Multimed Tools Appl (2014) 71:1975–1998
DOI 10.1007/s11042-012-1321-8

Content-aware optimization on rate-distortion and network traffic for scalable video multicast networks

Junni Zou · Lu Jiang · Chenglin Li

Published online: 22 December 2012

Abstract This paper aims to optimize the content-aware prioritization of scalable video multicast, which is coupled with multipath streaming and network coding based routing. It constructs multiple layer distribution meshes for the scalable video stream to minimize the total video distortion at all the receivers, determines the base layer meshes with minimum costs to maintain application-layer QoS and the layer synchronization of SVC streaming, and improves the network throughput by encouraging path-overlapping transmissions and thus allowing bandwidth sharing among different receivers for the same video layer by utilizing network coding. The targeted problem is formulated into a minimization programming in which the quality variation between layers, the transmission cost of the base layer, as well as the efficient resource utilization are jointly considered. By decomposition and dual approach, the global convex problem is solved by a two-level decentralized iterative algorithm. The implementation of the distributed algorithm is discussed with regard

The work has been partially supported by the NSFC grants (No. 61271211, No. 60972055), and the Research Program from Shanghai Science and Technology Commission (No. 11510707000, No. 11QA1402600).

J. Zou (✉)
Department of Electrical and Computer Engineering, University of California,
San Diego, CA 92093, USA
e-mail: zoujunni@gmail.com

J. Zou
Key Laboratory of Special Fiber Optics and Optical Access Networks, Shanghai University,
Shanghai, 200072, China

L. Jiang
Department of Communication and Information Engineering,
Shanghai University, Shanghai 200072, China

C. Li
Department of Electronic Engineering, Shanghai Jiao Tong University,
Shanghai 200240, China

to the communication overhead, and the convergence performance is validated by numerical experiments. Packet-level simulations demonstrate that the proposed algorithm could approximately achieve the maximum flow rates determined by Max-Flow Min-Cut Theorem and benefit the overall received video quality.

Keywords Scalable video coding · Multicast · Network coding · Rate distortion · Distributed algorithm

1 Introduction

Multirate multicast is superior to single rate multicast for media streaming distribution to a set of heterogeneous receivers [18]. Multirate multicast allows receivers to subscribe contents in compliance with the available bandwidth by joining a proper subset of multicast sessions. With the development of layered and scalable video coding [25], layered multicast emerges a variant of multirate multicast for scalable media streaming [3]. In layered multicast, on one hand, video can be transmitted and decoded at multiple bit rates with progressively improved video quality. On the other hand, rate adaptation is implemented at both the sender/receiver and intermediate network nodes, while achieving highly efficient video rate-distortion performance. Therefore, joint optimization on multirate flow control and video distortion is of paramount importance in scalable streaming dissemination.

An SVC elementary stream is encoded to contain an H.264/AVC compatible base layer and represent the bit stream in the fully scalable representation. Utilizing SVC technique, a scalable bit stream could be represented in two different ways: a layered representation (layered scalable) or a flexible combined scalability (fully scalable) [2]. Generally, the full scalability can benefit scenarios of unicast, where the target stream can be extracted at any bit rate from the SVC elementary stream in terms of single receiver's capability. In comparison, the layered scalability can benefit multicast distribution by offering simple adaptation to heterogeneous receivers, i.e., different receivers can subscribe to different combinations of layers under the constraints of network capacity and layer dependency. For practical streaming multicast, we adopt the layered scalability and assume that the SVC video stream is encoded into a set of multiple layers where higher layers can be viewed as progressively refinable layers for the lower layers to update the video from one quality to the next [2]. Rooted in a base layer, an SVC stream extends one or multiple enhancement layers with a flexible multi-dimension layer structure (at leat one dimension from temporal, spatial, or SNR) to provide various operating points in spatial resolution, temporal frame rate, and video reconstruction quality.

Rate control of scalable video streaming has been studied extensively in the past [11, 13, 27, 30]. For example, Zhu et al. [30] presented a packet-based rate adaption scheme for minimizing total distortion of multiple video streams for application-layer multicast with multipath transmission. van der Schaar et al. [27] proposed a packet-based channel access scheme for scalable streaming over wireless networks. A message-based pricing and access coordination scheme was presented in [11]. To support heterogeneous device capability in the video multicasting/broadcasting, statistical multiplexing for layered multicast was investigated with a complexity measure among all programs in all layers [13]. These rate control methods could

improve the performance of scalable video streaming over networks, however, they used predetermined distribution trees to improve the network throughput and overall video quality, which might cause decoding problem in scalable video streaming over networks since the synchronization among SVC layers has not been adequately addressed. For example, along predetermined distribution trees, video packets from higher layers may arrive before or without packets from lower layers, which will cause decoding failure. To address this layer synchronization issue and allocate paths with lower cost to lower layers according to different SVC streams, in this paper, we study rate-distortion minimization problem for scalable streaming multicast networks, where each receiver can have multiple alternative paths through the coded network (i.e., networks that use network coding) to receive the subscribed SVC layers in an incremental order. Also, the network coding assisted multirate multicast is employed to enhance network transmission performance in a further way.

