It can be seen from (83) that the subgradient $d_n^{(f)}(k)$ can also be computed at each node in a decentralized fashion.

There are several interesting networking insights in the subgradient-based first-order method. First, the dual variables $u_n^{(f)}$ can be interpreted as the price charged to session f by node n. For example, if the subgradient component $d_n^{(f)}(k) = \sum_{l\in\mathcal{O}(n)} x_l^{(f)}(k) - \sum_{l\in\mathcal{I}(n)} x_l^{(f)}(k) < 0$, i.e., the stability condition is violated, then the price $w_n^{(f)}$ will increase in the $(k+1)$-st iteration, thus discouraging session f from passing through node n. Second, it can be seen that by dividing by the step size π^k on both sides of (82) and letting $Q_n^{(f)}(k) \triangleq u_n^{(f)}/\pi^k$, we have $Q_n^{(f)}(k+1) = \max\{Q_n^{(f)}(k) - \sum_{l\in\mathcal{O}(n)} x_l^{(f)}(k) + \sum_{l\in\mathcal{I}(n)} x_l^{(f)}(k), 0\}$ if $n \neq \mathrm{Src}(f)$ or $Q_n^{(f)}(k+1) = \max\{Q_n^{(f)}(k) - \sum_{l\in\mathcal{O}(n)} x_l^{(f)}(k) + s_f, 0\}$ if $n = \mathrm{Src}(f)$. This is exactly the queue length evolution of session f at node n. Thus, the dual variables and queue lengths are intimately related (differ by a scaling factor). Lastly, note that the joint scheduling and routing problem $\Theta_{\mathrm{SchR}}\left(u_{\mathrm{Tx}(l)}^{(f)}, u_{\mathrm{Rx}(l)}^{(f)}\right)$ can be decomposed into an outer and an inner subproblems as follows:

$$\Theta_{\mathrm{SchR}}\left(u_{\mathrm{Tx}(l)}^{(f)}, u_{\mathrm{Rx}(l)}^{(f)}\right) = \left\{ \max_{\substack{\sum_{i=1}^I t_i=1,\\ t_i\geq 0,\forall i}} \left[\sum_{l=1}^{L} \left(\begin{array}{ll} \max & \sum_{f=1}^F \left(u_{\mathrm{Tx}(l)}^{(f)} - u_{\mathrm{Rx}(l)}^{(f)}\right) x_l^{(f)} \\ \text{s.t.} & \sum_{f=1}^F x_l^{(f)} \leq \sum_{i=1}^I t_i C_l^{(i)}. \end{array} \right) \right] \right\}. \tag{84}$$

Then, it is easy to see that, for given t_i-values, the joint scheduling and routing problem admits a simple solution: each link picks a flow that has the largest $\left(u_{\mathrm{Tx}(l)}^{(f)} - u_{\mathrm{Rx}(l)}^{(f)}\right)$-value, say f^*, and lets f^* use up the link capacity $\sum_{l=1}^I t_i C_l^{(i)}$. This is exactly the same strategy used in the celebrated "back-pressure" algorithm that was first discovered in [2], even though its throughput-optimality was established using tools in control theory. Next, we fix the $x_l^{(f)}$-values at the obtained solution of the inner subproblem and solve for t_i-variables to maximize the outer subproblem. This process alternates until convergence is reached. However, we note that despite its simplicity and interesting networking interpretations, the subgradient method usually does not work well in practice due to its slow rate of convergence and sensitivity to step-size selection.

It is also worth point out that if (84) needs to be solved in a unit time-slot so that time-sharing is not possible (i.e., the constraints of the outer problem are changed to $t_i \in \{0,1\}$ and $\sum_{i=1}^I t_i = 1$), then (84) reduces to the well-known MAX-Weight scheduling problem, which is NP-hard and many approximation scheduling algorithms have been actively researched.

B Proof of Theorem 4.4

By using the definitions of $\mathbf{D}$, $\mathbf{U}$, $\mathbf{V}$ and noting the *orthogonality* between $\widetilde{\mathbf{e}}_l$ and $\widetilde{\mathbf{c}}_l$, we have that

$$
\begin{aligned}
(\mathbf{I}+\mathbf{V}\mathbf{D}^{-1}\mathbf{U})^{-1} &= \left(\mathbf{I}+\begin{bmatrix} \frac{\widetilde{\mathbf{c}}_1^T}{\delta_1} \\ \vdots \\ \frac{\widetilde{\mathbf{c}}_L^T}{\delta_1} \\ \frac{\widetilde{\mathbf{e}}_1^T}{\delta_1} \\ \vdots \\ \frac{\widetilde{\mathbf{e}}_L^T}{\delta_1} \end{bmatrix}\mathbf{D}^{-1}\begin{bmatrix} \frac{\widetilde{\mathbf{e}}_1}{-\delta_1},\cdots,\frac{\widetilde{\mathbf{e}}_L}{-\delta_L},\frac{\widetilde{\mathbf{c}}_1}{-\delta_1},\cdots,\frac{\widetilde{\mathbf{c}}_L}{-\delta_L}, \end{bmatrix}\right)^{-1} \\
&= \left(\mathbf{I}+\begin{bmatrix} \mathbf{0}_{L\times L} & \mathbf{K}_1 \\ \mathbf{K}_2 & \mathbf{0}_{L\times L} \end{bmatrix}\right)^{-1} = \left(\begin{bmatrix} \mathbf{I} & \mathbf{K}_1 \\ \mathbf{K}_2 & \mathbf{I} \end{bmatrix}\right)^{-1},
\end{aligned}
$$

where the matrices $\mathbf{K}_1$ and $\mathbf{K}_2$ are defined as

$$
\mathbf{K}_1 \triangleq \begin{bmatrix} \frac{\widetilde{\mathbf{c}}_1^T\mathbf{D}^{-1}\widetilde{\mathbf{c}}_1}{-\delta_1^2} & \cdots & \frac{\widetilde{\mathbf{c}}_1^T\mathbf{D}^{-1}\widetilde{\mathbf{c}}_L}{-\delta_1\delta_L} \\ \vdots & & \vdots \\ \frac{\widetilde{\mathbf{c}}_L^T\mathbf{D}^{-1}\widetilde{\mathbf{c}}_1}{-\delta_L\delta_1} & \cdots & \frac{\widetilde{\mathbf{c}}_L^T\mathbf{D}^{-1}\widetilde{\mathbf{c}}_L}{-\delta_L^2} \end{bmatrix}, \text{ and } \mathbf{K}_2 \triangleq \begin{bmatrix} \frac{\widetilde{\mathbf{e}}_1^T\mathbf{D}^{-1}\widetilde{\mathbf{e}}_1}{-\delta_1^2} & \cdots & \frac{\widetilde{\mathbf{e}}_1^T\mathbf{D}^{-1}\widetilde{\mathbf{e}}_L}{-\delta_1\delta_L} \\ \vdots & & \vdots \\ \frac{\widetilde{\mathbf{e}}_L^T\mathbf{D}^{-1}\widetilde{\mathbf{e}}_1}{-\delta_L\delta_1} & \cdots & \frac{\widetilde{\mathbf{e}}_L^T\mathbf{D}^{-1}\widetilde{\mathbf{e}}_L}{-\delta_L^2} \end{bmatrix}.
$$

Next, we further examine each entry in $\mathbf{K}_1$ and $\mathbf{K}_2$. From the sparsity structure of $\widetilde{\mathbf{c}}_l$, the orthogonality among the $\mathbf{c}_l$-vectors, and according to Lemma 4.3, we have

$$
(\mathbf{K}_1)_{l_1l_2} = \frac{\widetilde{\mathbf{c}}_{l_1}^T\mathbf{D}^{-1}\widetilde{\mathbf{c}}_{l_2}}{-\delta_{l_1}\delta_{l_2}} = \frac{\mathbf{c}_{l_1}^T\mathbf{T}^{-1}\mathbf{c}_{l_2}}{-\delta_{l_1}\delta_{l_2}} = \begin{cases} -\frac{t_l^2}{t_l^2+\delta_l^2}, & 1\le l_1=l_2=l\le L, \\ 0, & l_1\neq l_2. \end{cases} \tag{85}
$$

Likewise, from the sparsity structure of $\widetilde{\mathbf{e}}_l$ and the orthogonality among the $\widetilde{\mathbf{e}}_l$-vectors, we have

$$
(\mathbf{K}_2)_{l_1l_2} = \frac{\widetilde{\mathbf{e}}_{l_1}^T\mathbf{D}^{-1}\widetilde{\mathbf{e}}_{l_2}}{-\delta_{l_1}\delta_{l_2}} = \begin{cases} \frac{\mathbf{1}^T\mathbf{X}_l^{-1}\mathbf{1}}{-\delta_l^2}, & 1\le l_1=l_2=l\le L, \\ 0, & l_1\neq l_2. \end{cases} \tag{86}
$$

Therefore, we can conclude from (85) and (86) that both $\mathbf{K}_1$ and $\mathbf{K}_2$ are both diagonal, and can be simplified as:

$$
\mathbf{K}_1 = \mathrm{Diag}\left\{-\frac{t_1^2}{t_1^2+\delta_1^2},\cdots,-\frac{t_L^2}{t_L^2+\delta_L^2}\right\}, \text{ and } \mathbf{K}_2 = \mathrm{Diag}\left\{\frac{\mathbf{1}^T\mathbf{X}_1^{-1}\mathbf{1}}{-\delta_1^2},\cdots,\frac{\mathbf{1}^T\mathbf{X}_L^{-1}\mathbf{1}}{-\delta_L^2}\right\}.
$$

We point out that, by using Lemma 4.2, $\mathbf{1}^T\mathbf{X}_l^{-1}\mathbf{1}$ can be computed in closed-form as follows:

$$
\mathbf{1}^T\mathbf{X}_l^{-1}\mathbf{1} = \sum_{f_1=1}^{F}\sum_{f_2=1}^{F}(\mathbf{X}_l^{-1})_{f_1f_2} = \sum_{f=1}^{F}(x_l^{(f)})^2 - \sum_{f_1=1}^{F}\sum_{f_2=1}^{F}\frac{(x_l^{(f_1)}x_l^{(f_2)})^2}{\|\hat{\mathbf{x}}_l\|^2}.
$$

However, for notational convenience, we retain the matrix term $\mathbf{1}^T\mathbf{X}_l^{-1}\mathbf{1}$ throughout in this paper.

Next, from the blockwise matrix inversion theorem [23], we have

$$\begin{bmatrix} \mathbf{I} & \mathbf{K}_1 \\ \mathbf{K}_2 & \mathbf{I} \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{Q}_1 & \mathbf{Q}_2 \\ \mathbf{Q}_3 & \mathbf{Q}_4 \end{bmatrix},$$

where the matrices $\mathbf{Q}_i$, $i = 1, \ldots, 4$, are defined as

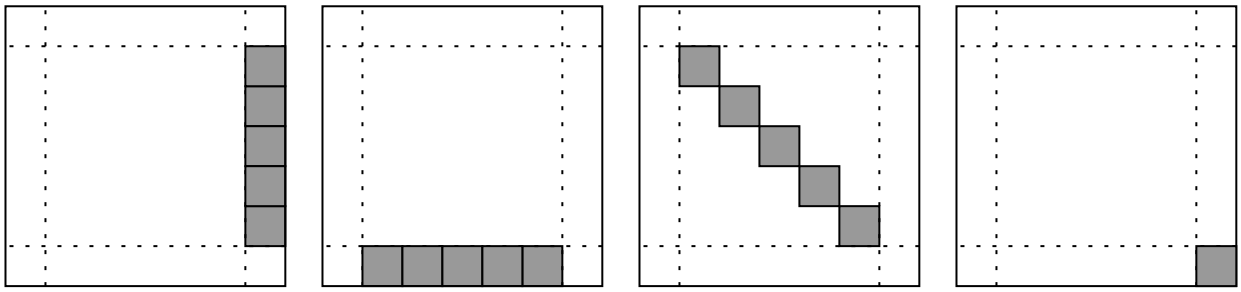
$$\begin{aligned}
\mathbf{Q}_1 &\triangleq \mathbf{I} + \mathbf{K}_1(\mathbf{I} - \mathbf{K}_2\mathbf{K}_1)^{-1}\mathbf{K}_2, \\
\mathbf{Q}_2 &\triangleq -\mathbf{K}_1(\mathbf{I} - \mathbf{K}_2\mathbf{K}_1)^{-1}, \\
\mathbf{Q}_3 &\triangleq -(\mathbf{I} - \mathbf{K}_2\mathbf{K}_1)^{-1}\mathbf{K}_2, \\
\mathbf{Q}_4 &\triangleq (\mathbf{I} - \mathbf{K}_2\mathbf{K}_1)^{-1}.
\end{aligned}$$

