

Optimal sub-graph selection over coded networks with delay and limited-size buffering

H. Ghasvari¹ M.A. Raayatpanah² B.H. Khalaj³ H. Bakhshi⁴

¹Department of Electrical Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran

²School of Mathematics, Statistics and Computer Science, University of Tehran, Tehran, Iran

³Department of Electrical Engineering and Advanced Communication Research Institute, Sharif University of Technology, Tehran, Iran

⁴Department of Electrical Engineering, Shahed University, Tehran, Iran

E-mail: ghasvari49@yahoo.com

Abstract: The authors consider the problem of finding a minimum cost multicast sub-graph based on network coding, where delay values associated with each link, limited buffer-size of the intermediate nodes and link capacity variations over time are taken into account. The authors consider static multicast (where membership of the multicast group remains constant for the duration of the connection) in lossless wireline networks. For such networks, first the continuous-time (asynchronous packet transmissions) model is formulated. Subsequently, the discrete-time model for the synchronous packet transmissions scenario is derived. Then, by using an auxiliary time-expanded network, a decentralised algorithm in the presence of link delays, limited buffer-size and time-varying link capacity is proposed. The proposed algorithm when coupled with decentralised code construction schemes results in a fully decentralised approach for achieving minimum-cost in such multicast networks. Also, how adding buffering capability at intermediate nodes reduces the overall cost of the optimal sub-graph is discussed in this study. In addition, as will be shown, inclusion of buffering capacity at intermediate nodes makes it possible to find minimum multicast solution in scenarios that such solutions do not exist otherwise.

1 Introduction

Network coding generalises the traditional routing paradigm, in which relaying nodes can only forward or replicate data, by allowing them to perform arbitrary operations on information received at each node. It is well known that network performance can be significantly improved through such network coding approach [1, 2]. As an important example, use of network coding makes the once intractable optimal multicast routing problem tractable [1, 3].

As shown in [1], network coding makes it possible to reach the achievable throughput of a multicast session by running the max-flow algorithm from the source to each individual receiver, and choosing its minimal value. Li *et al.* [4] further showed that the above result can be obtained by running a linear coding algorithm. In addition, Jaggi *et al.* [5] also showed that such network codes could be designed using a polynomial time algorithm.

For cyclic graphs, the multicast capacity can be achieved by performing linear time-invariant coding over a sufficiently large field, as shown by Koetter and Medard [2]. An alternative theoretical treatment of cyclic graphs is via linear time-varying network coding over a discrete-time trellis, used for example in [1, 4]. Specifically, the original graph is expanded into a larger acyclic trellis, whose edges correspond to transmissions along the original edge at different time instances. Since the trellis is acyclic,

instantaneous linear network coding for acyclic graphs are applicable in this scenario.

Chou *et al.* [6] were among the first to propose a practical network coding solution where each node maintains a buffer. Whenever a node receives a packet from one of its incoming links, it stores the packet in its buffer and as a transmission opportunity arises, the node generates an output packet by linearly mixing the packets in the buffer using random coefficients from a finite-field F .

Wu [7] analysed the asymptotic performance of typical network coding schemes. Extending the discrete-time trellis used in previous theoretical studies, he introduced a continuous-time trellis, which models the asynchronous packet transmissions in a practical network. The practical network coding scheme can then be interpreted as a random linear coding scheme implemented over the continuous-time trellis.

It should be noted that the minimum-cost multicast scheme in networks using a network coding approach can be decomposed into two parts: (i) finding the optimal sub-graph to code over and (ii) determining the code to use over the sub-graph [8]. In general, the first issue has been addressed in the literature to a less extent relative to the second one. For example, Lun *et al.* [3] proposed a decentralised network resource allocation scheme to find minimum-cost sub-graphs that allow multicast connections to be established (with appropriate coding) over coded packet networks. The decentralised methods in [3] generate the sub-graph which

the network codes designed in [4, 9] apply over the sub-graph. Cui and Ho [10] considered finding a minimum cost multicast sub-graph with network coding on directed graphs, where the packet transmission rate on each link is constrained to integer values and a greedy algorithm in combination with a LP rounding algorithm has been proposed to solve the problem. A more direct primal approach is taken in [11], where a distributed sub-gradient algorithm based on finding the critical cut is proposed. Also, Xi and Yeh [12] provided an analytical framework as well as a set of distributed solutions for optimising the configuration of network coding in both wireline and wireless networks.

However, one drawback of algorithms that address the problem of finding the optimal sub-graph is their assumption that packets going through each link do not incur any delays. In addition, the effect of limited buffer size in intermediate nodes is not taken into account in such approaches. Such simplistic assumptions result in a static sub-graph in which time is not represented explicitly and passage of packets through links seems to occur instantaneously. However, in practice, packets are delayed while transmitted along each link, and the output packets do not leave a node at the same time that packets are arriving at the same node. Needless to say, such assumptions affect the overall performance analysis of a given network. Another realistic characteristic of realistic networks is their variation over time. In fact, important characteristics of real-world networks such as link costs and capacities are often subject to fluctuations over time.

In this paper, we address the problem of finding a minimum cost multicast sub-graph with network coding on directed graphs, where the delay and buffering effects of packet transmission and link capacity variations over time are taken into account. The rest of this paper is organised as follows. In Section 2, the system model for the delayed network is presented. Section 3 extends the problem to the varying linear cost model over time. Simulation results are presented in Section 4. Finally, Section 5 concludes the paper.