The first optimization model for multirate multicast problems was proposed by Kar et al. [14, 15], and a distributed algorithm for the receiver to receive service at any rate within a continuous set of rates was proposed in [24]. Extending from one source scenario to maximize the overall utility of multiple sources constrained by the sources' transmission rates, a flow control and optimization scheme is presented in [20]. Based on this approach, a number of source-oriented rate control schemes have been developed [11, 27, 30], which have been previously introduced as rate control schemes for scalable video streaming. The multipath routing combined with congestion control was studied in [12]. Also, inter-session fairness for layered video multicast was investigated by considering layer-based congestion sensitivity, which lets different video layers have different sensitivity to congestion [17] to address the layer synchronization issue. However, these existing methods on network performance optimization have been focused on only resource allocation among receivers, the problem of utility maximization for heterogeneous receivers to subscribe to multiple video coding layers with prioritized multirate multicasting has not been adequately addressed.

Network coding represents a novel paradigm in information theory that first proposed by Ahlswede et al. in 2000 [1]. It extends the functionality of network nodes from storing/forwarding packets to performing algebraic operations on data received. Li et al. [19] proved that the max-flow multicast throughput can be reached through the linear network coding. Chen et al. [7] developed two adaptive rate control algorithms by considering networks with and without coding subgraphs. Wu [28] extended network utility maximization (optimizing QoS of the entire network based on a specific utility function) to network coding based multicasting. The authors in paper [29] attempted to address the layered multicasting problem by including network coding and multipath constraints in the objective function, and proposed a solution called LION algorithm. However, they simply formulated it as an integer linear programming without utility maximization and the prioritized path costs of different layered multicasting groups. Moreover, they only provided a heuristic approach instead of a rigorous distributed algorithm with global optimality. As a further improvement, we [31] proposed a prioritized flow optimization formulation for SVC and multicast over heterogeneous networks, which used the path cost and prices of each layer as the priority parameters to capture layer synchronization of SVC streaming. Although path cost of each layer and the layer synchronization problem

have been considered in this work, the successful decoding of the based layer cannot be guaranteed at all times especially when the path cost of higher layers are much smaller than that of the base layer. Moreover, the overall communication network in this work is simply modeled by a generalized network utility maximization problem, which did not take into account distortion property for video applications. In this paper, we consider rate-distortion modeling from the perspective of application-layer QoS, and improve optimization formulation by minimizing the total received video distortions of all receivers while also emphasizing on minimizing both the delay of the base layer to guarantee a basic quality for all receivers and minimizing the actual bandwidth assigned to all SVC layers to consider the efficient network resource utilization.

In this paper, we consider content-aware prioritization of scalable video coding and investigate how it could be coupled with multipath video streaming and network coding based routing to achieve optimum performance. To minimize the total video distortion at all the receivers, we propose an efficient flow control and resource allocation scheme. It constructs multiple layer distribution meshes for the scalable video stream with multipath routing, and determines the base layer meshes with minimum costs so as to guarantee application-layer QoS and tackle the layer synchronization issue of SVC streaming. Also, a specific strategy to efficiently allocate paths for receivers with minimum bandwidth consumption is proposed to improve the network throughput, which encourages path-overlapping transmissions and allows bandwidth sharing among different receivers for the same video layer by utilizing network coding. We formulate the flow control problem into a minimization programming in which the quality variation between layers, the transmission cost of the base layer, as well as the efficient resource utilization are jointly considered. By using primal decomposition and primal-dual approach, the global convex problem is solved by a two-level decentralized iterative algorithm. The implementation of the distributed algorithm is discussed with regard to the communication overhead, and the convergence performance of the proposed algorithm is validated by numerical experiments. Packet-level simulations demonstrate that the algorithm could approximately achieve the maximum flow rates determined by Max-Flow Min-Cut Theorem and benefit overall received video quality.

The remainder of this paper is organized as follows. Section 2 describes the motivation for SVC streaming multicast networks. In Section 3, we formulate the problem of rate allocation and performance optimization for scalable video coding and multicast over networks. In Section 4, we propose a decentralized algorithm for the original scheme, and discuss implementation issues related to the algorithm. Numerical and simulation results are presented in Section 6. Finally, the paper concludes in Section 7.