It can be seen that a common term in $\mathbf{Q}_i$, $i = 1, \ldots, 4$, is $(\mathbf{I} - \mathbf{K}_2\mathbf{K}_1)$, which is also known as the Schur complement [23]. Thanks to the diagonal structure of $\mathbf{K}_1$ and $\mathbf{K}_2$, we have

$$\begin{aligned}
\mathbf{I} - \mathbf{K}_2\mathbf{K}_1 &= \mathbf{I} - \mathrm{Diag}\left\{-\frac{t_1^2}{t_1^2+\delta_1^2}, \cdots, -\frac{t_L^2}{t_L^2+\delta_L^2}\right\} \mathrm{Diag}\left\{\frac{\mathbf{1}^T\mathbf{X}_1^{-1}\mathbf{1}}{-\delta_1^2}, \cdots, \frac{\mathbf{1}^T\mathbf{X}_L^{-1}\mathbf{1}}{-\delta_L^2}\right\} \\
&= \mathrm{Diag}\left\{1 - \frac{t_1^2(\mathbf{1}^T\mathbf{X}_1\mathbf{1})}{\delta_1^2(t_1^2+\delta_1^2)}, \cdots, 1 - \frac{t_L^2(\mathbf{1}^T\mathbf{X}_L^{-1}\mathbf{1})}{\delta_L^2(t_L^2+\delta_L^2)}\right\},
\end{aligned}$$

which shows that the Schur complement is again a diagonal matrix. Thus, we have

$$\begin{aligned}
\mathbf{Q}_4 &= (\mathbf{I} - \mathbf{K}_2\mathbf{K}_1)^{-1} = \mathrm{Diag}\left\{\frac{\delta_l^2(t_l^2+\delta_l^2)}{\delta_l^2(t_l^2+\delta_l^2) - t_l^2(\mathbf{1}^T\mathbf{X}_l^{-1}\mathbf{1})},\ l = 1, \ldots, L\right\}, \\
\mathbf{Q}_1 &= \mathbf{I} + \mathbf{K}_1(\mathbf{I} - \mathbf{K}_2\mathbf{K}_1)^{-1}\mathbf{K}_2 \\
&= \mathbf{I} + \mathrm{Diag}\left\{\frac{-t_l^2}{t_l^2+\delta_l^2}, \forall l\right\} \mathrm{Diag}\left\{\frac{\delta_l^2(t_l^2+\delta_l^2)}{\delta_l^2(t_l^2+\delta_l^2) - t_l^2(\mathbf{1}^T\mathbf{X}_l^{-1}\mathbf{1})}, \forall l\right\} \mathrm{Diag}\left\{\frac{\mathbf{1}^T\mathbf{X}_l^{-1}\mathbf{1}}{-\delta_l^2}, \forall l\right\} \\
&= \mathrm{Diag}\left\{\frac{\delta_l^2(t_l^2+\delta_l^2)}{\delta_l^2(t_l^2+\delta_l^2) - t_l^2(\mathbf{1}^T\mathbf{X}_l^{-1}\mathbf{1})},\ l = 1, \ldots, L\right\} = \mathbf{Q}_4, \\
\mathbf{Q}_2 &= -\mathbf{K}_1(\mathbf{I} - \mathbf{K}_2\mathbf{K}_1)^{-1} \\
&= -\mathrm{Diag}\left\{\frac{-t_l^2}{t_l^2+\delta_l^2}, \forall l\right\} \mathrm{Diag}\left\{\frac{\delta_l^2(t_l^2+\delta_l^2)}{\delta_l^2(t_l^2+\delta_l^2) - t_l^2(\mathbf{1}^T\mathbf{X}_l^{-1}\mathbf{1})}, \forall l\right\} \\
&= \mathrm{Diag}\left\{\frac{t_l^2\delta_l^2}{\delta_l^2(t_l^2+\delta_l^2) - t_l^2(\mathbf{1}^T\mathbf{X}_l^{-1}\mathbf{1})},\ l = 1, \ldots, L\right\}, \\
\mathbf{Q}_3 &= -(\mathbf{I} - \mathbf{K}_2\mathbf{K}_1)^{-1}\mathbf{K}_2 \\
&= \mathrm{Diag}\left\{\frac{(t_l^2+\delta_l^2)(\mathbf{1}^T\mathbf{X}_l^{-1}\mathbf{1})}{\delta_l^2(t_l^2+\delta_l^2) - t_l^2(\mathbf{1}^T\mathbf{X}_l^{-1}\mathbf{1})},\ l = 1, \ldots, L\right\}.
\end{aligned}$$



(a) $\mathbf{W}_1^T\mathbf{Q}_1\mathbf{W}_2$ (b) $\mathbf{W}_2^T\mathbf{Q}_1\mathbf{W}_1$ (c) $\mathbf{W}_1^T\mathbf{Q}_2\mathbf{W}_1$ (d) $\mathbf{W}_2^T\mathbf{Q}_3\mathbf{W}_2$

Figure 10: Illustration of the blockwise structural properties of the matrices terms in (87).

Note that $\mathbf{Q}_i$, $i = 1, \ldots, 3$, are exactly the same as stated in the theorem and $\mathbf{Q}_4 = \mathbf{Q}_1$. This completes the proof.

C Proof of Theorem 4.5

Recall that by the SMW formula, $\mathbf{H}_k^{-1} = (\mathbf{D} + \mathbf{UV})^{-1} = \mathbf{D}^{-1} - \mathbf{D}^{-1}\mathbf{U}(\mathbf{I} + \mathbf{VD}^{-1}\mathbf{U})^{-1}\mathbf{VD}^{-1}$. We will first derive $\mathbf{U}(\mathbf{I} + \mathbf{VD}^{-1}\mathbf{U})^{-1}\mathbf{V}$. For convenience, we define the following two matrices:

$$\mathbf{W}_1 \triangleq \left[\frac{\widetilde{\mathbf{e}}_1}{\delta_1}, \cdots, \frac{\widetilde{\mathbf{e}}_L}{\delta_L}\right]^T \in \mathbb{R}^{L\times[(L+1)F+L]}, \quad \mathbf{W}_2 \triangleq \left[\frac{\widetilde{\mathbf{c}}_1}{\delta_1}, \cdots, \frac{\widetilde{\mathbf{c}}_L}{\delta_L}\right]^T \in \mathbb{R}^{L\times[(L+1)F+L]}.$$

Then, we have $\mathbf{U} = \begin{bmatrix} -\mathbf{W}_1^T & -\mathbf{W}_2^T \end{bmatrix}$ and $\mathbf{V} = \begin{bmatrix} \mathbf{W}_2^T & \mathbf{W}_1^T \end{bmatrix}^T$. From Theorem 4.4, we have that

$$\begin{aligned}\mathbf{U}(\mathbf{I} + \mathbf{VD}^{-1}\mathbf{U})^{-1}\mathbf{V} &= \begin{bmatrix} -\mathbf{W}_1^T & -\mathbf{W}_2^T \end{bmatrix}\begin{bmatrix} \mathbf{Q}_1 & \mathbf{Q}_2 \\ \mathbf{Q}_3 & \mathbf{Q}_1 \end{bmatrix}\begin{bmatrix} \mathbf{W}_2 \\ \mathbf{W}_1 \end{bmatrix} \\ &= -\left[\mathbf{W}_1^T\mathbf{Q}_2\mathbf{W}_1 + \mathbf{W}_2^T\mathbf{Q}_3\mathbf{W}_2 + \mathbf{W}_1^T\mathbf{Q}_1\mathbf{W}_2 + \mathbf{W}_2^T\mathbf{Q}_1\mathbf{W}_1\right]. \qquad (87)\end{aligned}$$

Now, evaluating each term in (87), we have

$$\begin{aligned}
\mathbf{W}_1^T\mathbf{Q}_1\mathbf{W}_2 &= \left[\frac{\widetilde{\mathbf{e}}_1}{\delta_1},\cdots,\frac{\widetilde{\mathbf{e}}_L}{\delta_L}\right]\mathrm{Diag}\left\{\frac{\delta_l^2(t_l^2+\delta_l^2)}{\delta_l^2(t_l^2+\delta_l^2)-t_l^2(\mathbf{1}^T\mathbf{X}_l^{-1}\mathbf{1})},\forall l\right\}\begin{bmatrix}\frac{\widetilde{\mathbf{c}}_1^T}{\delta_1}\\ \vdots \\ \frac{\widetilde{\mathbf{c}}_L^T}{\delta_L}\end{bmatrix}\\
&= \sum_{l=1}^{L}\left(\frac{t_l^2+\delta_l^2}{\delta_l^2(t_l^2+\delta_l^2)-t_l^2(\mathbf{1}^T\mathbf{X}_l^{-1}\mathbf{1})}\right)\widetilde{\mathbf{e}}_l\widetilde{\mathbf{c}}_l^T,\\
\mathbf{W}_2^T\mathbf{Q}_1\mathbf{W}_1 &= \left[\frac{\widetilde{\mathbf{c}}_1}{\delta_1},\cdots,\frac{\widetilde{\mathbf{c}}_L}{\delta_L}\right]\mathrm{Diag}\left\{\frac{\delta_l^2(t_l^2+\delta_l^2)}{\delta_l^2(t_l^2+\delta_l^2)-t_l^2(\mathbf{1}^T\mathbf{X}_l^{-1}\mathbf{1})},\forall l\right\}\begin{bmatrix}\frac{\widetilde{\mathbf{e}}_1^T}{\delta_1}\\ \vdots \\ \frac{\widetilde{\mathbf{e}}_L^T}{\delta_L}\end{bmatrix}\\
&= \sum_{l=1}^{L}\left(\frac{t_l^2+\delta_l^2}{\delta_l^2(t_l^2+\delta_l^2)-t_l^2(\mathbf{1}^T\mathbf{X}_l^{-1}\mathbf{1})}\right)\widetilde{\mathbf{c}}_l\widetilde{\mathbf{e}}_l^T,\\
\mathbf{W}_1^T\mathbf{Q}_2\mathbf{W}_1 &= \left[\frac{\widetilde{\mathbf{e}}_1}{\delta_1},\cdots,\frac{\widetilde{\mathbf{e}}_L}{\delta_L}\right]\mathrm{Diag}\left\{\frac{t_l^2\delta_l^2}{\delta_l^2(t_l^2+\delta_l^2)-t_l^2(\mathbf{1}^T\mathbf{X}_l^{-1}\mathbf{1})},\forall l\right\}\begin{bmatrix}\frac{\widetilde{\mathbf{e}}_1^T}{\delta_1}\\ \vdots \\ \frac{\widetilde{\mathbf{e}}_L^T}{\delta_L}\end{bmatrix}\\
&= \sum_{l=1}^{L}\left(\frac{t_l^2}{\delta_l^2(t_l^2+\delta_l^2)-t_l^2(\mathbf{1}^T\mathbf{X}_l^{-1}\mathbf{1})}\right)\widetilde{\mathbf{e}}_l\widetilde{\mathbf{e}}_l^T,\\
\mathbf{W}_2^T\mathbf{Q}_3\mathbf{W}_2 &= \left[\frac{\widetilde{\mathbf{c}}_1}{\delta_1},\cdots,\frac{\widetilde{\mathbf{c}}_L}{\delta_L}\right]\mathrm{Diag}\left\{\frac{(t_l^2+\delta_l^2)(\mathbf{1}^T\mathbf{X}_l^{-1}\mathbf{1})}{\delta_l^2(t_l^2+\delta_l^2)-t_l^2(\mathbf{1}^T\mathbf{X}_l^{-1}\mathbf{1})},\forall l\right\}\begin{bmatrix}\frac{\widetilde{\mathbf{c}}_1^T}{\delta_1}\\ \vdots \\ \frac{\widetilde{\mathbf{c}}_L^T}{\delta_L}\end{bmatrix}\\
&= \sum_{l=1}^{L}\left(\frac{(1+t_l^2/\delta_l^2)(\mathbf{1}^T\mathbf{X}_l^{-1}\mathbf{1})}{\delta_l^2(t_l^2+\delta_l^2)-t_l^2(\mathbf{1}^T\mathbf{X}_l^{-1}\mathbf{1})}\right)\widetilde{\mathbf{c}}_l\widetilde{\mathbf{c}}_l^T.
\end{aligned}$$

It is worth pointing out that, due to the special sparsity structure of $\widetilde{\mathbf{c}}_l$ and $\widetilde{\mathbf{e}}_l$, the matrices $\mathbf{W}_1^T\mathbf{Q}_2\mathbf{W}_1$, $\mathbf{W}_2^T\mathbf{Q}_3\mathbf{W}_2$, $\mathbf{W}_1^T\mathbf{Q}_1\mathbf{W}_2$, and $\mathbf{W}_2^T\mathbf{Q}_1\mathbf{W}_1$ have blockwise structures as illustrated in Figure 10, where we can see that $\mathbf{W}_1^T\mathbf{Q}_1\mathbf{W}_2$ and $\mathbf{W}_2^T\mathbf{Q}_1\mathbf{W}_1$ correspond to vertical and horizontal bands, respectively, $\mathbf{W}_1^T\mathbf{Q}_2\mathbf{W}_1$ corresponds to the main diagonal blocks (except the first and last diagonal blocks), and $\mathbf{W}_2^T\mathbf{Q}_3\mathbf{W}_2$ corresponds to the last diagonal block. It is these blockwise structures that significantly simplify our later derivations.