2 Networks with delay and limited-size buffering

2.1 Network coding and minimum cost sub-graph generation (traditional model: links without delay)

We first overview the model given in [3], in which no link delay or buffer capacity at intermediate nodes is assumed. The communication network is represented by a directed graph $G = (V, A)$, where V is the set of nodes, $|V| = n$, and A is the set of links, $|A| = m$, in G . Each link $e = (i, j)$ represents a lossless point-to-point link from node i to node j . For the link $e = (i, j)$, we write $\text{head}(e) = j$ and $\text{tail}(e) = i$. For a node $i \in V$, the terms δ_i^+ and δ_i^- denote the set of links leaving node i ($\text{tail}(e) = i$) and entering node i ($\text{head}(e) = i$), respectively. Let Z_e denotes the rate at which coded packets are injected on link e . The linear, separable cost c_e denotes the cost per unit rate of sending coded packets over link $e \in A$. The capacity u_e of link e is defined to be the number of packets that can be sent over link e in one time unit. We assume that link capacities are non-negative integer numbers. Single-session multicast is considered in this paper, where a source node $s \in V$ must transmit an integer number of R packets per unit time to every terminal in a set of K terminals $K \subset V$. The minimum cost sub-graph with network coding is then given by the

following optimisation problem [3]

$$\sum_{e \in A} c_e z_e \quad (1)$$

s.t.

$$z_e \geq x_e^{(k)} \geq 0, \quad \forall e \in A, k \in K \quad (1a)$$

$$\sum_{e \in \delta_i^+} x_e^k - \sum_{e \in \delta_i^-} x_e^k = \sigma_i^k, \quad \forall i \in N, k \in K \quad (1b)$$

$$u_e \geq z_e, \quad \forall e \in A \quad (1c)$$

where

$$\sigma_i^{(k)} = \begin{cases} R, & i = s \\ -R, & i = k \\ 0, & 0w \end{cases}$$

and $x_e^{(k)}$ is the flow towards destination k that is passing through link e .

Decentralised algorithms for solving (1) have been proposed in [3]. The network codes designed in [4, 9] can then be applied to code over the resulting sub-graph.

2.2 Network coding over links with delays: continuous-time formulation

In our discussion, we first consider a more general scenario where information flows may vary continuously over time. Such cases, generally occur when asynchronous packet transmissions are considered in a practical network. The communication network with delay is represented by a directed graph $G = (V, A, D)$, where V is the set of nodes, $|V| = n$, A is the set of links, $|A| = m$ and D is the integer time horizon over which network behaviour at the destination is observed. The time horizon D is chosen such that all source flow units have arrived at the destination by the end of this period.

Let $z_e(p)$ denotes the rate at which coded packets are injected into link e at time instant p . The rate vector z , consisting of $z_e(p)$, $e \in A$ and $p \in [0, D]$, is called a sub-graph. Each link $e \in A$ is associated with three integer parameters: transit time d_e , transit cost c_e and a positive integer capacity at time p denoted by $u_e(p)$.

As d_e denotes the transit time over link e , if $x_e^{(k)}(p)$ is the rate of flow towards the destination k entering link e at time p , the flow $x_e^{(k)}(p)$ arrives at $\text{tail}(e)$ at time $p + d_e$. Throughout the paper, we assume that transit time of packets over each link is an integer value. The integer restriction on the delay does impose some loss of generality because the integer delay solution might not be as accurate as a solution based on continuous delay values. However, we can obtain an integer solution as close as desired to an optimal continuous solution by scaling the time axis properly (i.e. by multiplying all time indexes such as d_e by an integer value M to obtain a higher accuracy value of Md_e , for a proper choice of M , for each arc e).

The linear, separable cost c_e denotes the cost of sending unit rate coded packets $z_e(p)$ over link $e \in A$. Therefore the total weight of multicast connection is given by $\sum_{e \in A} c_e \int_0^D z_e(\xi) d\xi$.

Moreover, at each time instant $p \in [0, D]$, we should have $z_e(p) = \max_{k \in K} \{x_e^{(k)}(p)\}$. Consequently, we obtain

$$z_e(p) \geq x_e^{(k)}(p), \quad \forall e \in A, k \in K, p \in [0, D]$$

Also, as the time horizon D is chosen such that all links do not carry any traffic for $p > D$, $z_e(p) = 0$ at such time instants.

The capacity $u_e(p)$ is an upper bound on the rate of coded packets entering link e at time p

$$u_e(p) \geq z_e(p)$$

It should be noted that the conservation constraints for the case of links with delay will be included by integration of the flow conservation constraints (1b) over time. As storage of flow at intermediate nodes is allowed in our model, the flow entering a node can be stored at that node for a given time before it is sent out. Assuming b_i denotes the buffer length for $i \in V \setminus KU\{s\}$, $k \in K$, $p \in [0, D]$, buffering constraints at each node can be written as follows

$$\int_0^p \left(\sum_{e \in \delta_i^+} x_e^{(k)}(\xi) - \sum_{e \in \delta_i^-} x_e^{(k)}(\xi - d_e) \right) d\xi \leq \sigma_i^{(k)} \quad (2)$$

Consequently, for $p = D$ we have the equality

$$\int_0^D \left(\sum_{e \in \delta_i^+} x_e^{(k)}(\xi) - \sum_{e \in \delta_i^-} x_e^{(k)}(\xi - d_e) \right) d\xi = \sigma_i^{(k)}, \quad \forall i \in N, k \in K$$

where

$$\sigma_i^{(k)} = \begin{cases} R, & i = s \\ -R, & i = k \\ b_i, & i \in V \setminus K \cup \{s\} \end{cases}$$

Therefore the following optimisation problem for optimal sub-graph selection in network coding over networks with delay can be formulated

$$\min \sum_{e \in A} c_e \int_0^D z_e(\xi) d\xi \quad (3)$$

s.t.

$$z_e(p) \geq x_e^{(k)}(p) \geq 0, \quad \forall e \in A, k \in K, \forall p \in [0, D] \quad (3a)$$

$$\int_0^p \left(\sum_{e \in \delta_i^+} x_e^{(k)}(\xi) - \sum_{e \in \delta_i^-} x_e^{(k)}(\xi - d_e) \right) d\xi \leq \sigma_i^{(k)}, \quad \forall i \in N, k \in K, \forall p \in [0, D] \quad (3b)$$

$$\int_0^D \left(\sum_{e \in \delta_i^+} x_e^{(k)}(\xi) - \sum_{e \in \delta_i^-} x_e^{(k)}(\xi - d_e) \right) d\xi = \sigma_i^{(k)}, \quad \forall i \in N, k \in K \quad (3c)$$

$$z_e(p) \leq u_e(p), \quad \forall e \in A, \forall p \in [0, D] \quad (3d)$$

$$z_e(p) = 0, \quad \forall e \in A, \forall p > D \quad (3e)$$

It can be verified that the aforementioned model is similar to the dynamic multicommodity flow problem in [13]. Subsequently, we will present a corresponding discrete-time network model G for the problem.