2 Motivations

Our motivations in this paper could be derived from two aspects. The first one is related to the layer synchronization of SVC streaming, which requires that each receiver subscribes to scalable layers in an incremental order, since the successfully received higher layers cannot be decoded without the presence of lower layers. In other words, there exists layer dependency and priority constraint among scalable

video layers, where higher layers are dependent on the low layers and the low layers are with higher priority than the higher layers. Therefore, within the context of SVC, maximizing the total number of received layers or the overall throughput cannot guarantee the quality of video streaming, since the decoding of the enhancement layers depends on packets of the base layer. The layer synchronization requirement would lead the total utility of maximum received layers into suboptimal performance, where some higher layers, though successfully received, cannot be decoded because of the lack of their corresponding lower layers. Due to the lack of layer dependency and priority consideration in constructing multicasting paths, the higher layers may overwhelm the lower layers by low path costs and prices. That is, when packets of dependent lower layers are not all available till playback time, the packets of higher layers will have to be discarded, even if the bandwidth has been allocated for higher layers to maximize total utility for all receivers. This unexpected result obviously deviates from the original optimization objective.

To clearly illustrate this problem, we take a simple example. The network shown in Fig. 1a contains one source S, four relay nodes $R_1 \sim R_4$, and two receivers T_1, T_2, with the capacity marked on each link. Assume the source generates an SVC stream into three layers, each with rate of 2 (data units/second). According to the Max-Flow Min-Cut Theorem, the max flow to receivers T_1 and T_2 are 4 and 6. Thus, T_1 and T_2 can subscribe to 2 layers and 3 layers respectively. For simple specification, assume the data suffer similar propagation delay along each link.

When the LION model [29] is adopted to maximize the network throughput, its distribution meshes of the base layer and the first enhancement layer are shown in Fig.1b and c (with solid lines for T_1 and dashed lines for T_2, and the associated numbers on each link specifying the bandwidth assigned for T_1/T_2 on each layer). Observing the distribution mesh for T_1, we can find that it crosses four links ($S \rightarrow R_2 \rightarrow R_3 \rightarrow R_4 \rightarrow T_1$) at the base layer, while it only takes the propagation delay of two links ($S \rightarrow R_1 \rightarrow T_1$) at the enhancement layer, which introduces a reversed propagation delay of two links between the base layer and the enhancement layer. The layer synchronization of SVC decoding in T_1 will be greatly influenced

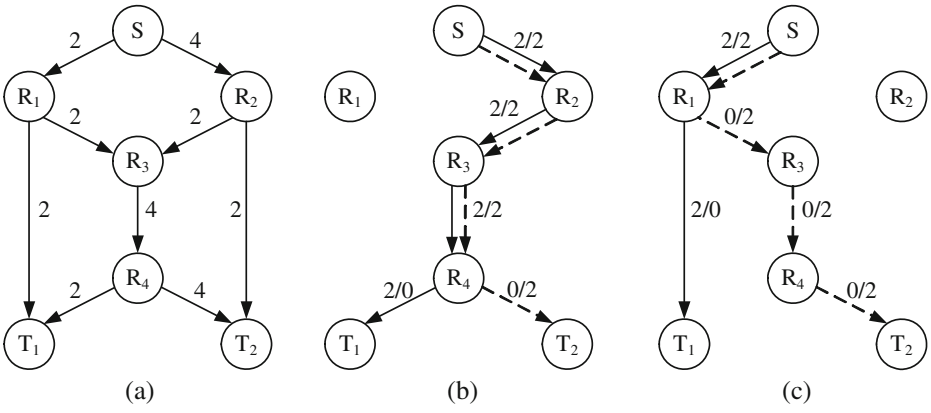


Fig. 1 An example of flow distribution meshes, where **a** is the initial topology, **b** and **c** are distribution meshes constructed by LION algorithm for the base layer and the first enhancement layer, respectively

by such reversed propagation delay, resulting in heavy buffer management and decoder burden. As the reversed propagation delay between lower and higher layers increases, the burden of buffer and decoder at the receivers would also be increased in order to have the higher layer successfully decoded. Moreover, this dilemma would be more critical when either the scale of the network or the number of total scalable video layers becomes large.