Next, from (87), we have that:

$$\begin{aligned}
&\mathbf{D}^{-1}\mathbf{U}(\mathbf{I}+\mathbf{V}\mathbf{D}^{-1}\mathbf{U})^{-1}\mathbf{V}\mathbf{D}^{-1}\\
=&-\mathbf{D}^{-1}\left[\mathbf{W}_1^T\mathbf{Q}_2\mathbf{W}_1+\mathbf{W}_2^T\mathbf{Q}_3\mathbf{W}_2+\mathbf{W}_1^T\mathbf{Q}_1\mathbf{W}_2+\mathbf{W}_2^T\mathbf{Q}_1\mathbf{W}_1\right]\mathbf{D}^{-1}. \qquad (88)
\end{aligned}$$

Evaluating each term in (88) and noting the blockwise structures, we have that

1) $\mathbf{D}^{-1}(\mathbf{W}_1^T\mathbf{Q}_2\mathbf{W}_1)\mathbf{D}^{-1}$:

$$\begin{aligned}\mathbf{D}^{-1}(\mathbf{W}_1^T\mathbf{Q}_2\mathbf{W}_1)\mathbf{D}^{-1} &= \mathrm{Diag}\left\{\mathbf{S}^{-1},\mathbf{X}_1^{-1},\ldots,\mathbf{X}_L^{-1},\mathbf{T}^{-1}\right\}\\ &\quad\times\left[\sum_{l=1}^{L}\left(\frac{t_l^2}{\delta_l^2(t_l^2+\delta_l^2)-t_l^2(\mathbf{1}^T\mathbf{X}_l^{-1}\mathbf{1})}\right)\widetilde{\mathbf{e}}_l\widetilde{\mathbf{e}}_l^T\right]\mathrm{Diag}\left\{\mathbf{S}^{-1},\mathbf{X}_1^{-1},\ldots,\mathbf{X}_L^{-1},\mathbf{T}^{-1}\right\}\\ &= \mathrm{Diag}\left\{\mathbf{0}_{F\times F},\widetilde{\mathbf{X}}_1,\ldots,\widetilde{\mathbf{X}}_L,\mathbf{0}_{L\times L}\right\},\end{aligned}$$

where $\widetilde{\mathbf{X}}_l$ can be computed as

$$\begin{aligned}\widetilde{\mathbf{X}}_l &= \mathbf{X}_l^{-1}\left[\left(\frac{t_l^2}{\delta_l^2(t_l^2+\delta_l^2)-t_l^2(\mathbf{1}^T\mathbf{X}_l^{-1}\mathbf{1})}\right)\mathbf{1}\mathbf{1}^T\right]\mathbf{X}_l^{-1} = \frac{t_l^2}{\delta_l^2(t_l^2+\delta_l^2)-t_l^2(\mathbf{1}^T\mathbf{X}_l^{-1}\mathbf{1})}\\ &\quad\times\left(\begin{bmatrix}(x_l^{(1)})^2 & & \\ & \ddots & \\ & & (x_l^{(F)})^2\end{bmatrix}-\frac{1}{\|\hat{\mathbf{x}}_l\|^2}\begin{bmatrix}(x_l^{(1)})^2\\ \vdots\\ (x_l^{(F)})^2\end{bmatrix}\begin{bmatrix}(x_l^{(1)})^2 & \cdots & (x_l^{(F)})^2\end{bmatrix}\right)\mathbf{1}\mathbf{1}^T\\ &\quad\times\left(\begin{bmatrix}(x_l^{(1)})^2 & & \\ & \ddots & \\ & & (x_l^{(F)})^2\end{bmatrix}-\frac{1}{\|\hat{\mathbf{x}}_l\|^2}\begin{bmatrix}(x_l^{(1)})^2\\ \vdots\\ (x_l^{(F)})^2\end{bmatrix}\begin{bmatrix}(x_l^{(1)})^2 & \cdots & (x_l^{(F)})^2\end{bmatrix}^T\right)^T\\ &= \frac{t_l^2\delta_l^4/\|\hat{\mathbf{x}}_l\|^2}{\delta_l^2(t_l^2+\delta_l^2)-t_l^2(\mathbf{1}^T\mathbf{X}_l^{-1}\mathbf{1})}\left(\frac{1}{\|\hat{\mathbf{x}}_l\|^2}\right)\begin{bmatrix}(x_l^{(1)})^2\\ \vdots\\ (x_l^{(F)})^2\end{bmatrix}\begin{bmatrix}(x_l^{(1)})^2 & \cdots & (x_l^{(F)})^2\end{bmatrix}.\end{aligned}$$

2) $\mathbf{D}^{-1}(\mathbf{W}_2^T\mathbf{Q}_3\mathbf{W}_2)\mathbf{D}^{-1}$:

$$\begin{aligned}\mathbf{D}^{-1}(\mathbf{W}_2^T\mathbf{Q}_3\mathbf{W}_2)\mathbf{D}^{-1} &= \mathrm{Diag}\left\{\mathbf{S}^{-1},\mathbf{X}_1^{-1},\ldots,\mathbf{X}_L^{-1},\mathbf{T}^{-1}\right\}\\ &\quad\times\sum_{l=1}^{L}\left(\frac{(1+t_l^2/\delta_l^2)(\mathbf{1}^T\mathbf{X}_l^{-1}\mathbf{1})}{\delta_l^2(t_l^2+\delta_l^2)-t_l^2(\mathbf{1}^T\mathbf{X}_l^{-1}\mathbf{1})}\right)\widetilde{\mathbf{c}}_l\widetilde{\mathbf{c}}_l^T\mathrm{Diag}\left\{\mathbf{S}^{-1},\mathbf{X}_1^{-1},\ldots,\mathbf{X}_L^{-1},\mathbf{T}^{-1}\right\}\\ &= \mathrm{Diag}\left\{\mathbf{0}_{F\times F},\mathbf{0}_{F\times F},\ldots,\mathbf{0}_{F\times F},\widetilde{\mathbf{T}}\right\}\in\mathbb{R}^{[(L+1)F+L]\times[(L+1)F+L]},\end{aligned}$$

where $\widetilde{\mathbf{T}}$ can be computed as

$$\begin{aligned}\widetilde{\mathbf{T}} &= \mathbf{T}^{-1}\mathrm{Diag}\left\{\frac{(1+t_l^2/\delta_l^2)(\mathbf{1}^T\mathbf{X}_l^{-1}\mathbf{1})}{\delta_l^2(t_l^2+\delta_l^2)-t_l^2(\mathbf{1}^T\mathbf{X}_l^{-1}\mathbf{1})}\right\}\mathbf{T}^{-1}\\ &= \mathrm{Diag}\left\{\frac{t_l^4\delta_l^2(\mathbf{1}^T\mathbf{X}_l^{-1}\mathbf{1})/(t_l^2+\delta_l^2)}{\delta_l^2(t_l^2+\delta_l^2)-t_l^2(\mathbf{1}^T\mathbf{X}_l^{-1}\mathbf{1})},\ l=1,\ldots,L\right\}.\end{aligned}$$

3) $\mathbf{D}^{-1}(\mathbf{W}_1^T\mathbf{Q}_1\mathbf{W}_2)\mathbf{D}^{-1}$:

$$\begin{aligned}\mathbf{D}^{-1}(\mathbf{W}_1^T\mathbf{Q}_1\mathbf{W}_2)\mathbf{D}^{-1} &= \mathrm{Diag}\left\{\mathbf{S}^{-1},\mathbf{X}_1^{-1},\ldots,\mathbf{X}_L^{-1},\mathbf{T}^{-1}\right\}\\ &\quad\times\left(\sum_{l=1}^{L}\left(\frac{t_l^2+\delta_l^2}{\delta_l^2(t_l^2+\delta_l^2)-t_l^2(\mathbf{1}^T\mathbf{X}_l^{-1}\mathbf{1})}\right)\widetilde{\mathbf{e}}_l\widetilde{\mathbf{c}}_l^T\right)\mathrm{Diag}\left\{\mathbf{S}^{-1},\mathbf{X}_1^{-1},\ldots,\mathbf{X}_L^{-1},\mathbf{T}^{-1}\right\}.\end{aligned}$$

Due to the special sparsity structure of $\widetilde{\mathbf{e}}_l$ and $\widetilde{\mathbf{c}}_l$, $\mathbf{D}^{-1}(\mathbf{W}_1^T\mathbf{Q}_1\mathbf{W}_2)\mathbf{D}^{-1}$ leads to a non-zero last column with L $(F \times L)$-dimensional blocks, which we denote as $\widetilde{\mathbf{C}}_l$, $l = 1, \ldots, L$. Each $\widetilde{\mathbf{C}}_l$ can be computed as:

$$\begin{aligned}
\widetilde{\mathbf{C}}_l &= \mathbf{X}_l^{-1}\left(\frac{t_l^2+\delta_l^2}{\delta_l^2(t_l^2+\delta_l^2)-t_l^2(\mathbf{1}^T\mathbf{X}_l^{-1}\mathbf{1})}\right)\mathbf{1}\mathbf{e}_l^T\mathbf{T}^{-1} = \left(\frac{t_l^2+\delta_l^2}{\delta_l^2(t_l^2+\delta_l^2)-t_l^2(\mathbf{1}^T\mathbf{X}_l^{-1}\mathbf{1})}\right) \\
&\times\left(\begin{bmatrix} (x_l^{(1)})^2 & & \\ & \ddots & \\ & & (x_l^{(F)})^2 \end{bmatrix} - \frac{1}{\|\hat{\mathbf{x}}_l\|^2}\begin{bmatrix} (x_l^{(1)})^2 \\ \vdots \\ (x_l^{(F)})^2 \end{bmatrix}\begin{bmatrix} (x_l^{(1)})^2 & \cdots & (x_l^{(F)})^2 \end{bmatrix}\right)\begin{bmatrix} 1 \\ \vdots \\ 1 \end{bmatrix} \\
&\times\begin{bmatrix} 0 & \cdots & 1 & \cdots & 0 \end{bmatrix}\mathrm{Diag}\left\{\frac{t_l^2\delta_l^2}{t_l^2+\delta_l^2},\ \forall l\right\} \\
&= \left(\frac{t_l^2\delta_l^4/\|\hat{\mathbf{x}}_l\|^2}{\delta_l^2(t_l^2+\delta_l^2)-t_l^2(\mathbf{1}^T\mathbf{X}_l^{-1}\mathbf{1})}\right)\begin{bmatrix} 0 & \cdots & (x_l^{(1)})^2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & (x_l^{(F)})^2 & \cdots & 0 \end{bmatrix}.
\end{aligned}$$

4) $\mathbf{D}^{-1}(\mathbf{W}_2^T\mathbf{Q}_1\mathbf{W}_1)\mathbf{D}^{-1}$:

$$\begin{aligned}
\mathbf{D}^{-1}(\mathbf{W}_2^T\mathbf{Q}_1\mathbf{W}_1)\mathbf{D}^{-1} &= \mathrm{Diag}\left\{\mathbf{S}^{-1}, \mathbf{X}_1^{-1}, \ldots, \mathbf{X}_L^{-1}, \mathbf{T}^{-1}\right\} \\
&\times\left(\sum_{l=1}^{L}\left(\frac{t_l^2+\delta_l^2}{\delta_l^2(t_l^2+\delta_l^2)-t_l^2(\mathbf{1}^T\mathbf{X}_l^{-1}\mathbf{1})}\right)\widetilde{\mathbf{c}}_l\widetilde{\mathbf{e}}_l^T\right)\mathrm{Diag}\left\{\mathbf{S}^{-1}, \mathbf{X}_1^{-1}, \ldots, \mathbf{X}_L^{-1}, \mathbf{T}^{-1}\right\}.
\end{aligned}$$

Due to the special sparsity structure of $\widetilde{\mathbf{e}}_l$ and $\widetilde{\mathbf{c}}_l$, the matrix $\mathbf{D}^{-1}(\mathbf{W}_1^T\mathbf{Q}_1\mathbf{W}_2)\mathbf{D}^{-1}$ has a non-zero bottom column with L $(L \times F)$-dimensional blocks, which happens to equal to $\widetilde{\mathbf{C}}_l^T$, $l =$

