

Research Article

A Distributed Cross-Layer Optimization Method for Multicast in Interference-Limited Multihop Wireless Networks

Mohammad H. Amerimehr,¹ Babak H. Khalaj,¹ and Pedro M. Crespo²

¹Department of Electrical Engineering, Sharif University of Technology, Tehran 11365-8639, Iran

²Centro de Estudios e Investigaciones Técnicas de Gipuzkoa (CEIT) and Tecnun, University of Navarra, Manuel de Lardizabal 15, 20018 San Sebastian, Spain

Correspondence should be addressed to Babak H. Khalaj, khalaj@sharif.edu

Received 2 February 2008; Accepted 6 June 2008

Recommended by Lawrence Yeung

We consider joint optimization of data routing and resource allocation in multicast multihop wireless networks where interference between links is taken into account. The use of network coding in such scenarios leads to a nonconvex optimization problem. By applying the probability collectives (PCs) technique the original problem is turned into a new problem which is convex over probability distributions. The resulting problem is then further decomposed into a data routing subproblem at network layer and a power allocation subproblem at physical layer in order to achieve a cross-layer distributed solution for the whole range of SINR values. The proposed approach is also extended to minimum cost multicast problems and routing problems based on multicommodity flow and single Steiner tree, resulting in new distributed algorithms for such problems.

Copyright © 2008 Mohammad H. Amerimehr et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

1. INTRODUCTION

In this paper, we consider the problem of resource allocation in wireless multihop networks, where a source node is simultaneously transmitting common information to a set of destinations via relay nodes. In contrast with the wired networks, link capacities are not fixed rather in general can be functions of communication resources such as transmit power. Hence achieving optimal throughput requires joint optimization of data flow routing and resource allocation.

As shown by Ahlswede et al. in [1], data routing can be performed efficiently through network coding where nodes are allowed to mix information and send certain functions of received data on their outgoing links. Network coding was originally developed for wired networks (more precisely, a network with fixed capacity and lossless links). In such networks, multicast capacity (which is an upper bound for multicast throughput) is always achievable by network coding, whereas in general it may not be achievable with routing [1, 2]. Li et al. showed in [3] that linear coding usually suffices in achieving the maximum rate. A polynomial-time algorithm to achieve the maximum multicast rate in directed

networks is proposed in [4]. Alternatively, Ho et al. in [5] designed a distributed algorithm based on random network coding. Li et al. in [6] formulated the problem of computing optimal throughput as a linear optimization problem and proposed a distributed algorithm to solve this problem.

The problem of joint optimization of data flow routing and resource allocation has also been investigated by different researchers. In wireline networks, where multicast routing scenarios is considered, the problem is equivalent to the Steiner tree problem which is known to be NP-hard [7]. However, by the use of network coding, this problem can be solved efficiently in a distributed manner. The main idea is to assume a convex (concave) cost (utility) objective function so that the problem can be formulated as a convex optimization problem and be solved efficiently by using Lagrange relaxation and subgradient methods [8, 9]. A game theoretic solution to this problem has also been proposed by Bhadra et al. in [10]. However, solving the aforementioned problem is more difficult in wireless networks. Since link capacity is in general a function of link power, achieving the optimal result requires consideration of both network and physical layers. Finding optimal multicast routing in routed

wireless network is an NP-hard problem [11]. The joint optimization of routing and resource allocation based on multicommodity is investigated in [12, 13] where distributed cross-layer solutions are offered. As shown in [12], with the assumption that link capacity is a concave and increasing function of the communication resources allocated to the link, the problem will become a convex optimization problem which can be solved efficiently by dual decomposition. In [13], CDMA wireless networks are considered and it is shown that for relatively high values of SINR, this problem can also be turned into a convex optimization problem. In addition, based on single Steiner tree routing, Cheng et al. in [14] addressed energy-efficient routing in multihop ad-hoc wireless networks. They proposed a distributed algorithm for optimal routing in interference-free networks through proper power allocation to each link.

Recently, the problem of joint optimization of data flow routing and resource allocation in wireless networks when network coding is used in the network layer has also become of interest. Minimum cost multicast problem has been considered in [15, 16]. They formulated the problem as a convex optimization problem using time sharing to eliminate interference between links and offered a centralized cross-layer approach. Yuan et al. in [17] have offered a cross-layer optimization framework to achieve optimal throughput in wireless networks. They showed that by use of time (frequency) sharing or applying logarithmic transformation at high SINR values, as well as assuming concave utility function, a distributed solution can be obtained via dual decomposition. An analogous approach has also been adopted in [18].

The main goal of this paper is to extend the scope of such problems to high interference scenarios (low SINR) as well as nonconvex (concave) cost (utility) objective functions, where we deal with a nonconvex optimization problem and traditional optimization techniques are not applicable any more. Our approach will focus on cases where network coding or routing is applied in network layer. When network coding is applied in network layer, we use max flow-min cut theorem [1] to formulate the problem as a nonlinear constrained problem. Then by the use of the new probability collectives (PCs) method, the problem is turned into a convex optimization problem over the space of probability distribution functions. Consequently, it will be shown that the new problem can be decomposed into two subproblems that are coupled via a set of Lagrangian multipliers: data routing in network layer and power management in physical layer. Subsequently, distributed cross-layer algorithms are proposed in order to obtain the solution in the new framework. It should be mentioned that one of the main features of our method is providing a distributed and parallel solution, in contrast with traditional centralized schemes for solving nonlinear constrained optimization problems (e.g., projection method [19]) or evolutionary algorithms (e.g., genetic algorithms [20] or particle swarm optimizations [21]). This feature provides the possibility of applying this method to multihop wireless networks without an infrastructure support.

Finally, extension of the proposed method to routing problems based on traditional multicommodity and single Steiner tree is also presented and it is shown that as expected, network coding-based solutions can generally lead to better performance in comparison with routing-based solutions.

The organization of the paper is as follows. Section 2 describes the original optimization problem addressed in the paper and in Section 3, it is shown how by use of probability collectives the problem will be transformed into a convex form and subsequently decomposed to achieve a fully distributed solution. Instead of maximizing the throughput, in some scenarios the goal is to minimize a cost function (e.g., energy) while fulfilling a certain achievable multicast throughput. Section 4 extends the methods described in Section 3 to such min-cost multicast problems. Subsequently, extension of the proposed approach to single tree solutions is provided in Section 5. Simulation results are presented in Section 6, and finally Section 7 concludes the paper. A summary of probability collectives optimization scheme is also presented in the appendix.

2. PROBLEM FORMULATION

In traditional routing, nodes are only allowed to replicate and forward received data packets. In such networks, each data unit is transmitted in a tree-structure. This tree includes a path from source to each destination known as the *Steiner tree*. Maximum achievable throughput can be obtained by computing the maximum number of pairwise capacity-disjoint trees resulting in a centralized process with high computational complexity. In order to reduce the complexity, two suboptimal solutions can be applied: *multicommodity flow routing* and *single Steiner tree routing* [22]. In multicommodity flow routing, multicast session is treated as multiple unicast sessions and dedicated bit rate resources are allocated to different destinations. In this case, the multicast rate r is feasible if there is a flow vector between a source and each destination with a rate equal or greater than r , and also the sum of these flows at each link does not exceed the link capacity. As will be shown in Section 6, this property simplifies the problem formulation and enables us to achieve a distributed solution. Another special case of the general routing problem is to send information via a single Steiner tree. Although this case is of special importance in networks modelled by unlimited capacity links (e.g., wireless optical networks [23]), it is still applicable in limited capacity networks if link capacity of each tree is not less than r , and data can be sent to destination at rate r via such tree. By use of network coding, the multicast rate r is feasible if and only if there is a flow vector between source and each destination (called *conceptual flow*) with a rate equal or greater than r , and also max of these flows (called *max of flows* or *link flow*) does not exceed the link capacity. In this paper, we will consider both approaches and provide corresponding optimization solutions in each scenario.