2.3 Network coding over links with delays: discrete-time formulation

In the continuous-time model, p can take any value in $[0, D]$. However, in discrete-time models, network is observed only at time instances $p = 0, 1, 2, \dots, D$. By considering the same network parameters as before, our model in discrete time can be formulated as follows

$$\min \sum_{e \in A} \sum_{p=0}^{D-1} c_e z_e(p) \quad (4)$$

s.t.

$$0 \leq x_e^{(k)}(p) \leq z_e(p), \quad \forall e \in A, k \in K, \forall p \in \{0, 1, \dots, D-1\} \quad (4a)$$

$$\sum_{e \in \delta_i^+} \sum_{p=0}^{\vartheta} x_e^{(k)}(p) - \sum_{e \in \delta_i^-} \sum_{p=0}^{\vartheta} x_e^{(k)}(p - d_e) \leq \sigma_i^{(k)}, \quad \forall i \in N, k \in K, \forall \vartheta \in \{0, 1, \dots, D-1\} \quad (4b)$$

$$\sum_{e \in \delta_i^+} \sum_{p=0}^D x_e^{(k)}(p) - \sum_{e \in \delta_i^-} \sum_{p=0}^D x_e^{(k)}(p - d_e) = \sigma_i^{(k)}, \quad \forall i \in N, k \in K \quad (4c)$$

$$\bar{z}_e(p) \leq u_e(p), \quad \forall e \in A, \forall p \in \{0, 1, \dots, D-1\} \quad (4d)$$

$$z_e(p) = 0, \quad \forall p > D-1, \quad \forall e \in A \quad (4e)$$

where for the discrete-time version of G , the rate of flow sent into link e during the interval $[p, p+1]$, denoted by $x_e^{(k)}(p)$, is equal to $x_e^{(k)}(p) = \int_p^{p+1} x_e^{(k)}(\xi) d\xi$ for each $p \in \{0, 1, 2, \dots, D\}$. In addition, the discrete-time capacity $u_e(p)$ and coded packet $z_e(p)$ for every time interval $[p, p+1]$ are defined as follows:

$$u_e(p) = \int_p^{p+1} u_e(\xi) d\xi$$

$$z_e(p) = \int_p^{p+1} z_e(\xi) d\xi$$

In order to simplify the problem, the usual approach for deriving practical algorithms for a continuous-time coded network problem is to reduce it to a discrete time one. As mentioned earlier, the approximation error can be reduced by choosing a smaller discrete time step at the cost of additional complexity.

The next step is to verify that the flows $x_e^{(k)}(p)$ and $z_e(p)$ satisfy the constraints in model (4) for the link capacities $u_e(p)$ as shown by the following lemma.

Lemma 1: By transforming the continuous variables into their corresponding discrete variables (as given above), the corresponding discrete-time model in (4) is obtained that satisfy the corresponding constraints.

Proof: Suppose $x_e^{(k)}(p)$ and $z_e(p)$ satisfy the constraints in model (3) for the link capacities $u_e(p)$, $e \in A$, $k \in K$, $p \in [0, D]$. For every integral time step $p \in \{0, 1, 2, \dots, D - 1\}$ and time horizon D , $x_e^{(k)}(p)$ can be bounded as follows

$$x_e^{(k)}(p) = \int_p^{p+1} x_e^{(k)}(\xi) d\xi \leq \int_p^{p+1} z_e(\xi) d\xi = z_e(p)$$

$$z_e(p) = \int_p^{p+1} z_e(\xi) d\xi \leq \int_p^{p+1} u_e(\xi) d\xi = u_e(p)$$

It is easy to verify that flow conservation constraints still hold. Let $\theta \in \{0, 1, 2, \dots, D - 1\}$, then for each $i \in N$, $k \in K$ we obtain

$$\begin{aligned} & \sum_{e \in \delta_i^+} \sum_{p=0}^{\theta} x_e^{(k)}(p) - \sum_{e \in \delta_i^-} \sum_{p=0}^{\theta} x_e^{(k)}(p - d_e) \\ &= \sum_{e \in \delta_i^+} \sum_{p=0}^{\theta} \int_p^{p+1} x_e^{(k)}(\xi) d\xi - \sum_{e \in \delta_i^-} \sum_{p=0}^{\theta} \int_p^{p+1} x_e^{(k)}(\xi - d_e) d\xi \\ &= \sum_{e \in \delta_i^+} \int_0^{\theta} x_e^{(k)}(\xi) d\xi - \sum_{e \in \delta_i^-} \int_0^{\theta} x_e^{(k)}(\xi - d_e) d\xi \\ &= \int_0^{\theta} \left(\sum_{e \in \delta_i^+} x_e^{(k)}(\xi) - \sum_{e \in \delta_i^-} x_e^{(k)}(\xi - d_e) \right) d\xi \leq \sigma_i^{(k)} \end{aligned}$$

and for $\theta = D$, the above equation will be satisfied as an equality. Therefore regarding the objective function we obtain

$$\begin{aligned} \sum_{e \in A} \sum_{p=0}^{D-1} c_e z_e(p) &= \sum_{e \in A} \sum_{p=0}^{D-1} c_e \int_p^{p+1} z_e(\xi) d\xi \\ &= \sum_{e \in A} c_e \int_e^D z_e(\xi) d\xi \end{aligned}$$

Conversely, let $x_e^{(k)}(p)$ and $z_e(p)$ satisfy the constraints in model (4) for the link capacities $u_e(p)$. We define $x_e^{(k)}(\theta) = x_e^{(k)}(p)$, the coded packet $z_e(\theta) = z_e(p)$ and capacity $u_e(\theta) = u_e(p)$ for $\theta \in [p, p + 1]$. It is obvious that $x_e^{(k)}(p)$ and $z_e(p)$ satisfy the constraints in model (3) for the link capacities $u_e(p)$. \square

In summary, the above Lemma 1 shows that every continuous-time problem can be transformed to a corresponding discrete-time one.

It should be noted that the model given in (4), although is a discrete-time model, is still not a static model to be solved in polynomial time. Hence, our proposed algorithm is based on reduction of this problem to a static time-expanded problem that can be solved in polynomial time with respect to the time-expanded network.

In the following, we will use the discrete-time model for our analysis.

2.4 Time-expanded network

We now present a method for solving model (4). In order to solve the above formulated problem, we propose an approach based on transforming the problem into a delay-free problem by using the time-expanded network proposed in [7].