$1, \ldots, L$. Thus, each $\widetilde{\mathbf{C}}_l^T$ can be computed as:

$$\begin{aligned}
\widetilde{\mathbf{C}}_l^T &= \mathbf{T}^{-1}\left(\frac{t_l^2+\delta_l^2}{\delta_l^2(t_l^2+\delta_l^2)-t_l^2(\mathbf{1}^T\mathbf{X}_l^{-1}\mathbf{1})}\right)\mathbf{e}_l\mathbf{1}^T\mathbf{X}_l^{-1} = \left(\frac{t_l^2+\delta_l^2}{\delta_l^2(t_l^2+\delta_l^2)-t_l^2(\mathbf{1}^T\mathbf{X}_l^{-1}\mathbf{1})}\right) \\
&= \mathrm{Diag}\left\{\frac{t_l^2\delta_l^2}{t_l^2+\delta_l^2},\ \forall l\right\}\begin{bmatrix} 0 \\ \vdots \\ 1 \\ \vdots \\ 0 \end{bmatrix}\begin{bmatrix} 1 & \cdots & 1 \end{bmatrix} \\
&\times\left(\begin{bmatrix} (x_l^{(1)})^2 & & \\ & \ddots & \\ & & (x_l^{(F)})^2 \end{bmatrix} - \frac{1}{\|\hat{\mathbf{x}}_l\|^2}\begin{bmatrix} (x_l^{(1)})^2 \\ \vdots \\ (x_l^{(F)})^2 \end{bmatrix}\begin{bmatrix} (x_l^{(1)})^2 & \cdots & (x_l^{(F)})^2 \end{bmatrix}\right) \\
&= \left(\frac{t_l^2\delta_l^4/\|\hat{\mathbf{x}}_l\|^2}{\delta_l^2(t_l^2+\delta_l^2)-t_l^2(\mathbf{1}^T\mathbf{X}_l^{-1}\mathbf{1})}\right)\begin{bmatrix} 0 & \cdots & 0 \\ (x_l^{(1)})^2 & \cdots & (x_l^{(F)})^2 \\ 0 & \cdots & 0 \end{bmatrix}.
\end{aligned}$$

We note that after all the above algebraic derivations, the matrix terms $\mathbf{D}^{-1}(\mathbf{W}_1^T\mathbf{Q}_2\mathbf{W}_1)\mathbf{D}^{-1}$, $\mathbf{D}^{-1}(\mathbf{W}_2^T\mathbf{Q}_3\mathbf{W}_2)\mathbf{D}^{-1}$, $\mathbf{D}^{-1}(\mathbf{W}_1^T\mathbf{Q}_1\mathbf{W}_2)\mathbf{D}^{-1}$, and $\mathbf{D}^{-1}(\mathbf{W}_2^T\mathbf{Q}_1\mathbf{W}_1)\mathbf{D}^{-1}$ still preserve the blockwise structure as shown in Figure 10. These nice blockwise structures play an important role in simplifying our next step, i.e.,

$$\mathbf{H}_k^{-1} = \mathbf{D}^{-1} + \mathbf{D}^{-1}\left[\mathbf{W}_1^T\mathbf{Q}_2\mathbf{W}_1 + \mathbf{W}_2^T\mathbf{Q}_3\mathbf{W}_2 + \mathbf{W}_1^T\mathbf{Q}_1\mathbf{W}_2 + \mathbf{W}_2^T\mathbf{Q}_1\mathbf{W}_1\right]\mathbf{D}^{-1}.$$

Note that $\mathbf{D}^{-1}$ is also block diagonal. As a result, the structure of $\widetilde{\mathbf{H}}_k^{-1}$ must possess the following structure:

$$\mathbf{H}_k^{-1} = \left[\begin{array}{c|ccc} \widehat{\mathbf{S}} & & & \\ \hline & \widehat{\mathbf{X}}_1 & & \widehat{\mathbf{C}}_1 \\ & & \ddots & \vdots \\ & & \widehat{\mathbf{X}}_L & \widehat{\mathbf{C}}_L \\ & \widehat{\mathbf{C}}_1^T & \cdots \quad \widehat{\mathbf{C}}_L^T & \widehat{\mathbf{T}} \end{array}\right],$$

where $\widehat{\mathbf{S}}$, $\widehat{\mathbf{X}}_l$, $\widehat{\mathbf{T}}$, and $\widehat{\mathbf{C}}_l$ can be computed as:

$$\begin{aligned}
&\widehat{\mathbf{S}} = \mathbf{S}^{-1}, \\
&\widehat{\mathbf{X}}_l = \mathbf{X}_l^{-1} + \widetilde{\mathbf{X}}_l, \quad \forall l, \\
&\widehat{\mathbf{T}} = \mathbf{T}^{-1} + \widetilde{\mathbf{T}}, \\
&\widehat{\mathbf{C}}_l = \widetilde{\mathbf{C}}_l, \quad \forall l.
\end{aligned}$$

This proves the first part of the theorem. Further, we note that

$$\mathbf{S}^{-1} = \text{Diag}\left\{ \frac{1}{-\mu U_1''(s_1) + 1/s_1^2}, \ldots, \frac{1}{-\mu U_F''(s_F) + 1/s_F^2} \right\},$$

$$\mathbf{X}_l^{-1} + \widetilde{\mathbf{X}}_l = \left(\begin{bmatrix} (x_l^{(1)})^2 & & \\ & \ddots & \\ & & (x_l^{(F)})^2 \end{bmatrix} - \frac{1}{\|\hat{\mathbf{x}}_l\|^2} \begin{bmatrix} (x_l^{(1)})^2 \\ \vdots \\ (x_l^{(F)})^2 \end{bmatrix} \begin{bmatrix} (x_l^{(1)})^2 & \cdots & (x_l^{(F)})^2 \end{bmatrix} \right)$$

$$+ \frac{t_l^2 \delta_l^4 / \|\hat{\mathbf{x}}_l\|^2}{\delta_l^2(t_l^2+\delta_l^2) - t_l^2(\mathbf{1}^T \mathbf{X}_l^{-1}\mathbf{1})} \left(\frac{1}{\|\hat{\mathbf{x}}_l\|^2} \right) \begin{bmatrix} (x_l^{(1)})^2 \\ \vdots \\ (x_l^{(F)})^2 \end{bmatrix} \begin{bmatrix} (x_l^{(1)})^2 & \cdots & (x_l^{(F)})^2 \end{bmatrix}$$

$$= \begin{cases} \left(x_l^{(f_1)}\right)^2 \left[1 - \frac{\left(x_l^{(f_1)}\right)^2}{\|\hat{\mathbf{x}}_l\|^2} \left(1 - \frac{t_l^2 \delta_l^4 / \|\hat{\mathbf{x}}_l\|^2}{\delta_l^2(t_l^2+\delta_l^2) - t_l^2(\mathbf{1}^T \mathbf{X}_l^{-1}\mathbf{1})} \right) \right], & \text{if } 1 \le f_1 = f_2 \le F, \\ -\frac{(x_l^{(f_1)} x_l^{(f_2)})^2}{\|\hat{\mathbf{x}}_l\|^2} \left(1 - \frac{t_l^2 \delta_l^4 / \|\hat{\mathbf{x}}_l\|^2}{\delta_l^2(t_l^2+\delta_l^2) - t_l^2(\mathbf{1}^T \mathbf{X}_l^{-1}\mathbf{1})} \right), & \text{if } 1 \le f_1 \ne f_2 \le F, \end{cases}$$

$$(\widetilde{\mathbf{C}}_l)_{f l_1} = \begin{cases} \frac{t_l^2 \delta_l^4 / \|\hat{\mathbf{x}}_l\|^2}{\delta_l^2(t_l^2+\delta_l^2) - t_l^2(\mathbf{1}^T \mathbf{X}_l^{-1}\mathbf{1})} (x_l^{(f)})^2, & \text{if } l_1 = l, \text{ f=1,}\ldots\text{,F,} \\ 0, & \text{otherwise,} \end{cases}$$

$$\mathbf{T}^{-1} + \widetilde{\mathbf{T}} = \text{Diag}\left\{ \frac{t_l^2 \delta_l^2}{t_l^2 + \delta_l^2} + \frac{t_l^4 \delta_l^2 (\mathbf{1}^T \mathbf{X}_l^{-1}\mathbf{1})/(t_l^2+\delta_l^2)}{\delta_l^2(t_l^2+\delta_l^2) - t_l^2(\mathbf{1}^T \mathbf{X}_l^{-1}\mathbf{1})},\ l = 1, \ldots, L \right\}.$$

Then, the results in Theorem 4.5 follow from letting

$$R_{l,1} = \frac{t_l^2 \delta_l^4 / \|\hat{\mathbf{x}}_l\|^2}{\delta_l^2(t_l^2+\delta_l^2) - t_l^2(\mathbf{1}^T \mathbf{X}_l^{-1}\mathbf{1})},$$

$$R_{l,2} = \frac{t_l^2 \delta_l^2}{t_l^2 + \delta_l^2} + \frac{t_l^4 \delta_l^2 (\mathbf{1}^T \mathbf{X}_l^{-1}\mathbf{1})/(t_l^2+\delta_l^2)}{\delta_l^2(t_l^2+\delta_l^2) - t_l^2(\mathbf{1}^T \mathbf{X}_l^{-1}\mathbf{1})}.$$

This completes the second part of the proof.

D Proof of Theorem 4.6

First, note that

$$\mathbf{M}^T\mathbf{w}^k = \left[\begin{array}{ccc:c} (\widetilde{\mathbf{b}}^{(1)})^T & & & \\ & \ddots & & \\ & & (\widetilde{\mathbf{b}}^{(F)})^T & \\ \hdashline -(\mathbf{a}_1^{(1)})^T & & & \\ & \ddots & & \\ & & -(\mathbf{a}_1^{(F)})^T & \\ \hdashline & \vdots & & \\ \hdashline -(\mathbf{a}_L^{(1)})^T & & & \\ & \ddots & & \\ & & -(\mathbf{a}_L^{(F)})^T & \\ \hdashline & & & \mathbf{1}_L \end{array}\right] \left[\begin{array}{c} \mathbf{w}^{(1)} \\ \vdots \\ \mathbf{w}^{(F)} \\ \hdashline w \end{array}\right] = \left[\begin{array}{c} (\widetilde{\mathbf{b}}^{(1)})^T\mathbf{w}^{(1)} \\ \vdots \\ (\widetilde{\mathbf{b}}^{(F)})^T\mathbf{w}^{(F)} \\ \hdashline -(\mathbf{a}_1^{(1)})^T\mathbf{w}^{(1)} \\ \vdots \\ -(\mathbf{a}_1^{(F)})^T\mathbf{w}^{(F)} \\ \hdashline \vdots \\ \hdashline -(\mathbf{a}_L^{(1)})^T\mathbf{w}^{(1)} \\ \vdots \\ -(\mathbf{a}_L^{(F)})^T\mathbf{w}^{(F)} \\ \hdashline w\mathbf{1}_L \end{array}\right] = \left[\begin{array}{c} w^{(1)}_{\mathrm{Src}(1)} \\ \vdots \\ w^{(F)}_{\mathrm{Src}(F)} \\ \hdashline w^{(1)}_{\mathrm{Rx}(1)} - w^{(1)}_{\mathrm{Tx}(1)} \\ \vdots \\ w^{(F)}_{\mathrm{Rx}(1)} - w^{(F)}_{\mathrm{Tx}(1)} \\ \hdashline \vdots \\ \hdashline w^{(1)}_{\mathrm{Rx}(L)} - w^{(1)}_{\mathrm{Tx}(L)} \\ \vdots \\ w^{(F)}_{\mathrm{Rx}(L)} - w^{(F)}_{\mathrm{Tx}(L)} \\ \hdashline w\mathbf{1}_L \end{array}\right],$$

where the last equality holds due to the special structure of $\widetilde{\mathbf{b}}^{(f)}$ and $\mathbf{a}_l^{(F)}$. More specifically, observe that $\widetilde{\mathbf{b}}^{(f)}$ is simply a unit vector where all entries are zeros except for a "1" at the entry corresponding to the node $\mathrm{Src}(f)$. Thus, we have $(\widetilde{\mathbf{b}}^{(f)})^T\mathbf{w}^{(f)} = w^{(f)}_{\mathrm{Src}(f)}$. Likewise, $\mathbf{a}_l^{(f)}$ has two non-zero entries (a "1" corresponding to node $\mathrm{Tx}(l)$ and a "-1" corresponding to node $\mathrm{Rx}(l)$) or only one non-zero entry when one end point of link l happens to be $\mathrm{Dst}(f)$. Thus, we have $-(\mathbf{a}_l^{(f)})^T\mathbf{w}^{(f)} = w^{(f)}_{\mathrm{Rx}(l)} - w^{(f)}_{\mathrm{Tx}(l)}$ (recall that we have defined $w^{(f)}_{\mathrm{Dst}(f)} = 0$). Hence,

$$(\nabla f(\mathbf{y}^k) + \mathbf{M}^T\mathbf{w}^k)_i = \begin{cases} -\mu U_f'(s_f) - \frac{1}{s_f} + w^{(f)}_{\mathrm{Src}(f)} & \text{if } 1 \le i = f \le F, \\ \frac{1}{\delta_l} - \frac{1}{x_l^{(f)}} + w^{(f)}_{\mathrm{Rx}(l)} - w^{(f)}_{\mathrm{Tx}(l)} & \text{if } i = (l+1)F + f, \\ \frac{1}{\delta_l} - \frac{1}{t_l} + w, & \text{if } i = (L+1)F + l. \end{cases} \tag{89}$$