A data network can be represented by a directed graph $G = (V, E)$, where the vertex set V and the edge set E denote the nodes and links, respectively. An $s - d$ flow with value

r is a length- $|E|$ nonnegative vector x satisfying the *flow conservation* constraint:

$$\sum_{l \in O(n)} x_l - \sum_{l \in I(n)} x_l = \begin{cases} r, & n = s, \\ 0, & \forall n \in V - \{s, d_i\}, \\ -r, & n = d_i, \end{cases} \quad (1)$$

where $I(n)$ and $O(n)$ are defined as the set of incoming and outgoing links at node n , respectively. Also, s , d_i , and N denote sender (source), i th receiver (destination) and the number of receivers, respectively. Let f_l , c_l , and $e_{i,l}$, respectively, denote flow, capacity, and conceptual flow associated with i th destination of link l . In order to achieve a tractable solution for the problem addressed in this paper, which is inherently difficult to solve due to its inherent nonconvex structure, it is assumed that network topology is time invariant, in other words nodes are static, not moving, and connected via fixed links. Such assumption is valid in quasistationary wireless mesh networks as well as static ad-hoc networks. However, in multihop wireless networks, due to interference, each achievable link rate not only depends on the power allocated to the link itself, but also on the power allocated to other links. Consequently, achievable rate of a link may be formulated as a function of SINR defined as

$$\text{SINR}_l = \frac{G_{ll} p_l}{\sum_{j \neq l} G_{lj} p_j + \sigma_l^2}. \quad (2)$$

For example, in CDMA wireless networks the achievable rate can be defined as

$$c_l(p) = \log(1 + \text{SINR}_l), \quad (3)$$

where G_{ll} , p_l , and σ_l^2 are the link gain, power, and noise variance, respectively, and G_{lj} is the interference gain from link j to link l . The power constraints for each link and node can then be constrained as

$$0 \leq p_l \leq p_{l,\max}, \quad \sum_{l \in O(n)} p_l \leq \bar{P}_{n,\max}. \quad (4)$$

Consequently, the maximum utility derived by a feasible multicast rate can be achieved by the following optimization problem:

$$\text{maximize } U(r) \equiv \text{minimize } -U(r) \quad (5)$$

subject to: $r \in [r_{\min}, r_{\max}]$,

$$\sum_{l \in O(n)} e_{i,l} - \sum_{l \in I(n)} e_{i,l} = s_{i,n} = \begin{cases} r, & n = s, \\ 0, & \forall n \in V - \{s, d_i\}, \\ -r, & n = d_i, \end{cases} \quad (6)$$

$$e_{i,l} \geq 0 \quad \forall i, \forall l \in E, \quad (7)$$

$$f_l \geq 0 \quad \forall l \in E, \quad (8)$$

$$e_{i,l} \leq f_l \quad \forall i, \forall l \in E, \quad (9)$$

$$f_l \leq c_l(p) \quad \forall l \in E, \quad (10)$$

$$0 \leq p_l \leq p_{l,\max} \quad \forall l \in E, \quad (11)$$

$$\sum_{l \in O(n)} p_l \leq \bar{P}_{n,\max} \quad \forall n \in V. \quad (12)$$

3. OBTAINING A DISTRIBUTED PC-BASED SOLUTION

In order to obtain a manageable solution for the problem presented in Section 2, we adopt the probability collectives (PCs) optimization method. As will be shown subsequently, by proper use of PC approach, the problem will be transformed into a convex form and subsequently decomposed to achieve a fully distributed solution. A brief introduction to PC and its key concepts such as Maxent Lagrangian is presented in the appendix.

3.1. The general framework of PC-based optimization

Lets assume that the variables r , f_l , $e_{i,l}$, and p_l take a finite number of values in the ranges $[r_{\min}, r_{\max}]$, $[0, r_{\max}]$, $[0, r_{\max}]$, and $[0, p_{l,\max}]$, respectively. In this way, it is ensured that the solutions obtained at each step satisfy the constraints (5), (7), (8), and (11). It should be noted that the other constraints are already included in Maxent Lagrangian and also all feasible values for $e_{i,l}$ and f_l are in the range $[0, r_{\max}]$. The equality constraint (6) can be rewritten as

$$\sum_{l \in O(n)} e_{i,l} - \sum_{l \in I(n)} e_{i,l} \leq s_{i,n}. \quad (13)$$

Since the above constraint ensures that the source node injects a flow of at most r in the network, at each intermediate node the outgoing flow is less than the incoming flow and each of the receivers receive at a flow rate greater than or equal to r . This is possible, if and only if, the flow conservation constraint (6) is satisfied. This is an important issue since we assumed that all constraints are of the form of nonequalities. Let $q_r^{(t)}$, $q_{e_{i,l}}^{(t)}$, $q_{f_l}^{(t)}$, and $q_{p_l}^{(t)}$ denote probability distributions associated with variables r , $e_{i,l}$, f_l , and p_l , at step t , respectively. By expanding the Lagrangian, the following convex optimization problem will be obtained:

$$\begin{aligned} \text{minimize}_{q_r, q_{f_l}, q_{e_{i,l}}, q_{p_l}} E \left\{ -U(r) + \sum_{i \in D} \sum_{n \in V} \mu_{i,n} \left(\sum_{l \in O(n)} e_{i,l} - \sum_{l \in I(n)} e_{i,l} - s_{i,n} \right) \right. \\ + \sum_{i \in D} \sum_{l \in E} \xi_{i,l} (e_{i,l} - f_l) \\ + \sum_{n \in V} \nu_n \left(\sum_{l \in O(n)} p_l - \bar{P}_{n,\max} \right) \\ \left. + \sum_{l \in E} \lambda_l (f_l - c_l) \right\} - T \sum_{i \in D} \sum_{l \in E} S(q_{e_{i,l}}) \\ - T \sum_{l \in E} S(q_{f_l}) - TS(q_r) - T \sum_{l \in E} S(q_{p_l}), \end{aligned} \quad (14)$$

where T and S are part of the PC optimization framework briefly described in the appendix. In addition, in order to reduce the number of equations, the constraints for nonnegative probabilities and unity probability distributions are not explicitly mentioned. Also, the time dependency of

probability distributions is assumed implicitly. Finally, the Lagrange multipliers are updated according to (A.9):

$$\mu_{i,n}^{(t+1)} = \left[\mu_{i,n}^{(t)} + \eta_{\mu_{i,n}} E \left(\sum_{l \in O(n)} e_{i,l} - \sum_{l \in I(n)} e_{i,l} - s_{i,n} \right) \right]^+, \quad (15)$$

$$\xi_{i,l}^{(t+1)} = [\xi_{i,l}^{(t)} + \eta_{\xi_{i,l}} E(e_{i,l} - f_l)]^+, \quad (16)$$

$$\nu_n^{(t+1)} = \left[\nu_n^{(t)} + \eta_{\nu_n} E \left(\sum_{l \in O(n)} p_l - \bar{P}_{n,\max} \right) \right]^+, \quad (17)$$

$$\lambda_l^{(t+1)} = [\lambda_l^{(t)} + \eta_{\lambda_l} E(f_l - c_l)]^+. \quad (18)$$

3.2. Problem decomposition

Subsequently, minimizing the Maxent Lagrangian can be decomposed into the following subproblems in network and physical layers, respectively, as follows:

$$\begin{aligned} & \underset{q_r, q_{e_{i,l}}, q_{f_l}}{\text{minimize}} E \left\{ -U(r) + \sum_{i \in D} \sum_{n \in V} \mu_{i,n} \left(\sum_{l \in O(n)} e_{i,l} - \sum_{l \in I(n)} e_{i,l} - s_{i,n} \right) \right. \\ & \quad \left. + \sum_{i \in D} \sum_{l \in E} \xi_{i,l} (e_{i,l} - f_l) + \sum_{l \in E} \lambda_l f_l \right\} \\ & \quad - T \sum_{i \in D} \sum_{l \in E} S(q_{e_{i,l}}) - T \sum_{l \in E} S(q_{f_l}) - TS(q_r), \\ & \underset{q_{p_l}}{\text{minimize}} E \left\{ \sum_{n \in V} \nu_n \left(\sum_{l \in O(n)} p_l - \bar{P}_{n,\max} \right) - \sum_{l \in E} \lambda_l c_l \right\} \\ & \quad - T \sum_{l \in E} S(q_{p_l}). \end{aligned} \quad (19)$$

The network layer subproblem can be further decomposed into a set of single variable subproblems as follows:

$$\begin{aligned} & \underset{q_r}{\text{minimize}} E \left\{ -U(r) + r \left(\sum_{i=1}^N \mu_{i,d_i} - \sum_{i=1}^N \mu_{i,s} \right) \right\} - TS(q_r), \\ & \underset{q_{e_{i,l}}}{\text{minimize}} E \{ e_{i,l} (\mu_{i,\text{head}(l)} - \mu_{i,\text{tail}(l)} + \xi_{i,l}) \} \\ & \quad - TS(q_{e_{i,l}}) \quad i \in D, l \in E, \\ & \underset{q_{f_l}}{\text{minimize}} E \left\{ f_l \left(\lambda_l - \sum_{i=1}^N \xi_{i,l} \right) \right\} - TS(q_{f_l}) \quad l \in E, \end{aligned} \quad (20)$$

where

$$\begin{aligned} \text{head}(l) &= \{n \mid n \in V \& l \in O(n)\}, \\ \text{tail}(l) &= \{n \mid n \in V \& l \in I(n)\}. \end{aligned} \quad (21)$$

The physical layer subproblem can also be decomposed into a set of the following subproblems at each link:

$$\underset{q_{p_l}}{\text{minimize}} E \left\{ \nu_{\text{head}(l)} p_l - \sum_{l \in E} \lambda_l c_l \right\} - TS(q_{p_l}) \quad l \in E. \quad (22)$$