It should be noted that in [7], a continuous-time network with random linear coding is considered. However, the objective in [7] has been to find the code to be used over the sub-graph and finding the optimal sub-graph in continuous-time has not been addressed there. On the other hand, in this paper, our goal is to obtain the optimal sub-graph by considering the discrete time model in which delay times will take integer values and all are bounded in the time horizon D . It can be shown that the network of links with delay, $G = (V, A, D)$ can be reduced to a delay-free problem (i.e. static network) through expanding time leading to an auxiliary time-expanded network $G^D = (V^D, A^D)$. The time-expanded version of network G is a digraph $G^D = (V^D, A^D)$, wherein there is a copy of the nodes for each time step in the time horizon $\{0, 1, \dots, D\}$. Consequently, links are redrawn between these copies to indicate their traversal times. The formal definitions are as follows:

$$\begin{aligned} V^D &= \{v_p | v \in V, p = 0, 1, \dots, D\} \\ A' &= \{e_p = (v_p, u_{p+d_e}) | e = (u, v) \in A; \\ & \quad p = 0, 1, \dots, D - d_e; v_p, u_{p+d_e} \in V^D\} \\ A'' &= \{(v_p, v_{p+1}) | v_p, v_{p+1} \in V^D; \\ & \quad p = 0, 1, \dots, D - 1; v \in \{s\} \cup K\} \\ A^D &= A' \cup A'' \end{aligned}$$

Note that for a network G^D , $|V^D| = D|V|$ and

$$|A^D| \sum_{e \in A} (D - d_e + 1) + D|V| = (|V| + |A|)D + |A| - \sum_{e \in A} (d_e)$$

Since the maximum amount of buffering at node v is denoted by b_v , at each time instant at most b_v units of information can be stored in that node. Consequently, in the equivalent time-expanded network, the same amount of information is transformed from v_p to v_{p+1} . Therefore the capacity of the link connecting nodes v_p and v_{p+1} in the time-expanded network will be equal to b_v .

It can be easily verified that network G from the source s to destination $k \in K$ is equivalent to a corresponding network G^D from the source set s_p to the sink set k_p ; $p = 0, 1, \dots, D$. It would then be possible to transform the network in G^D into a multicast problem by introducing a super-source S (i.e. by converting s_p , $p = 0, 1, \dots, D$ to a single source). For example, the time-expanded graph corresponding to the graph shown in Fig. 1 is demonstrated in Fig. 2.

Consequently, finding an optimal sub-graph in a network with delays can be solved by finding a sub-graph optimal in the time-expanded graph. With regard to the definition of

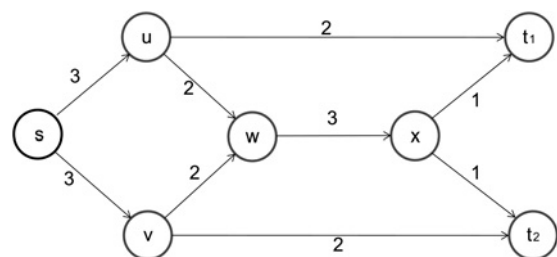


Fig. 1 Network with multicast from s to t_1 and t_2 . Each link is marked with its corresponding transmit time

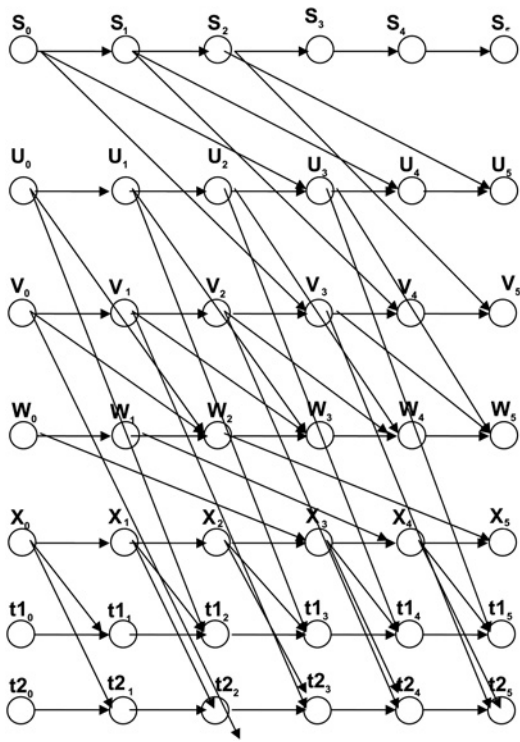


Fig. 2 Time-expanded network of Fig. 1

time-expanded network, G^D , we indicate the rate of flow on link $e_p = (u_p, v_{p+d_e})$ towards destination k by $x_{e_p}^{(k)}$. The coded packets and link capacities are also denoted by z_{e_p} and u_{e_p} , respectively.

The following equivalent selection formulation of the optimal sub-graph is then obtained

$$\min \sum_{e_p \in A^D} \sum_{p=0}^{D-d_e} c_e z_{e_p} \quad (5)$$

s.t.

$$x_{e_p}^{(k)} \leq z_{e_p}, \quad \forall e_p \in A^D, k \in K \quad (5a)$$

$$\sum_{e \in \delta_p^+} x_{e_p}^{(k)} - \sum_{e \in \delta_p^-} x_{e_p}^{(k)} = \sigma_i^{(k)}, \quad \forall i_p \in N^D, k \in K \quad (5b)$$

$$z_{e_p} \leq u_{e_p}, \quad \forall e_p \in A^D, p \in \{0, 1, \dots, D\} \quad (5c)$$

$$x_{e_p}^{(k)} \geq 0, \quad \forall e_p \in A^D, p \in \{0, 1, \dots, D\}, k \in K \quad (5d)$$

It is easy to verify that models (4) and (5) will be equivalent by setting $z_{e_p} = z_e(p)$, $x_{e_p}^{(k)} = x_e^{(k)}(p)$ and $u_{e_p} = u_e(p)$ for each $e_p \in A^D, p \in \{0, 1, \dots, D-d_e\}$.

Thus, we obtain the following lemma:

Lemma 2: Models (4) and (5) are equivalent in the sense that they both result in a solution with the same objective function and satisfying the same corresponding constraints.

2.5 Obtaining the distributed solution

In order to obtain an iterative solution to problem (5), we use the primal decomposition [14] and the dual-decomposition methods in [15, 16] to find upper and lower bounds,

respectively. As shown in [15] these two bounds iteratively converge to the optimal solution, as our problem satisfies the conditions for convergence [14].