Recall that $\mathbf{H}_k^{-1}$ has the following blockwise structure:

$$\mathbf{H}_k^{-1} = \left[\begin{array}{c:ccc} \widehat{\mathbf{S}} & & & \\ \hdashline & \widehat{\mathbf{X}}_1 & & \widehat{\mathbf{C}}_1 \\ & & \ddots & \vdots \\ & & \widehat{\mathbf{X}}_L & \widehat{\mathbf{C}}_L \\ & \widehat{\mathbf{C}}_1^T & \cdots \; \widehat{\mathbf{C}}_L^T & \widehat{\mathbf{T}} \end{array}\right].$$

Thus, we have that $\Delta\mathbf{y}^k = -\mathbf{H}_k^{-1}(\nabla f(\mathbf{y}^k) + \mathbf{M}^T\mathbf{w}^k)$ can be partitioned into three parts, each of which corresponds to one particular case in (89). For the entries in the first part, since $\mathbf{S}^{-1}$ is diagonal, we

have

$$(\Delta \mathbf{y}^k)_i = \frac{s_f(\mu s_f U_f'(s_f) + 1 - s_f w^{(f)}_{\mathrm{Src}(f)})}{1 - \mu s_f^2 U_f''(s_f)}, \quad \text{if } 1 \le i = f \le F. \tag{90}$$

In the second part, we have that $\Delta \mathbf{y}^k$ contains L blocks in the following form:

$$-\widehat{\mathbf{X}}_l \begin{bmatrix} \frac{1}{\delta_l} - \frac{1}{x_l^{(1)}} + w^{(1)}_{\mathrm{Rx}(l)} - w^{(1)}_{\mathrm{Tx}(l)} \\ \vdots \\ \frac{1}{\delta_l} - \frac{1}{x_l^{(F)}} + w^{(F)}_{\mathrm{Rx}(l)} - w^{(F)}_{\mathrm{Tx}(l)} \end{bmatrix} - \widehat{\mathbf{C}}_l \begin{bmatrix} \frac{1}{\delta_1} - \frac{1}{t_1} + w \\ \vdots \\ \frac{1}{\delta_L} - \frac{1}{t_L} + w \end{bmatrix}, \quad l = 1, \ldots, L.$$

Using the structural result of $\widehat{\mathbf{X}}_l$ and $\widehat{\mathbf{C}}_l$ in Theorem 4.5, we have that the i-th entry of $\Delta \mathbf{y}^k$, where $i = (l+1)F + f$, can be computed as

$$(\Delta \mathbf{y}^k)_i = \left(x_l^{(f)}\right)^2 \left[\left(1 - \frac{R_{l,1}}{\|\widehat{\mathbf{x}}_l\|^2} (x_l^{(f)})^2\right) \left(\frac{1}{x_l^{(f)}} - \frac{1}{\delta_l} + w^{(f)}_{\mathrm{Tx}(l)} - w^{(f)}_{\mathrm{Rx}(l)}\right) + R_{l,1}(x_l^{(f)})^2 \left(\frac{1}{t_l} - \frac{1}{\delta_l} - w\right) - \sum_{f'=1, f' \neq f}^{F} \frac{R_{l,1}}{\|\widehat{\mathbf{x}}_l\|^2} (x_l^{(f')})^2 \left(\frac{1}{x_l^{(f')}} - \frac{1}{\delta_l} + w^{(f')}_{\mathrm{Tx}(l)} - w^{(f')}_{\mathrm{Rx}(l)}\right) \right], \quad \forall l, f. \tag{91}$$

The third part of $\Delta \mathbf{y}^k$ is given by

$$\sum_{l=1}^{L} (-\widehat{\mathbf{C}}_l^T) \begin{bmatrix} \frac{1}{\delta_l} - \frac{1}{x_l^{(1)}} + w^{(1)}_{\mathrm{Rx}(l)} - w^{(1)}_{\mathrm{Tx}(l)} \\ \vdots \\ \frac{1}{\delta_l} - \frac{1}{x_l^{(F)}} + w^{(F)}_{\mathrm{Rx}(l)} - w^{(F)}_{\mathrm{Tx}(l)} \end{bmatrix} - \widehat{\mathbf{T}} \begin{bmatrix} \frac{1}{\delta_1} - \frac{1}{t_1} + w \\ \vdots \\ \frac{1}{\delta_L} - \frac{1}{t_L} + w \end{bmatrix}.$$

Using the structural result of $\widehat{\mathbf{C}}_l$ and $\widehat{\mathbf{T}}$ in Theorem 4.5, we have that the i-th entry of $\Delta \mathbf{y}^k$, where $i = (L+1)F + f$, can be computed as:

$$(\Delta \mathbf{y}^k)_i = R_{l,1} \left[\sum_{f=1}^{F} (x_l^{(f)})^2 \left(\frac{1}{x_l^{(f)}} - \frac{1}{\delta_l} + w^{(f)}_{\mathrm{Tx}(l)} - w^{(f)}_{\mathrm{Rx}(l)}\right) \right] + R_{l,2} \left(\frac{1}{t_l} - \frac{1}{\delta_l} - w\right). \tag{92}$$

Note that (90), (91), and (92) are the same as (33), (34), and (35), respectively. This completes the proof.

E Proof of Lemma 4.9

First, note that $\widetilde{\mathbf{H}}_k \succ 0$ because $f_\mu(\mathbf{y})$ is convex. It then follows that $(\mathbf{\Lambda}_k + \alpha\overline{\mathbf{\Omega}}) - (\alpha\overline{\mathbf{\Omega}} - \mathbf{\Omega}_k)$ is positive definite since $\widetilde{\mathbf{H}}_k = (\mathbf{\Lambda}_k + \alpha\overline{\mathbf{\Omega}}) - (\alpha\overline{\mathbf{\Omega}} - \mathbf{\Omega}_k)$. Next, we check the positive definiteness of $(\mathbf{\Lambda}_k + \alpha\overline{\mathbf{\Omega}}) + (\alpha\overline{\mathbf{\Omega}} - \mathbf{\Omega}_k)$. Note that

$$(\mathbf{\Lambda}_k + \alpha\overline{\mathbf{\Omega}}) + (\alpha\overline{\mathbf{\Omega}} - \mathbf{\Omega}_k) = \mathbf{\Lambda}_k + 2\alpha\overline{\mathbf{\Omega}}_k - \mathbf{\Omega}_k. \tag{93}$$

Therefore, for $(\mathbf{\Lambda}_k + \alpha\overline{\mathbf{\Omega}}) + (\alpha\overline{\mathbf{\Omega}} - \mathbf{\Omega}_k)$ to be positive definite, it suffices that $\mathbf{\Lambda}_k$ and $2\alpha\overline{\mathbf{\Omega}}_k - \mathbf{\Omega}_k$ are both positive definite. First, from the definition of $\mathbf{\Lambda}_k$, (14)–(17), and (18), we have that all diagonal entries in the diagonal matrix $\mathbf{\Lambda}_k$ are positive. Hence, $\mathbf{\Lambda}_k \succ 0$. On the other hand, by the definitions of $\overline{\mathbf{\Omega}}_k$ and $\mathbf{\Omega}_k$, we have that the entries of each row in $2\alpha\overline{\mathbf{\Omega}}_k - \mathbf{\Omega}_k$ satisfy

$$\begin{aligned}&(2\alpha\overline{\mathbf{\Omega}}_k - \mathbf{\Omega}_k)_{ii} - \sum_{j\neq i} |(2\alpha\overline{\mathbf{\Omega}}_k - \mathbf{\Omega}_k)_{ij}| \\ &= (2\alpha - 1)\sum_{j\neq i} |(\mathbf{\Omega}_k)_{ij}| > 0, \quad \text{for } \alpha > \frac{1}{2}.\end{aligned}$$

Also, it is clear from the definitions of $\overline{\mathbf{\Omega}}_k$ and $\mathbf{\Omega}_k$ that $(2\alpha\overline{\mathbf{\Omega}}_k - \mathbf{\Omega}_k)_{ii} > 0$. Thus, $2\alpha\overline{\mathbf{\Omega}}_k - \mathbf{\Omega}_k$ is diagonally dominant and hence positive definite. Therefore, $\mathbf{\Lambda}_k + 2\alpha\overline{\mathbf{\Omega}}_k - \mathbf{\Omega}_k$ is also positive definite, and the proof is complete.

F Proof of Lemma 4.10

To establish Lemma 4.10, we need the following result [24, Theorem 2.3]:

Lemma F.1. *Let* $\mathbf{A} = \mathbf{M}_1 - \mathbf{N}_1 = \mathbf{M}_2 - \mathbf{N}_2$ *be two splittings of* $\mathbf{A}$*, where* $\mathbf{A}^{-1} \succeq 0$*,* $\mathbf{M}_1^{-1} \succeq 0$*, and* $\mathbf{M}_2^{-1} \succeq 0$*. If* $\mathbf{M}_1^{-1} \succeq \mathbf{M}_2^{-1}$*, then* $\rho(\mathbf{M}_1^{-1}\mathbf{N}_1) \leq \rho(\mathbf{M}_2^{-1}\mathbf{N}_2)$*.*

Now, for $\frac{1}{2} < \alpha_1 \leq \alpha_2$, since $\mathbf{\Lambda}_k + \alpha\overline{\mathbf{\Omega}}_k$ is diagonal, we have that

$$\begin{aligned}&(\mathbf{\Lambda}_k + \alpha_2\overline{\mathbf{\Omega}}_k)_{ii} - (\mathbf{\Lambda}_k + \alpha_1\overline{\mathbf{\Omega}}_k)_{ii} \\ &= (\alpha_2 - \alpha_1)\sum_{j\neq i} |(\mathbf{\Omega}_k)_{ij}| > 0. \end{aligned} \tag{94}$$

Also, since $((\mathbf{\Lambda}_k + \alpha\overline{\mathbf{\Omega}}_k)^{-1})_{ii} = 1/(\mathbf{\Lambda}_k + \alpha\overline{\mathbf{\Omega}}_k)_{ii}$ (from the diagonal property again), Eq. (94) implies that $(\mathbf{\Lambda}_k + \alpha_1\overline{\mathbf{\Omega}}_k)^{-1} \succeq (\mathbf{\Lambda}_k + \alpha_2\overline{\mathbf{\Omega}}_k)^{-1}$. Thus, Lemma 4.10 simply follows from Lemma F.1, and the proof is complete.

G Proof of Theorem 4.11

To establish the result in Theorem 4.11, we need to compute the element-wise expansion of the proposed matrix-splitting scheme in (40), i.e.,

$$\Delta\mathbf{y}_{m+1}^k = (\mathbf{\Lambda}_k + \alpha\overline{\mathbf{\Omega}}_k)^{-1}(\alpha\overline{\mathbf{\Omega}}_k - \mathbf{\Omega}_k)\Delta\mathbf{y}_m^k + (\mathbf{\Lambda}_k + \alpha\overline{\mathbf{\Omega}}_k)^{-1}(-\nabla f_\mu(\mathbf{y}^k) - \mathbf{M}^T\mathbf{w}^{(k)}).$$

Recall that in Appendix G, by using the second-order properties of $\mathbf{a}_l$ and $\widetilde{\mathbf{b}}^{(f)}$, we have derived that

$$-(\nabla f(\mathbf{y}^k) + \mathbf{M}^T\mathbf{w}^{(k)})_j = \begin{cases} \frac{1}{x_l^{(f)}} - \frac{1}{\delta_l} + w_{\mathrm{Tx}(l)}^{(f)} - w_{\mathrm{Rx}(l)}^{(f)} & \text{if } j = (l+1)F + f, \\ \frac{1}{t_l} - \frac{1}{\delta_l} - w, & \text{if } j = (L+1)F + l. \end{cases}$$

So, we now need to examine the structure of $(\mathbf{\Lambda}_k + \alpha\overline{\mathbf{\Omega}}_k)$ and $(\alpha\overline{\mathbf{\Omega}}_k - \mathbf{\Omega}_k)$. First, from the definition of $\overline{\mathbf{H}}_k$, we have that

$$(\mathbf{\Lambda}_k)_{jj} = \begin{cases} \frac{1}{\delta_2^l} + \frac{1}{(x_l^{(f)})^2}, & \text{if } j = (l-1)F + f,\ l = 1, \ldots, L,\ f = 1, \ldots, F, \\ \frac{1}{t_i^2} + \sum_{l=1}^{L} \frac{1}{\delta_l^2}(C_l^{(i)})^2, & \text{if } j = LF + i,\ i = 1, \ldots, I. \end{cases}$$