By use of Newton updating scheme for subproblems (20)–(22), we will obtain updating rules similar to (A.7) for $q_r(x_i)$, $q_{e_{i,l}}(x_i)$, $q_{f_l}(x_i)$, and $q_{p_l}(x_i)$ where G is replaced by G_1 to G_4 , in each case as follows:

$$G_1 = -U(r) + r \left(\sum_{i=1}^N \mu_{i,d_i} - \sum_{i=1}^N \mu_{i,s} \right), \quad (23)$$

$$G_2 = e_{i,l} (\mu_{i,\text{head}(l)} - \mu_{i,\text{tail}(l)} + \xi_{i,l}), \quad (24)$$

$$G_3 = f_l \left(\lambda_l - \sum_{i=1}^N \xi_{i,l} \right), \quad (25)$$

$$G_4 = \nu_{\text{head}(l)} p_l - \sum_{l \in E} \lambda_l c_l. \quad (26)$$

3.3. Proposed distributed algorithm

The overall distributed algorithm is subsequently given by Algorithm 1

The “exact” convergence is achieved when all constraints are satisfied and the probability distributions converge to impulse function. However, in practice “approximate” convergence criteria can also be defined [24]. For example, if the following constraints are satisfied at iteration $t + 1$, then an “approximate” convergence is achieved

$$\begin{aligned} \|q_i^{(t+1)} - q_i^{(t)}\| &\leq \delta_i, \\ C_i(x^{*(t+1)}) &\leq \varepsilon_i, \end{aligned} \quad (27)$$

where C_i is a non-equally constraint of the form $C_i(x) \leq 0$, δ_i , and ε_i are sufficiently small positive scalars.

The aforementioned algorithm can consequently be performed in a distributed fashion: at network layer q_r , $q_{e_{i,l}}$, q_{f_l} , and $\xi_{i,l}$ are updated based on local information: updating q_r , $q_{e_{i,l}}$, and q_{f_l} needs only previous probability distributions associated with variables r , $e_{i,l}$, and f_l , respectively (see (23)–(25) and (A.7)). Also, $\xi_{i,l}$ can be updated by computing $E(e_{i,l} - f_l)$ requiring only probability distributions $q_{e_{i,l}}$ and q_{f_l} . $\mu_{i,n}$ can be updated at each node (except at the receivers) using probability distributions of flow and conceptual flows of the incoming and outgoing links. At the receivers, $E(r)$ should also be taken into account. Therefore, in step 2a, the source also broadcasts $E(r)$. At the physical layer, each link can calculate its expected capacity and broadcast $E(\lambda_l c_l)$ to other links. Consequently, each link can update its probability distribution based on (26).

The overall algorithm then works as follows: at network layer, at iteration t , each node n uses the previous probability distributions associated with its outgoing link (i.e., $q_{f_l}^{(t-1)}$, $q_{e_{i,l}}^{(t-1)}$, $l \in O(n)$) as well as Lagrange multipliers, $\mu_{i,n}^{(t-1)}$ and $\xi_{i,l}^{(t-1)}$, ($l \in O(n)$) in order to coordinate with other nodes and obtain new appropriate values for its outgoing links flows (i.e., $f_{i,l}^{*(t)}$). This procedure can be performed in parallel since each node uses previous probability distributions and Lagrange multipliers corresponding to its neighboring nodes (i.e., nodes that have at least a common link with this node). In a similar way, nodes at the physical layer update

(1) Initialize
 (a) Assign the starting probabilities for each variable, typically a uniform distribution over its possible values.
 (b) Set the parameters $\{T, \alpha, \eta\}$

(2) Optimize the Lagrangian
At network layer:
 (a) At the source node, q_r is updated according to (23) and (A.7). $E(r)$ is calculated and broadcasted to the network.
 (b) For each link, $q_{e,l}$ and $q_{f,l}$ is updated according to (24), (25), and (A.7).
 (c) Lagrange multipliers $\mu_{i,n}$ and $\xi_{i,l}$ are updated according to (15) and (16).
At physical layer:
 (d) q_{p_l} 's are updated according to (26) and (A.7) and broadcasted to the network.
 (e) Lagrange multipliers ν_n are updated based on (17).
Cross Layer Optimization:
 (f) Lagrange multipliers λ_i are updated based on (18).
 (g) T is decreased at the rate β ($T := \beta T, 0 < \beta < 1$)

(3) Repeat until convergence is achieved.

ALGORITHM 1

the probability distributions associated with their outgoing links power in order to achieve new appropriate values for link capacities. The two layers coordinate with each other in order to balance links flow and links capacities. Finally, the algorithm will continue until approximate convergence is achieved. In order to achieve approximate convergence, all the problem constraints (which can be rewritten in the form $C_i(x) \leq 0$, should not exceed a small specific positive value (i.e., $C_i(x) \leq \epsilon_i$) and for all probability distributions, we should have $\|q_i^{(t+1)} - q_i^{(t)}\| \leq \delta_i$. In other words, all constraints should be approximately satisfied and the probability distributions should converge to an approximate steady state condition. It is not hard to check that all the problem constraints can be calculated in a distributed fashion (see (6)–(12)) in appropriate node at physical or network layer. Therefore, at each step after updating probability distributions and achieving new appropriate values (i.e., $x_i^{*(t)}$), each node can calculate its related constraints and probability distributions in order to check if they meet the convergence conditions and subsequently announce it to the network. The algorithm will be terminated when each node achieves the aforementioned approximate convergence.

While the network layer tries to allocate appropriate flow (i.e., bandwidth) to each link in order to achieve an optimal multicast throughput, the physical layer assigns link powers in order to support the required bandwidths. Lagrange multipliers λ_l 's play an important rule in such coordination between layers. When (expected) capacity supported by physical layer is less than the expected flow of the link, λ_l is increased in order to enforce physical layer to increase link capacity by increasing link power and subsequently notifies network layer to decrease link flow. On the other hand, if physical layer assigns more bandwidth than is required in network layer, excess power is allocated by physical layer to the link. This effect will in turn cause interference to other links, resulting in a decrease in the capacity of other links. In this case, by decreasing Lagrange multipliers, physical

layer decreases link power and consequently the link capacity, while network layer realizes that it can inject more flow to this link. The optimal solution is achieved when link capacity and link flow become equal (if we are interested in maximum throughput, regardless of how much power is consumed it suffices that each link flow does not exceed link capacity). However, in the proposed method, since link powers as well as link flows are selected from a discrete set, these two values may not be equal in the final solution and the resulting link capacities are usually more than link flows. Lagrange multiplier λ_l can also be interpreted as the bandwidth cost of link l . Network layer tries to send data via links with relatively lower cost in order to minimize the total cost incurred, while physical layer tries to maximize the total benefit achieved by providing more bandwidth to network layer.

As mentioned in the appendix, the PC algorithm converges to at least a local minimum that satisfies the given constraints. Therefore, the proposed algorithm achieves a feasible multicast rate (corresponding to a local maximum of the utility function). The proposed method is more complex than traditional convex optimization problems since it requires updating a probability distribution (associated with each scalar variable) rather than a scalar value, resulting in a higher computational complexity as well as more memory space. However, this additional complexity is inevitable due to the nonconvexity of the original problem. It should be noted that it is possible to reduce this complexity by selecting variables from a smaller set, but this may result in further suboptimality.

4. EXTENSION TO MINIMUM COST MULTICAST

In Section 3, we considered joint optimization of data flow routing and link power adjustment in order to achieve the optimal throughput. Alternatively, we can investigate the problem of link power allocation in order to minimize a cost (e.g., total consumed power) while fulfilling a certain

achievable multicast throughput. This problem can be formulated as follows:

$$\begin{aligned}
& \text{minimize } \sum_{l \in E} g_l(\mathbf{p}), \\
& \text{subject to: } \sum_{l \in O(n)} e_{i,l} - \sum_{l \in I(n)} e_{i,l} = s_{i,n} \\
& \quad = \begin{cases} r_0, & n = s, \\ 0, & \forall n \in V - \{s, d_i\}, \\ -r_0, & n = d_i, \end{cases} \\
& e_{i,l} \geq 0 \quad \forall i, \forall l \in E, \\
& f_l \geq 0 \quad \forall l \in E, \\
& e_{i,l} \leq f_l \quad \forall i, \forall l \in E, \\
& f_l \leq c_l(\mathbf{p}) \quad \forall l \in E, \\
& 0 \leq p_l \leq p_{l,\max} \quad \forall l \in E, \\
& \sum_{l \in O(n)} p_l \leq \bar{P}_{n,\max} \quad \forall n \in V,
\end{aligned} \tag{28}$$

where $g_l(\mathbf{p})$ is an arbitrary (not necessarily convex) function of link powers. Following a similar approach as presented in Section 3, a distributed algorithm can be designed by decomposing the Maxent Lagrangian. In addition, we can modify the multicast rate optimization problem to maximization of a net utility function similar to [9] where the utility function can be defined as

$$U(r) = \sum_{l \in E} g_l(\mathbf{p}). \tag{29}$$

In aforementioned problems, we concentrated on finding the optimal data flow in network layer, rather than the code design problem. In order to establish a multicast session with network coding, it suffices to compute the appropriate data flow and then compute a code that determines the content of each link flow following the method presented in [4, 5]. Joint optimization of data routing and resource allocation using multicommodity flow can be formulated in a similar way, by replacing max flow with accumulated flow in the constraints. Therefore, the constraint $f_{i,l} \leq c_l$ should be replaced with $\sum_{i \in D} f_{i,l} \leq c_l$. Clearly, in this case, less flow can be dedicated to each destination, resulting in a suboptimal solution compared with the network coding-based solutions. In this respect, our solution can be considered as an extension of work in [13] to nonconvex cost functions. In addition, while in [13] only low-interference scenarios where link capacities are approximated by log (SINR) are taken into account, our approach does not assume such approximation and can consequently be applied in both low and high interference scenarios.