2.5.1 Primal decomposition: In order to find an upper bound of model (5) through primal decomposition [14], we first derive the dual problem D . By designating $\gamma_{e_p}^{(k)}$, $\pi_{i_p}^{(k)}$ and λ_{e_p} as the dual variables associated with the constraints (5a), (5b) and (5c), respectively, we will have

$$D: \text{Max} \sum_{e_p \in A^D} \sum_{p=0}^{D-d_e} u_{e_p} \lambda_{e_p} + \sum_{i \in N^D} \sum_{k \in K} \sum_{p=0}^{D-d_e} \sigma_{i_p}^{(k)} \pi_{i_p}^{(k)} \quad (6)$$

s.t.

$$c_e + \lambda_{e_p} - \sum_{k \in K} \gamma_{e_p}^{(k)} \geq 0, \quad \forall e_p \in A^D, p \in \{0, 1, \dots, D-d_e\} \quad (6a)$$

$$\gamma_{e_p}^{(k)} - \pi_{\text{head}(e_p)}^k + \pi_{\text{tail}(e_p)}^k \geq 0, \quad \forall e_p \in A^D, p \in \{0, 1, \dots, D-d_e\}, k \in K \quad (6b)$$

$$\gamma_{e_p}^{(k)} \geq 0, \quad \lambda_{e_p} \geq 0, \quad \forall e_p \in A^D, p \in \{0, 1, \dots, D-d_e\}, k \in K \quad (6c)$$

It should be noted that when λ_{e_p} and $\gamma_{e_p}^{(k)}$ are fixed at a given arbitrary value, we obtain a linear programming problem in the variable $\pi_{i_p}^{(k)}$. Therefore problem (6) decomposes into two sub-problems. By first assuming constant values for λ_{e_p} and $\gamma_{e_p}^{(k)}$, sub-problem A is obtained.

Sub-problem A

$$\text{Max} \sum_{e_p \in A^D} \sum_{p=0}^{D-d_e} u_{e_p} \lambda_{e_p} + \phi(\gamma) \quad (7)$$

s.t.

$$c_e + \lambda_{e_p} - \sum_{k \in K} \gamma_{e_p}^{(k)} \geq 0, \quad \forall e_p \in A^D, p \in \{0, 1, \dots, D-d_e\} \quad (7a)$$

$$\gamma_{e_p}^{(k)} \geq 0, \quad \lambda_{e_p} \geq 0, \quad \forall e_p \in A^D, p \in \{0, 1, \dots, D-d_e\}, k \in K \quad (7b)$$

where

$$\phi(\gamma) = \text{Max} \sum_{i \in N^D} \sum_{k \in K} \sum_{p=0}^{D-d_e} \sigma_{i_p}^{(k)} \pi_{i_p}^{(k)} \quad (8)$$

s.t.

$$\gamma_{e_p}^{(k)} - \pi_{\text{head}(e_p)}^k + \pi_{\text{tail}(e_p)}^k \geq 0, \quad \forall e_p \in A^D, p \in \{0, 1, \dots, D-d_e\}, k \in K$$

Model (8) is the dual-standard minimum cost network flow that can be solved as described in [17]. Problem (7) can also be solved by using sub-gradient optimisation (see, e.g. [15, Section 6.3.1]). It should also be noted that the model

in (7) is now a static model which is independent of time. Therefore it can be solved in a distributed manner [15].

We assume x^* is the optimal value of dual variable in model (8). If $z_e := \max_{k \in K} \{x_e^{*k}\}$, $\forall e \in A$, then x^* and z^* satisfy the constraints of model (5). Therefore the objective function of model (5) obtained by substituting x^* and z^* (denoted by φ) will be an upper bound for the optimum value of this model.

2.5.2 Dual decomposition: Constraint (5a) is a coupling constraint such that, when relaxed, the optimisation problem decouples into two sub-problems corresponding to variables X and Z . Such relaxed problem is given by

$$\text{Min} \sum_{e_p \in A^D} \sum_{p=0}^{D-d_e} c_e z_{e_p} + \sum_{e_p \in A^D} \sum_{k \in K} \sum_{p=0}^{D-d_e} \gamma_{e_p}^{(k)} (x_{e_p}^{(k)} - z_{e_p})$$

subject to constraints (5b), (5c) and (5d). If the objective function (8) is rewritten as follows

$$\sum_{e_p \in A^D} \sum_{p=0}^{D-d_e} \left(c_e - \sum_{k \in K} \gamma_{e_p}^{(k)} \right) z_{e_p} + \sum_{e_p \in A^D} \sum_{k \in K} \sum_{p=0}^{D-d_e} \gamma_{e_p}^{(k)} x_{e_p}^{(k)}$$

the optimisation problem can be separated into two levels of optimisation. At the lower level, we have the sub-problems (i.e. the Lagrangians), one for each $k \in K$, in which (9) decouples. We denote by $g^{(k)}(\gamma)$, $k \in K$ the dual function obtained as the minimum value of the Lagrangian

$$g^{(k)}(\gamma) = \text{Min} \sum_{e_p \in A^D} \sum_{p=0}^{D-d_e} \gamma_{e_p}^{(k)} x_{e_p}^{(k)} \quad (9)$$

subject to constraints (5b) and (5d). The corresponding function for z is

$$g^{(0)}(\gamma) = \text{Min} \sum_{e_p \in A^D} \sum_{p=0}^{D-d_e} \left(c_e - \sum_{k \in K} \gamma_{e_p}^{(k)} \right) z_{e_p} \quad (10)$$

s.t.

$$0 \leq z_{e_p} \leq u_{e_p}, \quad \forall e_p \in A^D, p \in \{0, 1, \dots, D - d_e\}$$

Note that the sub-problem (9) is the shortest path problem, which can be solved using standard methods (see, e.g. [Chapter 5 in [14, 17]). In the following, we use a modification of the sub-gradient algorithm called the volume algorithm [18, 19] which leads to faster convergence in comparison with the sub-gradient algorithm.

2.5.3 Volume algorithm: *Step 0:* Start with a vector γ and solve (9) for each k to obtain $g^{(k)}(\gamma)$, $x^{(k)}$. Then, solve (10) to obtain $g^{(0)}(\gamma)$, z . Set $f = \sum_{k \in K} g^{(k)}(\gamma) + g^{(0)}(\gamma)$, $x^{(k)}(0) = x^{(k)}$, $z(0) = z$, and $q = 1$.