Notice that each diagonal entry in $\overline{\mathbf{\Omega}}_k$ is the 1-norm of the corresponding non-diagonal row entries in $\overline{\mathbf{H}}_k$. Thus, from the definition of $\overline{\mathbf{H}}_k$, we have that

$$(\overline{\mathbf{\Omega}}_k)_{jj} = \begin{cases} \frac{F-1}{\delta_l^2} + \frac{1}{\delta_l^2}\sum_{i=1}^{I} C_l^{(i)}, & \text{if } j = (l-1)F + f,\ l = 1, \ldots, L,\ f = 1, \ldots, F, \\ \sum_{l=1}^{L}\left(\frac{FC_l^{(i)}}{\delta_l^2} + \sum_{i'=1,\neq i}^{I} \frac{C_l^{(i)}C_l^{(i')}}{\delta_l^2}\right), & \text{if } j = LF + i,\ i = 1, \ldots, I. \end{cases}$$

Therefore, we have

$$\begin{aligned}(\mathbf{\Lambda}_k + \alpha\overline{\mathbf{\Omega}}_k)_{jj} &= \begin{cases} \frac{1}{\delta_l^2} + \frac{1}{(x_l^{(f)})^2} + \alpha\left(\frac{F-1}{\delta_l^2} + \frac{1}{\delta_l^2}\sum_{i=1}^{I} C_l^{(i)}\right), & \text{if } j = (l-1)F + f, \\ \frac{1}{t_i^2} + \sum_{l=1}^{L}\frac{1}{\delta_l^2}(C_l^{(i)})^2 + \alpha\sum_{l=1}^{L}\left(\frac{FC_l^{(i)}}{\delta_l^2} + \sum_{i'=1,\neq i}^{I}\frac{C_l^{(i)}C_l^{(i')}}{\delta_l^2}\right), & \text{if } j = LF + i. \end{cases} \\ &= \begin{cases} \frac{1}{\delta_l^2}\left[1 + \alpha\left((F-1) + \sum_{i=1}^{I} C_l^{(i)}\right)\right] + \frac{1}{(x_l^{(f)})^2}, & \text{if } j = (l-1)F + f, \\ \frac{1}{t_i^2} + \sum_{l=1}^{L}\frac{1}{\delta_l^2}\left[(C_l^{(i)})^2 + \alpha C_l^{(i)}\left(F + \sum_{i'=1,\neq i}^{I} C_l^{(i')}\right)\right], & \text{if } j = LF + i. \end{cases}\end{aligned}$$

It can be seen that the last two expressions are exactly $P_{l,1}^{(f)}$ and $Q_{i,1}$ as stated in the theorem.

Next, consider the entries in $(\alpha\overline{\mathbf{\Omega}}_k - \mathbf{\Omega}_k)$. From the definition of $\overline{\mathbf{H}}_k$, we have that $(\alpha\overline{\mathbf{\Omega}}_k - \mathbf{\Omega}_k)$ also has the "arrow head" structure:

$$(\alpha\overline{\mathbf{\Omega}}_k - \mathbf{\Omega}_k) = \begin{bmatrix} \mathbf{\Omega}_{\mathbf{X}}^{(1)} & & & \mathbf{\Omega}_{\mathbf{C}}^{(1)} \\ & \ddots & & \vdots \\ & & \mathbf{\Omega}_{\mathbf{X}}^{(L)} & \mathbf{\Omega}_{\mathbf{C}}^{(L)} \\ (\mathbf{\Omega}_{\mathbf{C}}^{(1)})^T & \cdots & (\mathbf{\Omega}_{\mathbf{C}}^{(L)})^T & \mathbf{\Omega}_{\mathbf{T}} \end{bmatrix},$$

where $\mathbf{\Omega}_{\mathbf{X}}^{(l)}$, $\mathbf{\Omega}_{\mathbf{C}}^{(l)}$, and $\mathbf{\Omega}_{\mathbf{T}}$ are, respectively, defined as:

$$\begin{aligned}\mathbf{\Omega}_{\mathbf{X}}^{(l)} &= \text{Diag}\left\{\alpha\left(\frac{F-1}{\delta_l^2} + \sum_{i=1}^{I}\frac{C_l^{(i)}}{\delta_l^2}\right)\right\} - \frac{1}{\delta_l^2}\mathbf{1}_F\mathbf{1}_F^T, \\ \mathbf{\Omega}_{\mathbf{C}}^{(l)} &= -\mathbf{C}_l = \frac{1}{\delta_l^2}\mathbf{1}_I\mathbf{c}_l^T, \\ \mathbf{\Omega}_{\mathbf{T}} &= \text{Diag}\left\{\alpha\sum_{l=1}^{L}\frac{1}{\delta_l^2}\left(C_l^i(F + \sum_{i'=1,\neq i}^{I} C_l^{(i')})\right)\right\} - \sum_{l=1}^{L}\left(\frac{1}{\delta_l^2}\right)\mathbf{c}_l\mathbf{c}_l^T.\end{aligned}$$

Therefore, we have

$$((\alpha\overline{\mathbf{\Omega}}_k - \mathbf{\Omega}_k)\Delta\mathbf{y}_m^k)_j = \begin{bmatrix} \mathbf{\Omega}_{\mathbf{X}}^{(1)}\mathbf{x}_1 + \mathbf{\Omega}_{\mathbf{C}}^{(1)}\mathbf{t} \\ \vdots \\ \mathbf{\Omega}_{\mathbf{X}}^{(L)}\mathbf{x}_L + \mathbf{\Omega}_{\mathbf{C}}^{(L)}\mathbf{t} \\ \sum_{l=1}^{L}(\mathbf{\Omega}_{\mathbf{C}})^T\mathbf{x}_l + \mathbf{\Omega}_{\mathbf{T}}\mathbf{t} \end{bmatrix}_j$$

$$= \begin{cases} \frac{1}{\delta_l^2}\left[\alpha\left(F - 1 + \sum_{i=1}^{I} C_l^{(i)}\right)\Delta x_l^{(f)}(m) - \sum_{f'=1,\neq f}^{F}\Delta x_l^{(f')}(m) + \sum_{i=1}^{I} C_l^{(i)}\Delta t_i(m)\right], & \text{if } j = (l-1)F + f \\ \frac{1}{\delta_l^2}\left[\sum_{l=1}^{L} C_l^{(i)}\left(\sum_{f=1}^{F}\left(\Delta x_l^{(f)}(m) + \alpha F\right) + \sum_{i'=1,\neq i}^{I} C_l^{(i')}(\alpha\Delta t_i(m) - \Delta t_{i'}(m))\right)\right], & \text{if } j = LF + i. \end{cases}$$

It can be seen that the last two expressions are exactly $P_{l,2}^{(f)}$ and $Q_{i,2}$ as stated in the theorem. This completes the proof.

H Proof Proposition 4.12

Proposition 4.12 is also based on the same matrix-splitting idea used in Theorem 4.11, i.e., it iteratively solves the linear equation system $\mathbf{z}_j = \overline{\mathbf{H}}_k^{-1}\widehat{\mathbf{m}}_j$ by using

$$\mathbf{z}_{j,m+1} = (\mathbf{\Lambda}_k + \alpha\overline{\mathbf{\Omega}}_k)^{-1}(\alpha\overline{\mathbf{\Omega}}_k - \mathbf{\Omega}_k)\mathbf{z}_{j,m} + (\mathbf{\Omega}_k + \alpha\overline{\mathbf{\Omega}}_k)^{-1}\widehat{\mathbf{m}}_j. \tag{95}$$

As a result, we immediately have that the updates of $z_{j,m}^{(x_l^{(f)})}$ and $z_{j,m}^{(t_i)}$ should follow the same form as in (41) and (42). Moreover, since $(\mathbf{\Lambda}_k + \alpha\overline{\mathbf{\Omega}}_k)^{-1}$ is unchanged in computing $\mathbf{z}_{j,m}$, $\forall m$, we have that the same terms $P_{l,1}^{(f)}$ and $Q_{i,1}$ should also appear in (50) and (51). By replacing $\Delta x_l^{(f)}(m)$ and $\Delta t_i(m)$ in (41) and (42) with $z_{j,m}^{(x_l^{(f)})}$ and $z_{j,m}^{(t_i)}$, respectively, we obtain (52) and (54).

To establish the expressions in (53) and (55), note that $\widehat{\mathbf{m}}_j$ (the j-th column of $\widehat{\mathbf{M}}$) has the following special sparsity structure of in as follows:

1) For $j = 1, \ldots, F(N-1)$:

$$(\widehat{\mathbf{m}}_j)_k = \begin{cases} 1 & \text{if } k = (l-1)F + f \text{ and } \mathrm{Tx}(l) = j - (f-1)(N-1), \\ -1 & \text{if } k = (l-1)F + f \text{ and } \mathrm{Rx}(l) = j - (f-1)(N-1), \\ 0 & \text{otherwise.} \end{cases} \tag{96}$$

2) For $j = F(N-1) + 1$:

$$(\widehat{\mathbf{m}}_j)_k = \begin{cases} 1 & k = LF + 1, \ldots, LF + I, \\ 0 & \text{otherwise.} \end{cases} \tag{97}$$

Substituting the entries in (96) and (97) into $(\mathbf{\Omega}_k + \alpha\overline{\mathbf{\Omega}}_k)^{-1}\widehat{\mathbf{m}}_j$ and matching the corresponding blocks in $\mathbf{z}_{j,m}$, we obtain (53) and (55). This completes the proof.

I Proof of Proposition 4.13

Let $\mathbf{m}_{j_1}$ denote the j_1-th row in $\widehat{\mathbf{M}}$, which has the special sparsity structure as follows:

1) For $j_1 = 1, \ldots, F(N-1)$:

$$(\mathbf{m}_{j_1})_k = \begin{cases} 1 & \text{if } k = (l-1)F + f \text{ and } \mathrm{Tx}(l) = j_1 - (f-1)(N-1), \\ -1 & \text{if } k = (l-1)F + f \text{ and } \mathrm{Rx}(l) = j_1 - (f-1)(N-1), \\ 0 & \text{otherwise.} \end{cases} \tag{98}$$

2) For $j_1 = F(N-1)+1$:

$$(\mathbf{m}_{j_1})_k = \begin{cases} 1 & k = LF+1, \ldots, LF+I, \\ 0 & \text{otherwise.} \end{cases} \tag{99}$$

Then, the result in Proposition 4.15 follows from multiplying $\mathbf{m}_{j_1}$ and $\mathbf{z}_{j_2}$, $j_2 = 1, \ldots, (N-1)F+1$, and retaining only the non-zero entries. This completes the proof.

J Proof of Theorem 4.17

To show (67), we need to compute the element-wise expansion of (64). First, note that $(\mathbf{\Pi}_k + \alpha\overline{\mathbf{\Psi}}_k)$ is diagonal, and so its inverse can be easily computed by taking the inverse of each diagonal entry. Therefore, we start by computing each diagonal entry in $(\mathbf{\Pi}_k + \alpha\overline{\mathbf{\Psi}}_k)$. Since $\mathbf{\Pi}_k$ contains the main diagonal of $\mathbf{M}\mathbf{H}_k^{-1}\mathbf{M}^T$, similar to [11, Theorem 7], we obtain that

$$(\mathbf{\Pi}_k)_{ii} = \begin{cases} \sum_{\Phi(n)} (x_l^{(f)})^2 \Big(1 - \frac{R_{l,1}(x_l^{(f)})^2}{\|\widehat{\mathbf{x}}_l\|^2}\Big) + \frac{1}{-\mu U_f''(s_f) + \frac{1}{(s_f)^2}} & \text{if } n = \mathrm{Src}(f), \\ \sum_{\Phi(n)} (x_l^{(f)})^2 \Big(1 - \frac{R_{l,1}(x_l^{(f)})^2}{\|\widehat{\mathbf{x}}_l\|^2}\Big) & \text{if } n \neq \mathrm{Src}(f), \end{cases} \tag{100}$$

where the index i satisfies $i = (f-1)(N-1) + \beta_f(n)$. Notice that each diagonal entry in $\overline{\mathbf{\Psi}}_k$ is the row sum of non-diagonal entries in $\mathbf{M}\mathbf{H}_k^{-1}\mathbf{M}^T$. Thus, similar to [11, Theorem 7], we can derive that

$$(\overline{\mathbf{\Psi}})_{ii} = \sum_{l \in \Phi(n) \setminus \Psi(n,f)} (x_l^{(f)})^2 \Big(1 - \frac{R_{l,1}(x_l^{(f)})^2}{\|\widehat{\mathbf{x}}_l\|^2}\Big) + \sum_{f'=1, \neq f}^{F} \sum_{l \in \Psi(n,f')} \frac{R_{l,1}(x_l^{(f)} x_l^{(f')})^2}{\|\widehat{\mathbf{x}}_l\|^2} + \sum_{l \in \Phi(n)} |R_{l,2}| (x_l^{(f)})^2. \tag{101}$$