5. A SOLUTION BASED ON SINGLE TREE ROUTING

In earlier sections, we have offered a distributed algorithm for a general network by applying network coding at the network layer. Also, it has been shown that when routing is

used at the network layer, with some modifications, we can achieve a distributed solution by using multicommodity flow routing scheme. Another routing-based solution of interest is based on single Steiner tree. Although such solution is only suboptimal in relation to that of a general Steiner tree problem, it can be implemented in a distributed fashion with lower complexity. Therefore, in this section, we will also extend our method by presenting a solution based on single Steiner trees. We study both acyclic and general networks, where in each case, a Steiner tree is constructed through which data can be multicasted from source to the destinations.

5.1. Acyclic networks

First, we consider a network with no cycles (i.e., an acyclic network) and will address the general problem in Section 5.2. Consider an arbitrary subgraph $G' = (V', E')$ $V' \subseteq V$, $E' \subseteq E$. An indicator variable, e_l , is associated with each link defined as follows:

$$e_l = \begin{cases} 1, & l \in E', \\ 0, & l \notin E', \end{cases} \quad l \in E. \tag{30}$$

Note that a subgraph can be characterized by an *indicator vector*, \mathbf{e} , defined as

$$\mathbf{e} = [e_l], \quad \forall l \in E.$$

An intermediate node (a node which is neither a source nor a destination node) in optimum multicast subgraph should act as a relay node, that is, only retransmit received packets. Therefore, searching for optimum subgraphs can be restricted to subgraphs with such property.

Theorem 1. *A subgraph includes a path from source to each destination, if and only if, constraints (31)–(33) are satisfied*

$$\sum_{l \in O(s)} e_l > 0, \tag{31}$$

$$\begin{aligned}
& \sum_{l \in O(i)} e_l = 0 \& \sum_{l \in I(i)} e_l = 0 \\
& \text{or } \sum_{l \in O(i)} e_l > 0 \& \sum_{l \in I(i)} e_l > 0 \\
& \forall i \in E - \{s, d_1, d_2, \dots, d_n\},
\end{aligned} \tag{32}$$

$$\sum_{l \in I(d_i)} e_l > 0. \tag{33}$$

Proof. Assume a subgraph includes a path from a node to each destination, so it includes the source and one of its outgoing links and constraint (31) is satisfied. If an intermediate node included in the subgraph acts as a relay node, at least one of its outgoing links and one of its incoming links will be included in the subgraph. Otherwise, none of its links will be included in the subgraph. In both cases, constraint (32) is satisfied. The subgraph should include all destinations and at least one incoming link of each destination. Consequently, constraint (33) is also satisfied.

Satisfying constraints (31)–(33) ensures that the subgraph includes a path from source to each destination. Since the network has no cycles, if there is no path from source to a destination, it should make a cycle with some relay nodes and/or other destinations in order to satisfy constraint (31), contradicting the definition of an acyclic network. \square

Constraints (31)–(33) can be interpreted as follows: constraint (31) states that the source sends data packets to network via at least one of its outgoing links. Condition (32) states that intermediate nodes act as relay nodes and retransmit received packets. Constraint (33) insures that all destination nodes receive packets. Consequently, finding the optimal multicast subgraph can be performed via searching the set of subgraphs satisfying constraints (31)–(33). It should be noted that the minimum-cost subgraph has a tree structure corresponding to the minimum cost Steiner tree. Since the optimum subgraph includes a path from source to each destination, it comprises of a tree consisting of such paths. This tree is sufficient for transmitting information from source to receivers. Consequently, every other link in the optimum subgraph is redundant. A subgraph with minimum cost incurred is the optimal solution and can be formulated as follows:

$$\begin{aligned} & \text{minimize } \sum_{l \in E} e_l b_l, \\ & \text{subject to: } h_i(\mathbf{e}) = 0 \quad \forall i \in V, \end{aligned} \quad (34)$$

where $h_i(\mathbf{e})$ is defined as

$$\begin{aligned} h_s(\mathbf{e}) &= \begin{cases} 0, & \sum_{l \in O(s)} e_l > 0, \\ 1, & \text{Otherwise,} \end{cases} \\ h_i(\mathbf{e}) &= \begin{cases} 0, & \left(\sum_{l \in O(i)} e_l = 0 \& \sum_{l \in I(i)} e_l = 0 \right) \\ & \text{or } \left(\sum_{l \in \text{out}(i)} e_l > 0 \& \sum_{l \in \text{in}(i)} e_l > 0 \right), \\ 1, & \text{Otherwise,} \end{cases} \quad (35) \\ & \forall i \in E - \{S, d_1, d_2, \dots, d_N\}, \end{aligned}$$

$$h_{d_j}(\mathbf{e}) = \begin{cases} 0, & \sum_{l \in \text{in}(d_j)} e_l > 0, \\ 1, & \text{Otherwise,} \end{cases} \quad j = 1, 2, \dots, N.$$

By using the PC theory, the above problem can be solved as follows: a discrete probability distribution, q_{e_l} , is associated with each variable, e_l . Then the following problem is solved:

$$\text{minimize}_{q_{e_l}} E \left(\sum_{l \in E} e_l b_l + \sum_{i \in E} \xi_i h_i(\mathbf{e}) \right) - T \sum_{i \in V} S(q_{e_i}) \quad l \in E. \quad (36)$$

Assume the problem of multicasting data at an achievable rate, r_0 , with minimum cost incurred. Based on the earlier discussion, in order to multicast data at rate r_0 , it suffices

to construct a single Steiner tree with link capacities greater than or equal to r_0 . This problem can be formulated as follows:

$$\begin{aligned} & \text{minimize } \sum_{l \in E} g_l(\mathbf{p}), \\ & \text{subject to: } h_i(\mathbf{e}) = 0, \quad \forall i \in V, \\ & \quad r_0 e_l \leq c_l(\mathbf{p}), \quad \forall l \in E, \\ & \quad 0 \leq p_l \leq p_{l, \max}, \quad \forall l \in E, \\ & \quad \sum_{l \in O(n)} p_l \leq \bar{P}_{n, \max}, \quad \forall n \in V. \end{aligned} \quad (37)$$

Using PC, the above problem can be rewritten as follows:

$$\begin{aligned} & \text{minimize}_{q_{e_l}, q_{p_l}} E \left(\sum_{l \in E} g_l(p_l) + \sum_{i \in V} \xi_i h_i(\mathbf{e}) + \sum_{l \in E} \lambda_l r_0 e_l \right. \\ & \quad \left. - \sum_{l \in E} \lambda_l c_l + \sum_{n \in V} \nu_n \left(\sum_{l \in O(n)} p_l - \bar{P}_{n, \max} \right) \right) \\ & \quad - T \sum_{l \in E} S(q_{e_l}) - T \sum_{l \in E} S(q_{p_l}). \end{aligned} \quad (38)$$

The Lagrange multipliers are then updated according to (A.9):

$$\begin{aligned} \xi_i^{(t+1)} &= [\xi_i^{(t)} + \eta E(h_i(\mathbf{e}))]^+ \quad \forall i \in V, \\ \nu_n^{(t+1)} &= \left[\nu_n^{(t)} + \eta E \left(\sum_{l \in O(n)} p_l - \bar{P}_{n, \max} \right) \right]^+ \quad \forall n \in V, \\ \lambda_l^{(t+1)} &= [\lambda_l^{(t)} + \eta E(r_0 e_l - c_l)]^+ \quad \forall l \in E. \end{aligned} \quad (39)$$

The minimization problem in (38) can then be decomposed into the following subproblems in network and physical layers, respectively, as follows:

$$\text{minimize}_{q_{e_l}} E \left(\sum_{i \in V} \xi_i h_i(\mathbf{e}) + \sum_{l \in E} \lambda_l r_0 e_l \right) - T \sum_{l \in E} S(q_{e_l}), \quad \forall l \in E, \quad (40)$$

$$\begin{aligned} & \text{minimize}_{q_{p_l}} E \left(\sum_{l \in E} g_l(p_l) - \sum_{l \in E} \lambda_l c_l + \sum_{n \in V} \nu_n \left(\sum_{l \in O(n)} p_l - \bar{P}_{n, \max} \right) \right) \\ & \quad - T \sum_{l \in E} S(q_{p_l}), \quad \forall l \in E. \end{aligned} \quad (41)$$