Step 1: Compute $\gamma_{e_p}^{(k)}(q) = [\gamma_{e_p}^{(k)} + s(x_{e_p}^{(k)} - z_{e_p})]^+$, where $[\cdot]^+$ indicates that γ must be non-negative. s is also a positive step-size to guarantee convergence of the algorithm [15, 20]. Solve (9) and (10) with the computed value of $\gamma(q)$, and let $x^{(k)}(q)$ and $z(q)$ be the solutions obtained resulting in corresponding values $g^{(k)}(\gamma(q))$, $g^{(0)}(\gamma(q))$. Let $f(q) = \sum_{k \in K} g^{(k)}(\gamma(q)) + g^{(0)}(\gamma(q))$. Then, $x^{(k)}$ and z are

updated as

$$x^{(k)} := \alpha x^{(k)}(q) + (1 - \alpha)x^{(k)}, \quad k \in K$$

$$z := \alpha z(q) + (1 - \alpha)z$$

where α is a real number between 0 and 1.

Step 2: If $f(q) > f$, update γ and f as

$$\gamma := \gamma(q)$$

$$f := f(q)$$

Let $q := q + 1$ and go to Step 1.

Note that the solution of (8) results in a lower bound of the objective function of problem (5).

2.5.4 Algorithm: The primal decomposition for model (5) results in an upper bound through decomposing the problem into two sub-problems that can be solved using sub-gradient optimisation. The dual decomposition for model (5) results in a lower bound through decomposing the problem into $|K| + 1$ sub-problems that are solved using the volume algorithm. By noting that our problem satisfies the conditions for convergence, the lower and upper bounds of the optimal value of the model (5) will both tend to the optimal value; a near-optimal solution of any required

Input: $G = (V, A, D)$, R, K, c, u, N, ϵ

Output: A near-optimal flow z^{opt} and bounds on the optimal value [LUB, UBD].

Begin

To build the time-expanded network G^D for the given network G

LBD := $-\infty$, UBD := ∞ , $q := 1$

repeat

begin

run sub-gradient algorithm for solving primal decomposition and obtain the value of $\bar{\varphi}$

if $\bar{\varphi} < \text{UBD}$ then

begin

UBD := $\bar{\varphi}$;

$z_e^{opt} := \max_{k \in K} \{x_e^{*k}\} \quad \forall e \in A$

end

run volume algorithm to solve dual decomposition and obtain value \bar{f}

LBD := $\max\{\bar{f}, \text{LBD}\}$

$q := q + 1$;

end

until $\frac{\text{UBD} - \text{LBD}}{\text{LBD}} \leq \epsilon$ or $q \leq N$

output z^{opt} , UBD and LBD

end.

Fig. 3 Decentralised algorithm for the optimal sub-graph selection-over network-coded networks with delay and limited-size buffering

accuracy, thus, can always be found within finite number of iterations. The algorithm is summarised in Fig. 3.

The algorithm considered in [3] is based on the Lagrangian dual problem of the model in [3] solved by sub-gradient optimisation. However, we consider an iterative solution to the problem that obtains lower and upper bounds of the solution by using primal and dual decompositions. Such approach is adopted in this case as the size of our model might become large enough (due to expansion of the dynamic problem over time and obtaining a static network). Finally, our algorithm terminates whenever the lower and upper bounds of the objective function become close enough.

It should be noted that in some cases, the solution obtained in [3] by sub-gradient method may not satisfy the constraint in the original model at every step, but our algorithm always leads to a feasible solution at every step. Moreover, using the volume method leads to a faster solution in comparison with the sub-gradient algorithm [18, 19].

In the traditional case that arcs do not incur any delay, the minimum cost sub-graph with network coding proposed by Lun *et al.* [3] in model (1) is a convex optimisation problem, with a polynomial-time solution. However, when link delays are taken into account, the size of our proposed time-expanded graph would not be polynomial in the size of original problem, as there are D copies of the network in the time-expanded version. Consequently, although the original problem is not polynomial, after time-expansion it will become a static polynomial problem with respect to the new time-expanded version.

3 Extension to the varying-cost scenario

In real networks, link cost functions are not necessarily fixed and may vary over time. In this section, we extend the problem to the case of networks with varying costs. In this case, we modify model (3) such that c_e is not constant and varies over time. As a result, the new objective function will be

$$\sum_{e \in A} \int_0^D c_e(\xi) z_e(\xi) d\xi$$

Model (5) will be subsequently modified as follows

$$\text{Min} \sum_{e \in A^D} \sum_{p=0}^{D-d_e} c_{e_p} z_{e_p} \quad (11)$$

s.t.

$$x_{e_p}^{(k)} \leq z_{e_p}, \quad \forall e_p \in A^D, p \in \{0, 1, \dots, D - d_e\}, k \in K \quad (11a)$$

$$z_{e_p} \leq u_{e_p}, \quad \forall e_p \in A^D, p \in \{0, 1, \dots, D - d_e\} \quad (11b)$$

$$\sum_{e \in \delta_p^+} x_{e_p}^{(k)} - \sum_{e \in \delta_p^-} x_{e_p}^{(k)} = \sigma_i^{(k)} \quad \forall i_p \in N^D, p \in \{0, 1, \dots, D - d_e\}, k \in K \quad (11c)$$

$$x_{e_p}^{(k)} \geq 0 \quad \forall e_p \in A^D, p \in \{0, 1, \dots, D - d_e\}, k \in K \quad (11d)$$

It is evident that model (11) can also be solved by the method proposed earlier.

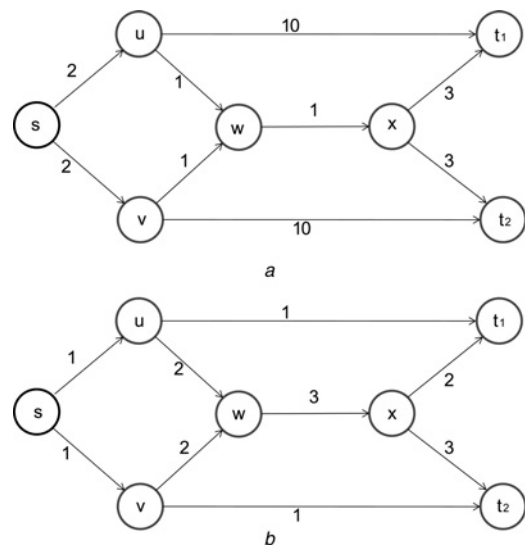


Fig. 4 Network with multicast from s to t_1 and t_2
 a Each link is marked with its cost per unit rate
 b Each link is marked with its transmit time

4 Simulation results: comparison with the minimum-cost multicast approach

In this section, we compare the performance of our algorithm with minimum-cost multicast algorithm described in [3]. It is expected that in the absence of any delays in the network, our algorithm will reduce to the minimum-cost multicast approach in [3]. Therefore MCM techniques can be considered as special cases in our algorithm.