Therefore, using the indicator function $\mathbb{1}_{\Psi(n,f)}$ and adding (100) and (101), we obtain that, for $i = 1, \ldots, (N-1)F$,

$$(\boldsymbol{\Pi}_k + \alpha\overline{\boldsymbol{\Psi}}_k)_{ii} = \begin{cases} \sum_{l\in\Phi(n)}[1+\alpha(1-\mathbb{1}_{\Psi(n,f)}(l))](x_l^{(f)})^2\Big(1-\frac{R_{l,1}(x_l^{(f)})^2}{\|\widehat{\mathbf{x}}_l\|^2}\Big)+ \\ \quad \sum_{f'=1,\neq f}^{F}\Big(\sum_{l\in\Psi(n,f')}\frac{\alpha R_{l,1}(x_l^{(f)}x_l^{(f')})^2}{\|\widehat{\mathbf{x}}_l\|^2}\Big)+\alpha\sum_{l\in\Phi(n)}|R_{l,2}|(x_l^{(f)})^2 & \text{if } n\neq \mathrm{Src}(f), \\ \sum_{l\in\Phi(n)}[1+\alpha(1-\mathbb{1}_{\Psi(n,f)}(l))](x_l^{(f)})^2\Big(1-\frac{R_{l,1}(x_l^{(f)})^2}{\|\widehat{\mathbf{x}}_l\|^2}\Big)+ \\ \quad \sum_{f'=1,\neq f}^{F}\Big(\sum_{l\in\Psi(n,f')}\frac{\alpha R_{l,1}(x_l^{(f)}x_l^{(f')})^2}{\|\widehat{\mathbf{x}}_l\|^2}\Big)+\alpha\sum_{l\in\Phi(n)}|R_{l,2}|(x_l^{(f)})^2 \\ \quad +\frac{1}{-\mu U_f''(s_f)+\frac{1}{(s_f)^2}} & \text{if } n=\mathrm{Src}(f), \end{cases}$$

which is the same as the definition of $U_n^{(f)}(m)$ in (69). For $i = (N-1)F+1$, it is easy to see that

$$(\boldsymbol{\Pi}_k + \alpha\overline{\boldsymbol{\Psi}}_k)_{ii} = \sum_{l=1}^{L}|R_{l,3}| + \alpha\Big[\sum_{f=1}^{F}\sum_{\substack{n=1,\\ n\neq\mathrm{Dst}(f)}}^{N}|R_{l,2}|(x_l^{(f)})^2\Big],$$

which is the same as the definition of $\rho(m)$ in (74).

Next, consider the entries in $(\alpha\overline{\boldsymbol{\Psi}}_k - \boldsymbol{\Psi}_k)\mathbf{w}^k$. Recall that the matrix $\mathbf{M}\mathbf{H}_k^{-1}\mathbf{M}^T$ has a partitioned matrix structure. Thus, the vector $(\alpha\overline{\boldsymbol{\Psi}}_k - \boldsymbol{\Psi}_k)\mathbf{w}^k$ can be partitioned into F blocks plus one additional row, where each block is of the form

$$((\alpha\overline{\boldsymbol{\Psi}}_k - \boldsymbol{\Psi}_k)\mathbf{w}^k)_f = -\mathbf{R}_f\mathbf{w}^f + \sum_{f'=1,\neq f}^{F}\mathbf{G}_{ff'}\mathbf{w}^{(f')} + \boldsymbol{\zeta}_f\tilde{w}, \forall f, \tag{102}$$

where $\mathbf{R}_f$ is obtained by replacing the main diagonal of $\mathbf{D}_f - \widehat{\mathbf{D}}_f$ (cf. [11, Theorem 7]) with the corresponding entries in $-\alpha\overline{\boldsymbol{\Psi}}_k$, and $\boldsymbol{\zeta}_f$ is the f-th block in $(\sum_{l=1}^{L}\mathbf{A}_l\widehat{\mathbf{C}}_l)\mathbf{1}$. Hence, by computing the entries in $-\mathbf{R}_f\mathbf{w}^f$ and noting the special structure in $\mathbf{R}_f$, where it only contains entries 1, -1, and 0, we have

$$\begin{aligned}(-\mathbf{R}_f\mathbf{w}^f)_n = &\sum_{l\in\mathcal{I}(n)}(x_l^{(f)})^2\Big(1-\frac{R_{l,1}(x_l^{(f)})^2}{\|\widehat{\mathbf{x}}_l\|^2}\Big)(w^{(f)}_{\mathrm{Tx}(l)}-\alpha w^{(f)}_{\mathrm{Rx}(l)})+ \\ &\sum_{l\in\mathcal{O}(n)\setminus\Psi(n,f)}(x_l^{(f)})^2\Big(1-\frac{R_{l,1}(x_l^{(f)})^2}{\|\widehat{\mathbf{x}}_l\|^2}\Big)(w^{(f)}_{\mathrm{Rx}(l)}-\alpha w^{(f)}_{\mathrm{Tx}(l)})- \\ &\sum_{f'=1,\neq f}^{F}\Big(\sum_{l\in\Psi(n,f')}\frac{\alpha R_{l,1}(x_l^{(f)}x_l^{(f')})^2}{\|\widehat{\mathbf{x}}_l\|^2}\Big)w_n^f+\alpha\sum_{l\in\Phi(n)}|R_{l,2}|(x_l^{(f)})^2\tilde{w},\end{aligned}$$

which is the same as the definition of $V_{n,1}^{(f)}(m)$ in (70). Similarly, by computing the entries in

$\sum_{f'=1,\neq f}^{F} \mathbf{G}_{ff'}\mathbf{w}^{(f')}$, we have

$$\Big(\sum_{f'=1,\neq f}^{F} \mathbf{G}_{ff'}\mathbf{w}^{(f')}\Big)_n = \sum_{f'=1,\neq f}^{F} \Big(\Big(\sum_{l\in\mathcal{O}(n)} \frac{R_{l,2}(x_l^{(f)}x_l^{(f')})^2}{\|\widehat{\mathbf{x}}_l\|^2} - \sum_{l\in\mathcal{I}(n)} \frac{R_{l,2}(x_l^{(f)}x_l^{(f')})^2}{\|\widehat{\mathbf{x}}_l\|^2}\Big)(w_{\mathrm{Tx}(l)}^{(f')} - w_{\mathrm{Rx}(l)}^{(f')})\Big) - \alpha \sum_{l\in\Phi(n)} R_{l,2}(x_l^{(f)})^2,$$

which is the same as the definition of $V_{n,2}^{(f)}(m)$ in (71). For the additional row in $(\alpha\overline{\mathbf{\Psi}}_k - \mathbf{\Psi}_k)\mathbf{w}^k$, it can be seen that

$$\begin{aligned}
&((\alpha\overline{\mathbf{\Psi}}_k - \mathbf{\Psi}_k)\mathbf{w}^k)_{(N-1)F+1} \\
&= \sum_{f=1}^{F}(-\boldsymbol{\zeta}_f^T)\mathbf{w}^{(f)} + \left[\sum_{l=1}^{L} R_{l,3} + \alpha\sum_{f=1}^{F}\sum_{\substack{n=1\\ n\neq \mathrm{Dst}(f)}}^{N} |R_{l,2}|(x_l^{(f)})^2\right]\tilde{w} \\
&= \sum_{l=1}^{L} R_{l,3} + \sum_{f=1}^{F}\sum_{\substack{n=1\\ n\neq \mathrm{Dst}(f)}}^{N} (R_{l,2}w_n^{(f)} + \alpha|R_{l,2}|\tilde{w})(x_l^{(f)})^2,
\end{aligned}$$

which is the same as the definition of $\sigma(m)$ in (75).

Finally, consider the term $\mathbf{M}\mathbf{H}_k^{-1}\nabla f(\mathbf{y}^k)$. Note that $\mathbf{M}\mathbf{H}_k^{-1}\nabla f(\mathbf{y}^k)$ can be decomposed into

$$\mathbf{M}\mathbf{H}_k^{-1}\nabla f(\mathbf{y}^k) = -\begin{bmatrix} \widetilde{\mathbf{B}}\widehat{\mathbf{S}}\nabla_{\mathbf{s}} f_\mu(\mathbf{y}^k) + \sum_{l=1}^{L} \mathbf{A}_l[\widehat{\mathbf{X}}_l\nabla_{\mathbf{x}_l} f_\mu(\mathbf{y}^k) + \widehat{\mathbf{C}}_l\nabla_{\mathbf{t}} f_\mu(\mathbf{y}^k)] \\ \mathbf{1}^T\left[\left(\sum_{l=1}^{L}\widehat{\mathbf{C}}_l^T\nabla_{\mathbf{x}_l} f_\mu(\mathbf{y}^k) + \widehat{\mathbf{T}}\nabla_{\mathbf{t}} f_\mu(\mathbf{y}^k)\right)\right] \end{bmatrix},$$

Accordingly, consider first the term $\widetilde{\mathbf{B}}\mathbf{S}^{-1}\nabla_{\mathbf{s}} f(\widetilde{\mathbf{y}}^k)$. Using the diagonal structure of $\widetilde{\mathbf{B}}$ and $\mathbf{S}$, it is easy to obtain that

$$(\widetilde{\mathbf{B}}\mathbf{S}^{-1}\nabla_{\mathbf{s}} f(\widetilde{\mathbf{y}}^k))_n^{(f)} = \begin{cases} \frac{s_f(1+\mu s_f U_f'(s_f))}{\mu s_f^2 U_f''(s_f) - 1} & \text{if } n = \mathrm{Src}(f), \\ 0 & \text{otherwise.} \end{cases}$$

Recalling that $\widetilde{\mathbf{H}}_k^{-1}$ can be decomposed into a diagonal matrix and a rank-one update matrix, we have

$$-\mathbf{A}_l[\widehat{\mathbf{X}}_l\nabla_{\mathbf{x}_l} f_\mu(\mathbf{y}^k) + \widehat{\mathbf{C}}_l\nabla_{\mathbf{t}} f_\mu(\mathbf{y}^k)]) = -\mathbf{A}_l\mathrm{Diag}\left\{(x_l^{(1)})^2,\ldots,(x_l^{(F)})^2\right\}\begin{bmatrix} \frac{1}{\delta_l} - \frac{1}{x_l^{(1)}} \\ \vdots \\ \frac{1}{\delta_l} - \frac{1}{x_l^{(F)}} \end{bmatrix} +$$

$$\frac{R_{l,1}}{\|\widehat{\mathbf{x}}_l\|^2}\mathbf{A}_l\begin{bmatrix} (x_l^{(1)})^4 & \cdots & (x_l^{(1)}x_l^{(F)})^2 \\ \vdots & \ddots & \vdots \\ (x_l^{(F)}x_l^{(1)})^2 & \cdots & (x_l^{(F)})^4 \end{bmatrix}\begin{bmatrix} \frac{1}{\delta_l} - \frac{1}{x_l^{(1)}} \\ \vdots \\ \frac{1}{\delta_l} - \frac{1}{x_l^{(F)}} \end{bmatrix} - \mathbf{A}_l\widehat{\mathbf{C}}_l\begin{bmatrix} \frac{1}{\delta_l} - \frac{1}{t_1} \\ \vdots \\ \frac{1}{\delta_l} - \frac{1}{t_L} \end{bmatrix}.$$

Hence, computing each term in the above decomposition, then adding $\widetilde{\mathbf{B}}\mathbf{S}^{-1}\nabla_{\mathbf{s}}f(\widetilde{\mathbf{y}}^k)$, and then summing over all l, we obtain that

$$
\begin{aligned}
&-\left(\widetilde{\mathbf{B}}\widehat{\mathbf{S}}\nabla_{\mathbf{s}}f_\mu(\mathbf{y}^k)+\sum_{l=1}^{L}\mathbf{A}_l[\widehat{\mathbf{X}}_l\nabla_{\mathbf{x}_l}f_\mu(\mathbf{y}^k)+\widehat{\mathbf{C}}_l\nabla_{\mathbf{t}}f_\mu(\mathbf{y}^k)]\right)\\
&=\begin{cases}
\sum_{l\in\mathcal{O}(n)}(x_l^{(f)}-\frac{(x_l^{(f)})^2}{\delta_l})-\sum_{l\in\mathcal{I}(n)}\frac{R_{l,2}}{\|\hat{x}_l\|^2}(x_l^{(f)})^2\sum_{f'=1}^{F}(x_l^{(f')}-\frac{(x_l^{(f')})^2}{\delta_l})\\
\qquad-(\sum_{l\in\mathcal{O}(n)}R_{l,2}(x_l^{(f)})^2(\frac{1}{\delta_l}+\frac{1}{t_l})-\sum_{l\in\mathcal{I}(n)}R_{l,2}(x_l^{(f)})^2(\frac{1}{\delta_l}+\frac{1}{t_l})), & \text{if } n\neq \mathrm{Src}(f),\\
\sum_{l\in\mathcal{O}(n)}(x_l^{(f)}-\frac{(x_l^{(f)})^2}{\delta_l})-\sum_{l\in\mathcal{I}(n)}\frac{R_{l,2}}{\|\hat{x}_l\|^2}(x_l^{(f)})^2\sum_{f'=1}^{F}(x_l^{(f')}-\frac{(x_l^{(f')})^2}{\delta_l})\\
\qquad+\frac{s_f(1+\mu s_f U_f'(s_f))}{\mu s_f^2U_f''(s_f)-1}\\
\qquad-(\sum_{l\in\mathcal{O}(n)}R_{l,2}(x_l^{(f)})^2(\frac{1}{\delta_l}+\frac{1}{t_l})-\sum_{l\in\mathcal{I}(n)}R_{l,2}(x_l^{(f)})^2(\frac{1}{\delta_l}+\frac{1}{t_l})), & \text{if } n=\mathrm{Src}(f),
\end{cases}
\end{aligned}
$$

which is the same as the definition of $W_n^{(f)}(m)$ given in (73). Thus, the result in (67) simply follows from Proposition 4.16.