Comparing (40) with (36), it can be realized that the network layer problem corresponds to finding minimum-cost multicast subgraph (i.e., Steiner tree) with link costs equal to $\lambda_l r_0$. The network problem can in turn be decomposed into the following single-variable subproblems:

$$\text{minimize}_{q_{e_l}} E(\lambda_l r_0 e_l + \xi_{\text{head}(l)} h_{\text{head}(l)}(\mathbf{e}) + \xi_{\text{tail}(l)} h_{\text{tail}(l)}(\mathbf{e})) - TS(q_{e_l}). \quad (42)$$

It should be noted that link l , corresponding to e_l , is in connection with exactly two nodes, the node whose link exits from it (head (l)) and the node whose link enters it (tail (l)). Therefore, only $h_{\text{tail}(l)}$ and $h_{\text{head}(l)}$ will be functions of e_l and should be considered in (42). The physical layer problem can then be decomposed as follows:

$$\text{minimize}_{q_{p_l}} E \left(g_l(p_l) + p_l v_{\text{head}(l)} - \sum_{l \in E} \lambda_l c_l \right) - TS(q_{p_l}), \quad (43)$$

and q_{e_l} 's and q_{p_l} 's are updated according to (A.7), where G is replaced by G_5 and G_6 as follows:

$$\begin{aligned} G_5 &= \lambda_l r_0 e_l + \xi_{\text{head}(l)} h_{\text{head}(l)}(\mathbf{e}) + \xi_{\text{tail}(l)} h_{\text{tail}(l)}(\mathbf{e}), \\ G_6 &= g_l(p_l) + p_l v_{\text{head}(l)} - \sum_{l \in E} \lambda_l c_l. \end{aligned} \quad (44)$$

The probability distributions associated with indicator variables and link powers can be updated in a distributed fashion, at network and physical layers, respectively. Updating q_{e_l} 's requires computing $E(e_l)$, $E(h_{\text{head}(l)}(\mathbf{e}))$, and $E(h_{\text{tail}(l)}(\mathbf{e}))$. $E(h_{\text{head}(l)}(\mathbf{e}))$ and $E(h_{\text{tail}(l)}(\mathbf{e}))$ can be computed by using probability distribution of indicator variables associated with links connected to nodes head (l) and tail (l), respectively. Each node can update its outgoing links by exchanging links probability distributions with its neighbors. Lagrange multipliers can also be updated at each link l (more precisely at node this link originates from), using q_{e_l} and q_{p_l} . Hence a distributed algorithm can be designed and the proposed approach can be extended to find the maximum net utility function:

$$\begin{aligned} &\text{maximize } U(r) - \sum_{l \in E} g_l(\mathbf{p}), \\ &\text{subject to: } h_i(\mathbf{e}) = 0, \quad \forall i \in V, \\ &\quad r e_l \leq c_l(\mathbf{p}), \quad \forall l \in E, \\ &\quad 0 \leq p_l \leq p_{l,\max}, \quad \forall l \in E, \\ &\quad \sum_{l \in O(n)} p_l \leq \bar{P}_{n,\max}, \quad \forall n \in V. \end{aligned} \quad (45)$$

It can easily be shown that each subproblem at network layer is given by

$$\begin{aligned} &\text{minimize}_{q_r, q_{e_l}} E \left\{ -U(r) + \sum_{l \in E} r \lambda_l e_l + \sum_{i \in V} \xi_i h_i(\mathbf{e}) \right\} \\ &\quad - T \sum_{l \in E} S(q_{e_l}). \end{aligned} \quad (46)$$

It should be noted that the subproblems in physical layer are also of the form given in (41). However, in this case since variable r couples the subproblems, the network layer problem cannot be decomposed in a way similar to (40).

5.2. General networks

In this part, we propose a method that can be applied in an arbitrary (cyclic or acyclic) network, however, at a higher

complexity cost. The $s - d_i$ binary flow with rate r is defined as a length- $|E|$ vector \mathbf{f}_i satisfying the flow constraint:

$$\sum_{l \in O(n)} f_{i,l} - \sum_{l \in I(n)} f_{i,l} = \begin{cases} r, & n = s, \\ 0, & \forall n \in V - \{s, d_i\} \triangleq \mathcal{J}_{s,d_i}(r), \\ -r, & n = d_i, \end{cases} \quad (47)$$

where each component of \mathbf{f}_i , $f_{i,l}$ takes its value from the set $\{0, 1\}$. Note that, by this definition, the $s - d_i$ binary flow with unit value corresponds to a path from source to i th destination. A set of N binary flows from source to destinations constructs a multicast subgraph, since it ensures existence of a path between source and each destination. Therefore, the link l of network graph (G) is included in this multicast graph (i.e., $e_l = 1$) if it is included in at least one path from a source node to a destination (or equivalently: $\bigvee_{i=1}^N f_{i,l} = 1$, where \vee denotes logical or). Consequently, we define e_l as

$$e_l = \bigvee_{i=1}^N f_{i,l}. \quad (48)$$

The optimum graph can be found by exploring all subgraphs constructed in this way. This problem can be formulated as

$$\begin{aligned} &\text{minimize } \sum_{l \in E} e_l b_l, \\ &\text{subject to: } \sum_{l \in O(n)} f_{i,l} - \sum_{l \in I(n)} f_{i,l} = \mathcal{J}_{s,d_i}(1), \quad i = 1, 2, \dots, N, \\ &\quad h_j(\mathbf{e}) = 0, \quad j \in V, \end{aligned} \quad (49)$$

where $h_j(\mathbf{e})$ is defined as before. A probability distribution is associated with each variable $f_{i,l}$ (rather than e_l) and $q_{f_{i,l}}$. Then by solving the following problem:

$$\begin{aligned} &\text{minimize}_{q_{f_{i,l}}} E \left\{ \sum_{l \in E} e_l b_l + \sum_{i=1}^N \sum_{n \in V} \mu_{i,n} \left(\sum_{l \in O(n)} f_{i,l} - \sum_{l \in I(n)} f_{i,l} - \mathcal{J}_{s,d_i}(1) \right) \right. \\ &\quad \left. + \sum_{i=1}^N \lambda_i c_i(\mathbf{e}) \right\} - T \sum_{i=1}^N \sum_{l \in E} S(q_{f_{i,l}}), \quad i = 1, \dots, N, l \in E, \end{aligned} \quad (50)$$

and based on discussion presented in Section 5.1, the minimum cost multicast problem at rate r_0 can be formulated as

$$\begin{aligned} &\text{minimize } \sum_{l \in E} g_l(\mathbf{p}), \\ &\text{subject to: } \sum_{l \in O(n)} f_{i,l} - \sum_{l \in I(n)} f_{i,l} = \mathcal{J}_{s,d_i}(1), \\ &\quad h_i(\mathbf{e}) = 0, \quad \forall i \in V, \\ &\quad r_0 e_l \leq c_l(\mathbf{p}), \quad \forall l \in E, \\ &\quad 0 \leq p_l \leq p_{l,\max}, \quad \forall l \in E, \\ &\quad \sum_{l \in O(n)} p_l \leq \bar{P}_{n,\max}, \quad \forall n \in V. \end{aligned} \quad (51)$$

The associated PC based problem is then given by

$$\begin{aligned}
 \text{minimize } E \left\{ \sum_{l \in E} g_l(p_l) + \sum_{i \in V} \xi_i h_i(\mathbf{e}) + \sum_{l \in E} \lambda_l r_0 e_l \right. \\
 - \sum_{l \in E} \lambda_l c_l + \sum_{n \in V} \nu_n \left(\sum_{l \in O(n)} p_l - \bar{P}_{n,\max} \right) \\
 \left. + \sum_{i=1}^N \sum_{n \in V} \mu_{i,n} \left(\sum_{l \in O(n)} f_{i,l} - \sum_{l \in I(n)} f_{i,l} - \mathfrak{J}_{s,d_i}(1) \right) \right\} \\
 - T \sum_{l \in E} S(q_{e_l}) - T \sum_{l \in E} S(q_{p_l}). \quad (52)
 \end{aligned}$$

Lagrange multipliers $\mu_{i,n}$ can be updated as follows:

$$\mu_{i,n}^{(t+1)} = \left[\mu_{i,n}^{(t)} + \eta E \left(\sum_{l \in O(n)} f_{i,l} - \sum_{l \in I(n)} f_{i,l} \right) \right]^+, \quad (53)$$

where Lagrange multipliers ξ_i , λ_i , ν_n can be updated as before.

Finally, the above problem can be decomposed into subproblems (54) and (40):

$$\begin{aligned}
 \text{minimize } E \{ f_{i,l} (\mu_{i,\text{head}(l)} - \mu_{i,\text{tail}(l)}) + r_0 \lambda_l e_l \\
 + \xi_{\text{head}(l)} h_{\text{head}(l)}(\mathbf{e}) + \xi_{\text{tail}(l)} h_{\text{tail}(l)}(\mathbf{e}) \} \\
 - TS(q_{f_{i,l}}), \quad (54)
 \end{aligned}$$

where $q_{f_{i,l}}$'s can be updated based on the above equations and q_{p_l} 's are updated as before. In this way, a distributed algorithm can be designed in a similar way as the algorithm presented earlier.