For our simulations, we first consider the Butterfly network of Fig. 4 that denotes a typical scenario in which network coding is known to increase the multicast throughput [21]. Cost and delay of each link are shown in Figs. 4a and b, respectively. Assume that there is only one source s and two multicast receivers, t_1 and t_2 requesting same flows. We consider three different scenarios with rates of 1, 2 and 3 units. We also consider three different time horizons $D = 2$, $D = 8$ and $D = 9$ and obtain the optimum solution in each scenario as shown in Table 1.

Considering network variations, the simulations are performed for the following four scenarios as well. In the first case, no link delay and buffer size are assumed that naturally leads to the optimal sub-graph solution obtained by Lun's algorithm [3]. In the second case, we include buffer size and link costs; however, assume that their values do not change over time. In the third case, we assume link costs to be constant but buffer sizes vary over time (randomly chosen from a uniform distribution in the interval three to ten). Finally, we have also considered a fourth scenario in which both buffer size and link costs vary over time, chosen randomly from uniform distributions in the intervals three to ten and one to five, respectively. The results are shown in Table 2.

Table 1 Overall network cost for the cases 1, 2 and 3

Butterfly network	$R=1$	$R=1$	$R=1$	$R=2$	$R=2$	$R=3$	$R=3$
	$D=2$	$D=8$	$D=9$	$D=8$	$D=9$	$D=8$	$D=9$
first case	24	17	10	41	27	-	50
second case	24	17	10	34	20	51	30
third case	10	10	10	33	33	-	-

Table 2 Overall network cost for the case 4

Butterfly network	$R = 1$ $D = 2$	$R = 1$ $D = 8$	$R = 1$ $D = 9$	$R = 2$ $D = 8$	$R = 2$ $D = 9$	$R = 3$ $D = 8$	$R = 3$ $D = 9$
fourth case	9	4	5	8	10	12	12

The results of Tables 1 and 2 can be summarised as follows:

1. For small values of time horizon, sending data through low-delay paths may lead to higher cost results. In addition, if time horizons are assigned to very small values, it is possible to encounter scenarios in which finding an optimal sub-graph is impossible.
2. By increasing the time horizon, the proposed algorithm has more flexibility in choosing lower cost solutions.
3. Increasing the buffer size of nodes can reduce the overall cost, as intermediate nodes have the flexibility to store incoming packets in their buffer when their lower-cost output links are occupied. Such nodes can then send out the stored packet at a later time instant at which the given lower-cost links become available again.
4. When source data rate is increased beyond some limit, selection of optimal sub-graph becomes impossible. However, by proper choice of buffer size, the proposed model may still lead to optimal sub-graphs while such solution may not be found for the traditional solutions (i.e. note the case of $R = 3$ in Table 1).

It should be noted that since the algorithm proposed in [3] finds the optimal sub-graph per time unit, we should compare the average value of the cost function obtained from the proposed algorithm over the time horizon D . In summary, the proposed approach may naturally lead to solutions with higher cost values if delay constraints are too strict. However, they provide higher flexibility through packet storage at intermediate nodes which may lead to lower cost solutions when enough time horizon is taken into account.

In the next step, we tested our algorithms on random graphs and compared the total cost for both network-coding-based and routing-based solutions over the time-expanded network described in Section 2.4. As expected, in routing-based case the minimum-cost multicast solution is equivalent to solving the Steiner tree problem on directed graphs which is NP-hard [22]. Therefore one approach to solving the routing-based problem is using polynomial-time heuristics such as the multicast incremental power algorithm proposed in [23]. The graphs, and their associated link weights, are obtained from [24]. We first generate $|N|$ nodes randomly, according to a uniform distribution in a 10×10 square. The capacity of each link and buffer size of each

Table 3 Overall cost of random multicast connections of unit rate for random networks of different sizes

Original network size	Expanded-time network size	Approach	2 sinks	4 sinks
10 nodes 53 arcs	60 nodes 259 arcs	network coded	6.65	10.99
20 nodes 220 arcs	120 nodes 973 arcs	routed network	11.81	17.6
		network coded	4.96	10.35
		routed network	6.83	13.47

Source and sink nodes are selected randomly among all nodes (time horizon $D = 5$)

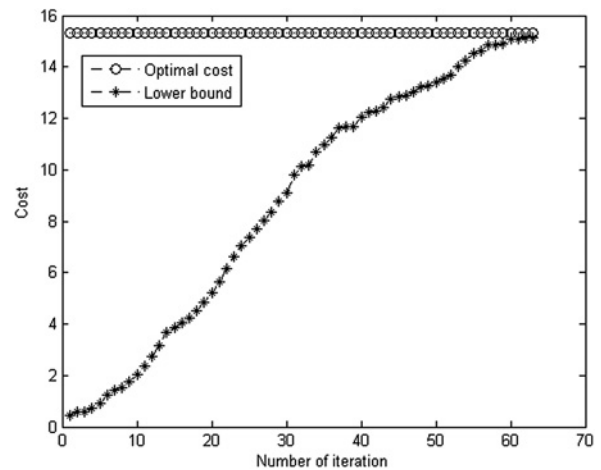


Fig. 5 Dynamic network of 10 nodes with two-sinks multicast with time horizon 5, that is, the time-expanded network contains 60 nodes and 235 arcs

By using of the volume algorithm described in Section 2.5, obtained lower bound of optimal cost

node in the network is one unit and the transmit rate is $R = 1$. Moreover, link costs and delays were assigned randomly and the source node and the set of multicast receivers were randomly chosen as well.