Next, we consider the row $\mathbf{1}^T\left[\left(\sum_{l=1}^{L}\widehat{\mathbf{C}}_l^T\nabla_{\mathbf{x}_l}f_\mu(\mathbf{y}^k)+\widehat{\mathbf{T}}\nabla_{\mathbf{t}}f_\mu(\mathbf{y}^k)\right)\right]$, for which it is easy to derive that

$$
\mathbf{1}^T\sum_{l=1}^{L}(-\widehat{\mathbf{C}}_l^T)\nabla_{\mathbf{x}_l}f_\mu(\mathbf{y}^k)=\sum_{l=1}^{L}\sum_{f=1}^{F}R_{l,2}\left(x_l^{(f)}-\frac{(x_l^{(f)})^2}{\delta_l}\right), \tag{103}
$$

$$
\mathbf{1}^T(-\widehat{\mathbf{T}}\nabla_{\mathbf{t}}f_\mu(\mathbf{y}^k))=\sum_{l=1}^{L}R_{l,3}\left(\frac{1}{t_l}+\frac{1}{\delta_l}\right). \tag{104}
$$

Adding (103) and (104), we have

$$
\mathbf{1}^T\left[\left(\sum_{l=1}^{L}\widehat{\mathbf{C}}_l^T\nabla_{\mathbf{x}_l}f_\mu(\mathbf{y}^k)+\widehat{\mathbf{T}}\nabla_{\mathbf{t}}f_\mu(\mathbf{y}^k)\right)\right]=\sum_{l=1}^{L}\left[R_{l,3}\left(\frac{1}{t_l}+\frac{1}{\delta_l}\right)+\sum_{f=1}^{F}R_{l,2}\left(x_l^{(f)}-\frac{(x_l^{(f)})^2}{\delta_l}\right)\right],
$$

which is same as the definition of $\tau(m)$ in (76). This completes the proof.

K Proof of Proposition 4.18

Define the following three vectors: $\Delta\mathbf{s}\triangleq[\Delta s_1,\ldots,\Delta s_F]^T$, $\Delta\mathbf{x}_l=[\Delta x_l^{(1)},\ldots,\Delta x_l^{(F)}]^T$, and $\Delta\mathbf{t}=[\Delta t_1,\ldots,\Delta t_I]$. Also, from the "arrow head" structure of $\widetilde{\mathbf{H}}_k$, we have

$$
(\Delta\widetilde{\mathbf{y}})^T\widetilde{\mathbf{H}}_k\Delta\widetilde{\mathbf{y}}=(\Delta\mathbf{s})^T\mathbf{S}\Delta\mathbf{s}+\sum_{l=1}^{L}(\Delta\mathbf{x}_l)^T\mathbf{X}_l\Delta\mathbf{x}_l+2\sum_{l=1}^{L}(\Delta\mathbf{x}_l)^T\mathbf{C}_l\Delta\mathbf{t}+(\Delta\mathbf{t})^T\mathbf{T}\Delta\mathbf{t}. \tag{105}
$$

Now, consider first $(\Delta\mathbf{s})^T\mathbf{S}\Delta\mathbf{s}$, which, due to the diagonal structure of $\widetilde{\mathbf{H}}_k$, can be simply computed as

$$
(\Delta\mathbf{s})^T\mathbf{S}\Delta\mathbf{s}=\sum_{f=1}^{F}(\Delta s_f)^2\Big(-\mu U_f''(s_f)+\frac{1}{(s_f)^2}\Big). \tag{106}
$$

Next, consider $\sum_{l=1}^{L}(\Delta\mathbf{x}_l)^T\mathbf{X}_l\Delta\mathbf{x}_l$. Recall that $\mathbf{X}_l$ can be further decomposed into a diagonal matrix plus a rank-one update matrix. Thus, we have

$$\begin{aligned}
&\sum_{l=1}^{L}(\Delta\mathbf{x}_l)^T\mathbf{X}_l\Delta\mathbf{x}_l \\
&= \sum_{l=1}^{L}\begin{bmatrix} x_l^{(1)} & \cdots & x_l^{(F)} \end{bmatrix}\left(\begin{bmatrix} \frac{1}{(x_l^{(1)})^2} & & \\ & \ddots & \\ & & \frac{1}{(x_l^{(F)})^2} \end{bmatrix} + \frac{1}{\delta_l}\mathbf{1}\cdot\mathbf{1}^T\right)\begin{bmatrix} x_l^{(1)} \\ \vdots \\ x_l^{(F)} \end{bmatrix} \\
&= \sum_{l=1}^{L}\Big[\sum_{f=1}^{F}\Big(\frac{\Delta x_l^{(f)}}{x_l^{(f)}}\Big)^2 + \frac{1}{\delta_l}\Big(\sum_{f=1}^{F}\Delta x_l^{(f)}\Big)^2\Big]. \qquad (107)
\end{aligned}$$

For the term $2\sum_{l=1}^{L}(\Delta\mathbf{x}_l)^T\mathbf{C}_l\Delta\mathbf{t}$, we have

$$\begin{aligned}
&2\sum_{l=1}^{L}(\Delta\mathbf{x}_l)^T\mathbf{C}_l\Delta\mathbf{t} = 2\sum_{l=1}^{L}(\Delta\mathbf{x}_l)^T\left(-\frac{\mathbf{1}_F\mathbf{c}_l^T}{\delta_l^2}\right)\Delta\mathbf{t} = -2\sum_{l=1}^{L}\frac{((\Delta\mathbf{x}_l)^T\mathbf{1}_F)(\mathbf{c}_l^T\Delta\mathbf{t})}{\delta_l^2} \\
&= -2\sum_{l=1}^{L}\frac{1}{\delta_l^2}\Big(\sum_{f=1}^{F}\Delta x_l^{(f)}\Big)\Big(\sum_{i=1}^{I}C_l^{(i)}\Delta t_i\Big). \qquad (108)
\end{aligned}$$

For the last term $(\Delta\mathbf{t})^T\mathbf{T}\Delta\mathbf{t}$, we have

$$\begin{aligned}
&(\Delta\mathbf{t})^T\mathbf{T}\Delta\mathbf{t} = (\Delta\mathbf{t})^T\Big(\mathrm{Diag}\left\{\frac{1}{t_1^2},\ldots,\frac{1}{t_I^2}\right\} + \sum_{l=1}^{L}\frac{1}{\delta_l^2}\mathbf{c}_l\mathbf{c}_l^T\Big)\Delta\mathbf{t} \\
&\sum_{i=1}^{I}\Big(\frac{\Delta t_i}{t_i}\Big)^2 + \sum_{l=1}^{L}\frac{1}{\delta_l^2}\Big(\sum_{i=1}^{I}C_l^{(i)}\Delta t_i\Big)^2. \qquad (109)
\end{aligned}$$

Thus, adding (106) and (107) gives the desired result in Proposition 4.18, and the proof is complete.

References

[1] X. Lin, N. B. Shroff, and R. Srikant, "A tutorial on cross-layer optimization in wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 8, pp. 1452–1463, Aug. 2006.

[2] L. Tassiulas and A. Ephremides, "Stability properties of constrained queuing systems and scheduling policies for maximum throughput in multihop radio networks," *IEEE Trans. Autom. Control*, vol. 37, no. 2, pp. 466–478, Mar. 1993.

[3] X. Lin and N. B. Shroff, "Joint rate control and scheduling in multihop wireless networks," in *Proc. IEEE CDC*, Atlantis, Paradise Island, Bahamas, Dec. 2006, pp. 1484–1489.

[4] M. J. Neely, E. Modiano, and C. E. Rohrs, "Dynamic power allocation and routing for time varying wireless networks," *IEEE J. Sel. Areas Commun.*, vol. 23, no. 1, pp. 89–103, Jan. 2005.

[5] A. Eryilmaz and R. Srikant, "Fair resource allocation in wireless networks using queue-length-based scheduling and congestion control," in *Proc. IEEE INFOCOM*, Miami, FL, Mar. 2005, pp. 1804–1814.

[6] X. Lin and N. B. Shroff, "The impact of imperfect scheduling on cross-layer congetion control in wireless networks," *IEEE/ACM Trans. Netw.*, vol. 14, no. 2, pp. 302–315, Apr. 2006.

[7] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty, *Nonlinear Programming: Theory and Algorithms*, 3rd ed. New York, NY: John Wiley & Sons Inc., 2006.

[8] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, UK: Cambridge University Press, 2004.

[9] A. Jadbabaie, A. Ozdaglar, and M. Zargham, "A distributed Newton method for network optimization," in *Proc. IEEE Conference on Decision and Control (CDC)*, Shanghai, China, Dec. 16-18, 2009.

[10] E. Wei, A. Ozdaglar, and A. Jadbabaie, "A distributed Newton method for network utitlity maximization," in *Proc. IEEE Conference on Decision and Control (CDC)*, Atlanta, GA, Dec. 15-17, 2010.

[11] J. Liu and H. D. Sherali, "A distributed Newton's method for joint multi-hop routing and flow control: Theory and algorithm," in *Proc. IEEE INFOCOM*, Orlando, FL, Mar. 25-30, 2012, pp. 2489–2497.

[12] D. P. Bertsekas and E. M. Gafni, "Projected Newton methods and optimization of multicommodity flows," *IEEE Trans. Autom. Control*, vol. 28, no. 12, pp. 1090–1096, Dec. 1983.

[13] J. G. Klincewicz, "A Newton method for convex separable network flow problems," *Networks*, vol. 13, no. 3, pp. 427–442, Mar. 1983.

[14] A. Zymnis, N. Trichakis, S. Boyd, and D. ONeill, "An interior-point method for large scale network utility maximization," in *Proc. Allerton Conference on Communication, Control, and Computing*, Monticello, IL, Sep. 26-28, 2007.

[15] D. Bickson, Y. Tock, O. Shental, and D. Dolev, "Polynomial linear programming with Gaussian belief propagation," in *Proc. Allerton Conference on Communication, Control, and Computing*, Monticello, IL, Sep. 23-26, 2008, pp. 895–901.

[16] D. Bickson, Y. Tock, A. Zymnis, S. Boyd, and D. Dolev, "Distributed large scale network utility maximization," in *Proc. IEEE International Symposium on Information Theory (ISIT)*, Seoul, Korea, Jun.28–Jul.3, 2009, pp. 829–833.

[17] Y. Nesterov and A. Nemirovskii, *Interior-Point Polynomial Algorithms in Convex Programming*, 3rd ed. Philadelphia, PA: SIAM, 2001.

[18] D. Bickson, "Gaussian belief propagation: Theory and application," Ph.D. dissertation, Hebrew University of Jerusalem, 2009.

[19] F. R. K. Chung, *Spectral Graph Theory*. Providence, RI: American Mathematical Society, 1994.

[20] M. S. Bazaraa, J. J. Jarvis, and H. D. Sherali, *Linear Programming and Network Flows*, 4th ed. New York: John Wiley & Sons Inc., 2010.

[21] J. Liu, C. H. Xia, N. B. Shroff, and H. D. Sherali, "Distributed cross-layer optimization in wireless networks: A second-order approach," *Technical Report, Dept. of ECE, Ohio State University*, Jul. 2012. [Online]. Available: http://www2.ece.ohio-state.edu/~liu/publications/DNewton_Wireless.pdf

[22] J. Liu and H. D. Sherali, "A distributed Newton's method for joint multi-hop routing and flow control: Theory and algorithm," *Technical Report, Dept. of ECE, Ohio State University*, Jul. 2011. [Online]. Available: http://www2.ece.ohio-state.edu/~liu/publications/DNewton.pdf

[23] R. A. Horn and C. R. Johnson, *Matrix Analysis*. New York, NY: Cambridge University Press, 1990.

[24] Z. I. Woznicki, "Matrix splitting principles," *International Journal of Mathematics and Mathematical Sciences*, vol. 28, no. 5, pp. 251–284, May 2001.

[25] F. P. Kelly, A. K. Malullo, and D. K. H. Tan, "Rate control in communications networks: Shadow prices, proportional fairness and stability," *Journal of the Operational Research Society*, vol. 49, pp. 237–252, 1998.