Based on the above discussions, the difference between the proposed method and the distributed algorithm presented in [14] becomes more evident. In fact, in [14] a link is assumed to be either enabled, if the received power exceeds a threshold value and data can be transferred via this link at a desired rate, or disabled, if the received power does not reach the threshold level. However, our approach takes both link capacity and interference into account, reflecting a more realistic cooperation between different nodes (in physical and network layers) in order to achieve the desired throughput.

6. SIMULATION RESULTS

Consider the network represented in Figure 1, as an example. The source node (S) multicasts data to receivers d_1 and d_2 via the network. The goal is to achieve optimal throughput in the range $[0, 2]$. The utility function is assumed to be equal to r^2 (which is a not a concave but monotonic function of r). We define net-utility as $r^2 - \sum_{l \in E} 0.001 * p_l$ (therefore, the main emphasis will be on achieving maximum rate rather than minimizing total consumed power). Each link is assumed to select its transmit power from a discrete set of values $\{0, 1, 2, \dots, 5\}$ and each node has a power budget

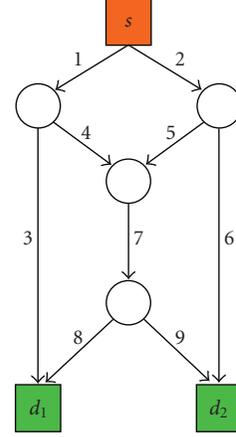


FIGURE 1: Network topology.

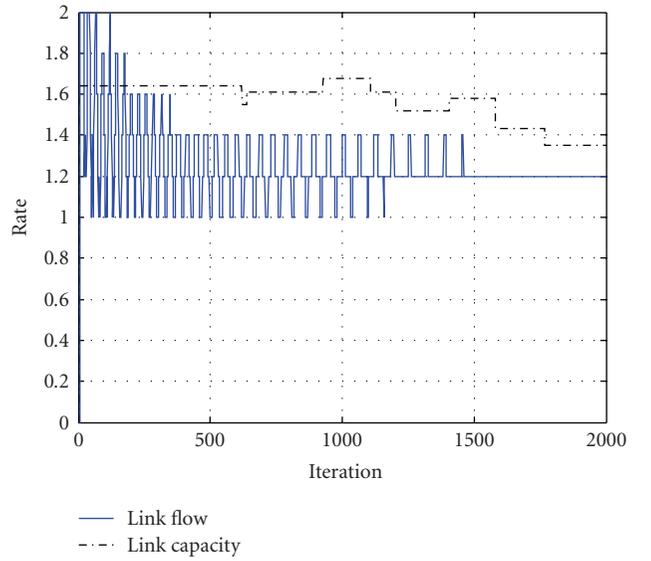
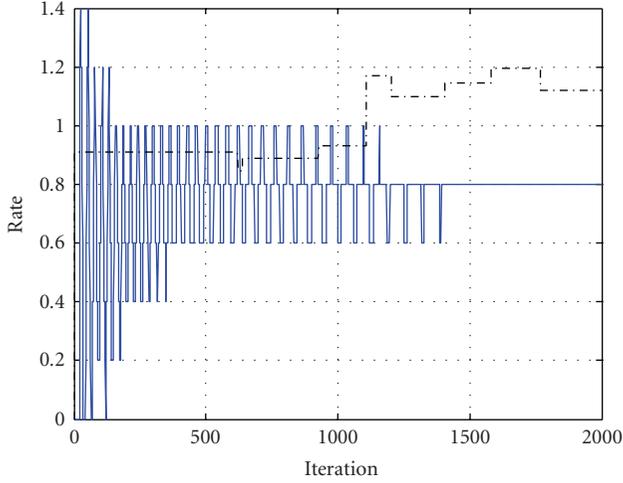


FIGURE 2: The flow and capacity of links 1 and 2.

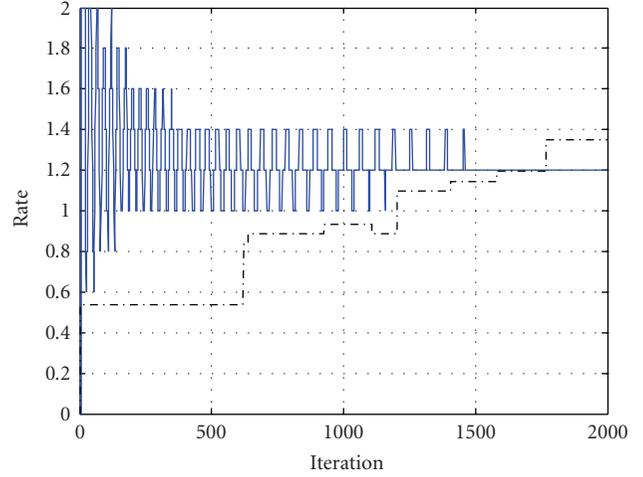
equal to 10. f_l and $e_{i,l}$ are also assumed to take values from the set $\{0, .2, .4, \dots, 2\}$. Link gains, interference gains, and noise variances are assumed to be equal to 1, 0.05 and 0.1, respectively. Also, we assume that achievable rate of each link is given by $c_l(p) = \log(1 + \text{SINR}_l)$. Figures 2, 3, 4, 5 show the flow and capacity associated with each link in this

scenario. Due to the symmetric structure of the network, some links have the same link flow and capacity. After 2000 iterations, the optimal multicast throughput is achieved. Figure 6 shows conceptual flows and flows ($e_{1,l}, e_{2,l}, f_l$) of the links, where $e_{1,l}$ and $e_{2,l}$ satisfy flow conservation and link capacity constraints and also the multicast rate of 2 is achieved. This multicast rate is feasible since all the link flows are supported by the physical layer (i.e., each link has a capacity greater than its flow). Link capacities and link power vectors are consequently given by $[1.35, 1.35, 1.12, 0.84, 0.84, 1.12, 1.35, 1.35, 1.35]$ and $[4, 4, 3, 2, 2, 3, 4, 4, 4]$, respectively.



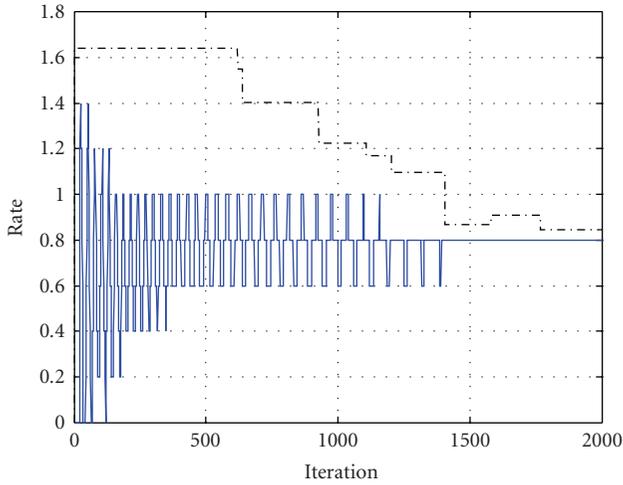
— Link flow
- - - Link capacity

FIGURE 3: The flow and capacity of links 3 and 6.



— Link flow
- - - Link capacity

FIGURE 5: The flow and capacity of links 7, 8, and 9.



— Link flow
- - - Link capacity

FIGURE 4: The flow and capacity of links 4 and 5.

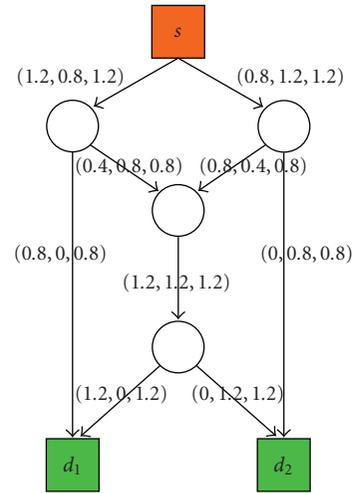


FIGURE 6: $(e_{1,i}, e_{2,i}, f_i)$ of each link at iteration 2000.

It should be noted that if the utility function is assumed to be given by $\log(1 + r)$ rather than r^2 , we can apply the method proposed in [13] (i.e., underestimate link capacity by $\log(\text{SINR}_l)$) and use logarithmic transformation. However, the maximum multicast throughput in this case would only reach the value of 1.54. This is due to the fact that as a result of the relatively high interference between links, such underestimation will not lead to the optimal solution.

As an example of minimum cost multicast, consider the case of multicasting data based on single Steiner tree at rate $r_0 = 1.9$ with minimum total link power. It can be verified that the optimal solution is achieved when link power vector is equal to $[4, 4, 4, 0, 0, 4, 0, 0, 0]$ and link capacity vector is given by $[1.9, 1.9, 1.9, 0, 0, 1.9, 0, 0, 0]$.