The results of the simulations are shown in Table 3. It can be observed that depending on the network and the size of the multicast group, the total cost reduction ranges from 23 to 43%. Simulations of our decentralised algorithm have been performed for a network of 10 nodes and multicast group of two sinks. The corresponding time-expanded network contains 60 nodes and 235 arcs. Fig. 5 shows the result of applying the volume algorithm described in Section 2.5 to problem (5).

It should be noted that the value of parameter α in the volume algorithm is set based on ideas proposed in [18, 19] as follows. Let $\mathbf{v} = \mathbf{x}^{(k)} - \mathbf{z}$, $v(q) = x^{(k)}(q) - z(q)$, and let α_{\max} be an upper bound for α . Then, we would compute α_{opt} as the value that minimises $\|\alpha v(q) + (1 - \alpha)v\|$. If $\alpha_{\text{opt}} < 0$ we would set $\alpha = (\alpha_{\max}/10)$, otherwise we would set $\alpha = \min\{\alpha_{\max}, \alpha_{\text{opt}}\}$. We start with the value of $\alpha_{\max} = 0.1$ and then decrease its value as needed. Also, the step-size in volume algorithm is set at $s = n^{-0.3}$. The optimal cost obtained for problem (5) by using the central method is then equal to 15.34. It can be observed that in fewer than 60 iterations, the cost obtained from the volume algorithm is within 5% of the optimal value.

5 Conclusion

In this paper, we addressed the problem of finding optimal sub-graphs in network coding over networks with link delays and buffering capability at intermediate nodes. Having formulated the problem of continuous delay network coding, we described the relationship between continuous and discrete models in coded networks. The derived discrete model was then converted to an expanded delay-free network which can be solved through well-known algorithms. As shown by the simulation results, inclusion of node buffering in our analysis may lead to significant cost reduction. This is due to the fact that with the aid of buffering, intermediate nodes can adopt a more ‘opportunistic’ transmission scheme over time. In addition,

the proposed scheme leads to optimal sub-graphs when such solutions may not be available for the traditional schemes that do not take delay and buffering into account.

6 Acknowledgments

This work was supported in part by Iran National Science Foundation under grant number 87041174.

7 References

- 1 Ahlswede, R., Cai, N., Li, S.Y.R., Yeung, R.W.: 'Network information flow', *IEEE Trans. Inf. Theory*, 2000, **46**, (4), pp. 1204–1216
- 2 Koetter, R., Medard, M.: 'An algebraic approach to network coding', *IEEE/ACM Trans. Netw.*, 2003, **11**, (5), pp. 782–795
- 3 Lun, D.S., Ratnakar, N., Medard, M., *et al.*: 'Minimum-cost multicast over coded packet networks', *IEEE Trans. Inf. Theory*, 2006, **52**, (6), pp. 2608–2623
- 4 Li, S.Y.R., Yeung, R.W., Cai, N.: 'Linear network coding', *IEEE Trans. Inf. Theory*, 2003, **49**, (2), pp. 371–381
- 5 Jaggi, S., Sanders, P., Chou, P.A., *et al.*: 'Polynomial time algorithms for multicast', *IEEE Trans. Inf. Theory*, 2005, **51**, (6), pp. 1973–1982
- 6 Chou, P.A., Wu, Y., Jain, K.: 'Practical network coding'. Allerton Conf. Comm. Cont. Comp., 2003
- 7 Wu, Y.: 'A trellis connectivity analysis of random linear network coding with buffering'. Proc. IEEE Int. Conf. Information Theory, 2006, no. 4036067, pp. 768–772
- 8 Lun, D.S., Medard, M., Ho, T., Koetter, R.: 'Network coding with a cost criterion'. Proc. ISITA, October 2004, pp. 1232–1237
- 9 Ho, T., Koetter, R., Medard, M., Effros, M., Shi, J., Karger, D.: 'A random linear network coding approach to multicast', *IEEE Trans. Inf. Theory*, 2006, **52**, (10), pp. 4413–4430
- 10 Cui, T., Ho, T.: 'Minimum cost integral network coding'. Proc. IEEE Int. Conf. Information Theory, 2007, pp. 2736–2740
- 11 Wu, Y., Chiang, M., Kung, S.: 'Distributed utility maximization for network coding based multicasting: a critical cut approach'. Proc. Second Network Coding Workshop, 2006
- 12 Xi, Y., Yeh, E.M.: 'Distributed algorithms for minimum cost with network coding', *IEEE/ACM Trans. Netw.*, 2010, **18**, (2), pp. 379–392
- 13 Fonoberova, M., Lozovanu, D.: 'Optimal dynamic multicommodity flows in networks', *Electron. Notes Discrete Math.*, 2006, **25**, pp. 93–100
- 14 Bazaraa, M.S., Jarvis, J.J., Sherali, H.D.: 'Linear programming and network flows' (Wiley, New York, 1990, 2nd edn.)
- 15 Bertsekas, D.P.: 'Nonlinear programming' (Athena Scientific, Belmont, MA, 1995, 2nd edn.)
- 16 Palomar, D., Chiang, M.: 'A tutorial on decomposition method and distributed network resource allocation', *IEEE J. Sel. Areas Commun.*, 2006, **24**, pp. 1439–1451
- 17 Ahuja, R.K., Magnanti, T.L., Orlin, J.B.: 'Network flows: theory, algorithms, and applications' (Prentice-Hall, Upper Saddle River, NJ, 1993)
- 18 Barahona, F., Anbil, R.: 'The volume algorithm: producing primal solutions with a subgradient method', *Math. Program.*, 2000, **87**, (3, Ser. A), pp. 385–399
- 19 Bahiense, L., Maculan, N., Sagastizabal, C.: 'The volume algorithm revisited: relation with bundle methods', *Math. Program*, 2002, Ser. A, **94**, pp. 41–69
- 20 Shor, N.Z.: 'Minimization methods for non-differentiable functions' (Springer, Berlin, Germany, 1985)
- 21 Ho, T., Lun, D.S.: 'Network coding an introduction' (Cambridge University Press, 2008)
- 22 Ramanathan, S.: 'Multicast tree generation in networks with asymmetric links', *IEEE/ACM Trans. Netw.*, 1996, **4**, (4), pp. 558–568
- 23 Wieselthier, J.E., Nguyen, G.D., Ephremides, A.: 'Energy-efficient broadcast and multicast trees in wireless networks', *Mobile Netw. Appl.*, 2002, **7**, pp. 481–492
- 24 <http://www.ifp.illinois.edu/~koetter/NWC/index.html>