Figure 7 shows the optimum Steiner tree which can be shown by indicator vector $\mathbf{e} = [1, 1, 1, 0, 0, 1, 0, 0, 0]$. As shown in Figures 8 and 9, the proposed method converges to the optimum value where each link flow is equal to $r_0 e_l$. These figures show that such multicast rate is feasible since all links flow are supported by the physical layer. Also, Figures 10 and 11 show that optimum link powers are also achieved. Considering the problem of minimum power multicast at rate 2, simulation results show that rate 2 can be achieved by a total power of 14. The associated parameter values are then given by $\mathbf{p} = [2, 2, 1, 1, 1, 1, 2, 2, 2]$, $\mathbf{f} = [1.2, 1.2, 0.8, 0.8, 0.8, 0.8, 1.2, 1.2, 1.2]$, $\mathbf{e}_1 = [1.2, 0.8, 0.8, 0.4, 0.8, 0, 1.2, 1.2, 0]$, $\mathbf{e}_2 = [0.8, 1.2, 0, 0.8, 0.4, 0.8, 0, 1.2, 1.2]$, and $\mathbf{c} = [1.35, 1.35, 0.84, 0.84, 0.84, 0.84, 1.35, 1.35, 1.35]$.

Therefore, as expected, by using network coding, we can multicast at a higher rate and with less consumed power in comparison with single Steiner tree routing. Finally, it is

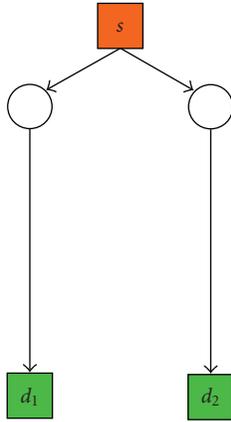


FIGURE 7: Steiner tree corresponding to indicator vector $\mathbf{e} = [1, 1, 1, 0, 0, 1, 0, 0, 0]$.

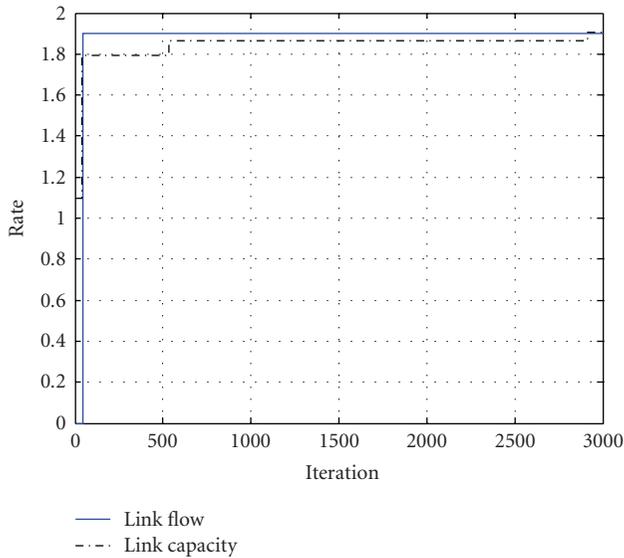


FIGURE 8: The flow and capacity of links 1,2,3, and 6.

important to note that while we have considered the network presented in Figure 1 in our simulations, as presented in earlier sections, our proposed algorithms are quite general and can be applied to any network with arbitrary topology. Since in problem formulation the network is assumed to be quite arbitrary, our PC-based algorithm converges to at least a suboptimum solution, corresponding to a local minimum cost.

7. CONCLUSIONS

In this paper, the problem of finding an optimal multicast solution in multihop wireless networks with interference has been addressed. Using the PC method, the problem has been turned into a convex optimization problem over probability distributions. Consequently, it was shown that the new problem can be decomposed into two subproblems at network and physical layers and a corresponding

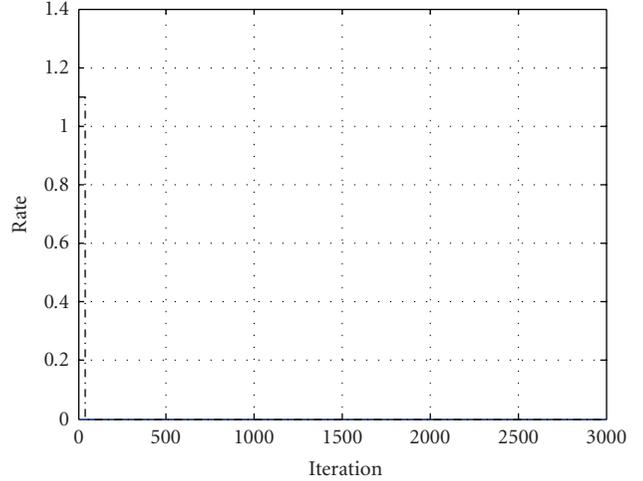


FIGURE 9: The flow and capacity of links 4, 5, 7, 8, and 9.

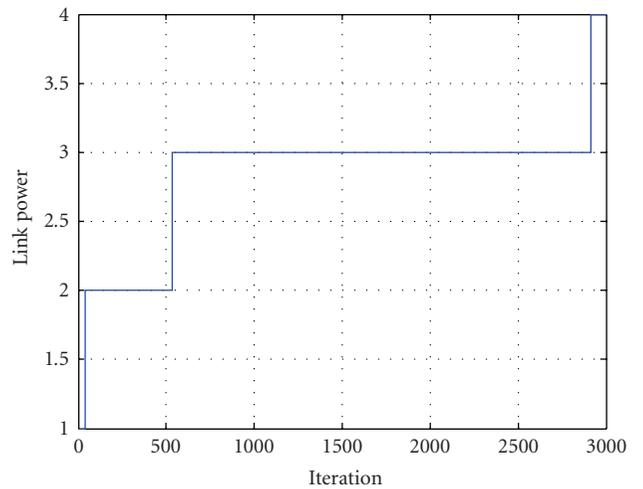


FIGURE 10: Power of links 1, 2, 3, and 6.

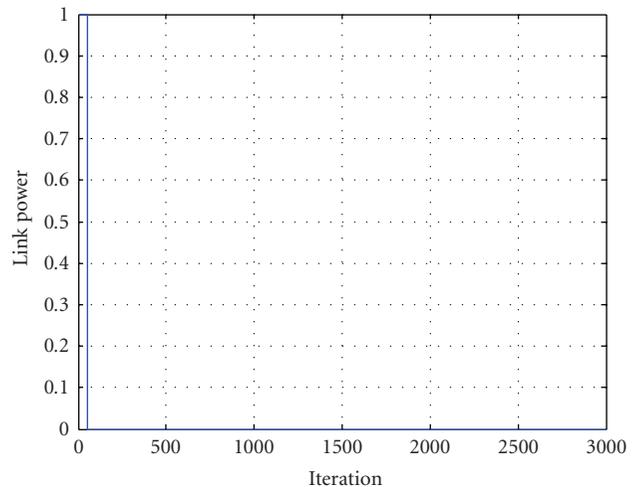


FIGURE 11: Power of links 4, 5, 7, 8, and 9.

cross-layer distributed approach to solve the problem has been proposed. Also, distributed cross-layer algorithms for multicommodity flow-based routing and single Steiner tree routing have been proposed.

As expected, the network coding-based solution performs better than the solution based on routing.

APPENDIX

PROBABILITY COLLECTIVES (PCS) OPTIMIZATION SCHEME

Consider the optimization problem, $\min_x G(x)$, where each component of x , x_i is a discrete scalar variable assumed to take a finite number of possible values $|X_i|$. A continuous variable with a finite domain can also be approximated with a variable with a finite number of values. In PC method, each variable is considered as an *agent* [24–26] where at each step, each agent updates its probability distribution according to the *maximum entropy (Maxent) principle* and independently from other agents [27]. Based on this approach, given the prior knowledge of the utility function, the goal is to find the probability distribution that is consistent with the a priori knowledge and also attains the maximum entropy. We will denote the entropy by

$$S(q_i) = - \sum_{x_i \in X_i} q_i(x_i) \ln q_i(x_i), \quad (\text{A.1})$$

where q_i denotes the probability distribution of agent i and $q_i(x_i)$ is the probability that agent i takes the value x_i . Since agents are assumed to be independent, the joint distribution q of the agents will be of the form:

$$q(x) = \prod_i q_i(x_i), \quad (\text{A.2})$$

and consequently,

$$S(q) = \sum_i S(q_i). \quad (\text{A.3})$$

Using the Maxent principle, the original optimization problem can then be converted into the following optimization problem over the probability distribution [25]:

$$\begin{aligned} & \text{minimize } l(q, T) = E(G) - TS(q), \\ & \text{subject to: } \sum_{x_i} q_i(x_i) = 1, \quad q_i(x_i) \geq 0, \quad \forall x_i, \end{aligned} \quad (\text{A.4})$$

where $l(q, T)$ is the *Maxent Lagrangian*, T is a positive Lagrange multiplier, and

$$E(G) = \sum_x G(x) \prod_i q_i(x_i). \quad (\text{A.5})$$

Each agent is aware of the previous probability distribution of other nodes and updates its probability distribution as the solution of following convex optimization problem:

$$\begin{aligned} q_i^* &= \arg\{\text{minimize}_{q_i} E[G] - TS(q)\} \\ &= \arg\left\{ \text{minimize}_{q_i} \sum_{x_i} q_i(x_i) E[G | x_i] \right. \\ &\quad \left. - T \sum_{x_i} q_i(x_i) \ln [q_i(x_i)] \right\}, \end{aligned} \quad (\text{A.6})$$

$$\text{subject to: } \sum_{x_i} q_i(x_i) = 1, \quad q_i(x_i) \geq 0, \quad \forall x_i,$$

where $E[G | x_i] = \sum G(x) \prod_{j \neq i} q_j(x_j)$. Using *Newton Updating*, the following updating rule can be obtained [24]:

$$\begin{aligned} q_i^{(t+1)}(j) &= q_i^{(t)}(j) - \alpha q_i^{(t)}(j) \\ &\quad \times \left\{ \frac{E(G | x_i = j) - E(G)}{T} + S(q_i^{(t)}) + \ln [q_i^{(t)}] \right\}, \\ &\quad \forall j \in X_i, \end{aligned} \quad (\text{A.7})$$

where α and t , respectively, denote the step size and the iteration number. As the parameter T is gradually decreased, and in the limit when $T \rightarrow 0$, the set of probability distributions that simultaneously minimize Maxent Lagrangian will become the same as the set of delta functions at the local minima of the objective function G [24, 25]. Updating can be performed in parallel for different agents, since calculating expectations in (A.7) requires previous (step) probability distributions. Therefore, by exchanging previous probability distributions, each agent can update its probability distributions, simultaneously.

Other constraints can also be included by augmenting the objective function with Lagrange multipliers λ_i and the constraint functions $C_i(x)$:

$$G(x) \rightarrow G(x) + \sum_i \lambda_i C_i(x), \quad (\text{A.8})$$

where $c_i(x)$ is a nonequality constraint of the form: $C_i(x) \leq 0$ [28]. The updating rule for Lagrange multipliers is obtained by taking the derivative of the augmented Lagrangian with respect to each Lagrange multiplier:

$$\lambda_i^{(t+1)} = \left[\lambda_i^{(t)} + \eta E[C_i(x)] \right]^+ \quad (\text{A.9})$$

where $[\cdot]^+ = \max\{\cdot, 0\}$ and η is a positive step size. Consequently, the minimizer of (A.4) when the function G is augmented with Lagrange multipliers λ_i corresponds to at least a local minimum of the original objective function G subject to the same constraints [26, 29].

ACKNOWLEDGMENTS

This work has been partially supported by Iran National Science Foundation (INSF) under Grant no. 84094/14 and also supported in part by the Ministerio de Educacion y Ciencia of Spain under Grant no. TEC2007-68020-C04-03.

REFERENCES

- [1] R. Ahlswede, N. Cai, S.-Y. R. Li, and R. W. Yeung, "Network information flow," *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1204–1216, 2000.
- [2] R. Koetter and M. Médard, "An algebraic approach to network coding," *IEEE/ACM Transactions on Networking*, vol. 11, no. 5, pp. 782–795, 2003.
- [3] S.-Y. R. Li, R. W. Yeung, and N. Cai, "Linear network coding," *IEEE Transactions on Information Theory*, vol. 49, no. 2, pp. 371–381, 2003.
- [4] S. Jaggi, P. Sanders, P. A. Chou, et al., "Polynomial time algorithms for multicast network code construction," *IEEE Transactions on Information Theory*, vol. 51, no. 6, pp. 1973–1982, 2005.
- [5] T. Ho, M. Medard, R. Koetter, et al., "A random linear network coding approach to multicast," *IEEE Transactions on Information Theory*, vol. 52, no. 10, pp. 4413–4430, 2006.
- [6] Z. Li, B. Li, and L. C. Lau, "On achieving maximum multicast throughput in undirected networks," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2467–2485, 2006.
- [7] K. Jain, M. Mahdian, and M. R. Salavatipour, "Packing Steiner trees," in *Proceedings of the 14th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA '03)*, pp. 266–274, Baltimore, Md, USA, January 2003.
- [8] D. S. Lun, N. Ratnakar, M. Medard, et al., "Minimum-cost multicast over coded packet networks," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2608–2623, 2006.
- [9] Y. Wu and S.-Y. Kung, "Distributed utility maximization for network coding based multicasting: a shortest path approach," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 8, pp. 1475–1488, 2006.
- [10] S. Bhadra, S. Shakkottai, and P. Gupta, "Min-cost selfish multicast with network coding," *IEEE Transactions on Information Theory*, vol. 52, no. 11, pp. 5077–5087, 2006.
- [11] J. E. Wieselthier, G. D. Nguyen, and A. Ephremides, "Energy-efficient broadcast and multicast trees in wireless networks," *Mobile Networks and Applications*, vol. 7, no. 6, pp. 481–492, 2002.
- [12] L. Xiao, M. Johansson, and S. P. Boyd, "Simultaneous routing and resource allocation via dual decomposition," *IEEE Transactions on Communications*, vol. 52, no. 7, pp. 1136–1144, 2004.
- [13] M. Johansson, L. Xiao, and S. Boyd, "Simultaneous routing and resource allocation in CDMA wireless data networks," in *Proceedings of the IEEE International Conference on Communications (ICC '03)*, vol. 1, pp. 51–55, Anchorage, Alaska, USA, May 2003.
- [14] M. X. Cheng, J. Sun, M. Min, Y. Li, and W. Wu, "Energy-efficient broadcast and multicast routing in multihop ad hoc wireless networks," *Wireless Communications and Mobile Computing*, vol. 6, no. 2, pp. 213–223, 2006.
- [15] Y. Wu, P. A. Chou, and S.-Y. Kung, "Minimum-energy multicast in mobile ad hoc networks using network coding," *IEEE Transactions on Communications*, vol. 53, no. 11, pp. 1906–1918, 2005.
- [16] Y. Wu, P. A. Chou, Q. Zhang, K. Jain, W. Zhu, and S.-Y. Kung, "Network planning in wireless ad hoc networks: a cross-layer approach," *IEEE Journal on Selected Areas in Communications*, vol. 23, no. 1, pp. 136–150, 2005.
- [17] J. Yuan, Z. Li, W. Yu, and B. Li, "A cross-layer optimization framework for multicast in multi-hop wireless networks," in *Proceedings of the 1st International Conference on Wireless Internet (WICON '05)*, pp. 47–54, Budapest, Hungary, July 2005.
- [18] Y. Xi and E. M. Yeh, "Distributed algorithms for minimum cost multicast with network coding in wireless networks," in *Proceedings of the 4th International Symposium on Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks (WiOpt '06)*, pp. 1–9, Boston, Mass, USA, April 2006.
- [19] E. K. P. Chong and S. H. Zak, *An Introduction to Optimization*, Wiley-Interscience, New York, NY, USA, 2001.
- [20] B. W. Wah and Y. X. Chen, "Constrained genetic algorithms and their applications in nonlinear constrained optimization," in *Proceedings of the 12th IEEE International Conference on Tools with Artificial Intelligence (ICTAI '00)*, pp. 286–293, Vancouver, Canada, November 2000.
- [21] A. I. de Freitas Vaz and E. M. D. G. Pinto Fernandes, "Optimization of nonlinear constrained particle swarm," *Technological and Economic Development of Economy*, vol. 12, no. 1, pp. 30–36, 2006.
- [22] Y. Wu, *Network coding for multicasting*, Ph.D. dissertation, Princeton University, Princeton, NJ, USA, 2006.
- [23] A. M. Mahdy and J. S. Deogun, "Multicast optimization in broadband adaptive optical wireless networks," in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '05)*, vol. 3, pp. 1780–1785, New Orleans, La, USA, March 2005.
- [24] S. R. Bieniawski, I. M. Kroo, and D. H. Wolpert, "Discrete, continuous, and constrained optimization using collectives," in *Proceedings of the 10th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, pp. 3079–3087, Albany, NY, USA, August–September 2004.
- [25] S. R. Bieniawski, *Distributed optimization and flight control using collectives*, Ph.D. dissertation, Stanford University, Stanford, Calif, USA, 2005.
- [26] D. H. Wolpert and S. Bieniawski, "Distributed control by Lagrangian steepest descent," in *Proceedings of the 43rd IEEE Conference on Decision and Control (CDC '04)*, vol. 2, pp. 1562–1567, Nassau, Bahamas, December 2004.
- [27] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, Wiley-Interscience, New York, NY, USA, 1991.
- [28] W. Macready and D. H. Wolpert, "Distributed constrained optimization," in *Proceedings of the 5th International Conference on Complex Systems (ICCS '04)*, Y. Bar-Yam, Ed., Boston, Mass, USA, May 2004.
- [29] D. H. Wolpert, "Factoring a canonical ensemble," cond-mat/0307630, 2003.