The second issue is associated with efficient bandwidth utilization. Under multi-path routing mechanism, each receiver would have multiple candidate paths from the source to receive the video streams. To receive the same layer with minimum bandwidth consumption, the paths that contain more joint links with other receivers' paths are preferred. As shown in Fig. 2a, the example topology consists of a source node S, three relay nodes $R_1 \sim R_3$, and three receiver nodes $T_1 \sim T_3$, and the available capacity in the number of packets is also marked on each link. Suppose the base layer has 3 packets to be transmitted. Also, assume the data suffer similar propagation delay along each link. Figure 2b and c display the distribution meshes for the base layer by two different routing strategies. We can find that, although three receivers successfully achieve the base layer with roughly similar latency in the two strategies, their aggregate bandwidth consumptions are quite different (the former consumes 17, while the latter uses 14). The delivery of the base layer packets with two strategies are shown in Fig. 2d and e, where a, b and c denote three packets in the base layer and $b + c$ corresponds to the packet after network coding operation. It is observed from Fig. 2e that due to the selection of overlapping paths $S \rightarrow R_2 \rightarrow T_1$ and $S \rightarrow R_2 \rightarrow T_3$ as well as the employment of network coding, the latter solution

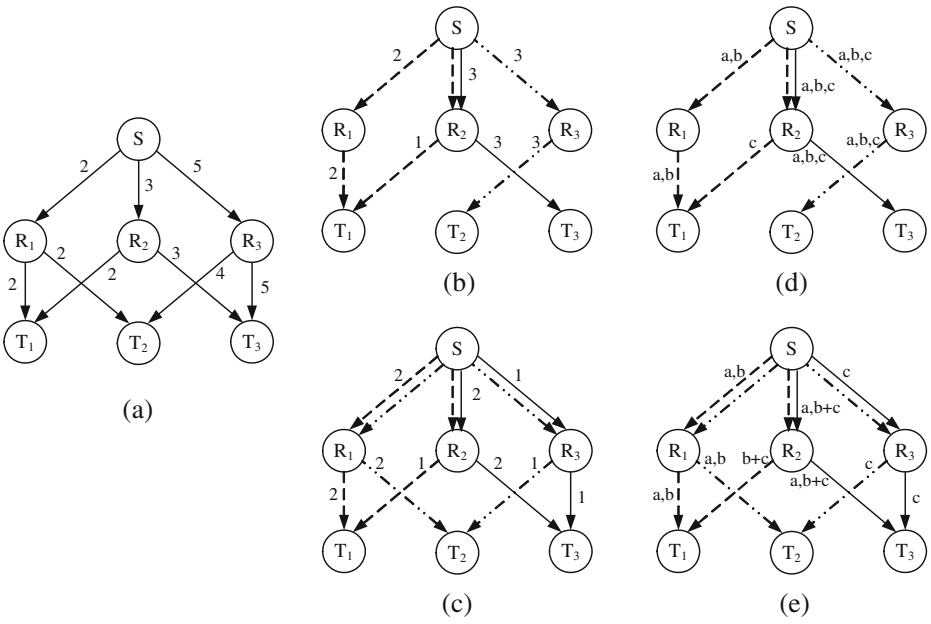


Fig. 2 An example of distribution meshes on the base layer by two different routing solutions, where **a** is the topology, **b** and **c** are the base layer mesh, **d** and **e** are packet transmissions

utilizes less bandwidth at the base layer, thus leaves more available resource for the higher layers.

3 Problem statement

3.1 Notations

The video distribution network can be modeled as a directed graph $G(V, E)$, where V is the set of nodes and E is the set of directed links. The set V comprises three kinds of nodes: S, R and T, representing the set of source nodes, relay nodes and receiver nodes respectively. The SVC stream is encoded into M layers, with each layer m corresponding to a multicast session at expected transmission rate B_m. For any link (i, j), let c_{ij} denote its capacity, and f_{ij}^m represent the bandwidth consumed on layer m.

Suppose from the source to receiver t there exist multiple alternative paths $P(t)$. For each receiver t, let $R_{t,k}^m$ denote the information flow rate assigned to its k-th path for transmitting packets of layer m. As a path consists of consecutive links, we use a matrix Z^t to denote whether the links are included in t's paths:

$$Z_{k,ij}^{tm} = \begin{cases} 1, & \text{if edge } (i, j) \in \text{path } k \text{ on layer } m\ ; \\ 0, & \text{otherwise.} \end{cases}$$

The ultimate goal of video streaming is to provide receiver the best video quality. To estimate the quality of the SVC video stream that is received by each destination node, in this work, we take the rate-distortion model in [26]:

$$D_e(R_e) = \frac{\theta}{R_e - R_0} + D_0$$

where D_e is the distortion of the encoded video sequence, measured by the mean squared error (MSE), and R_e is the encoded rate. The variables θ, R_0 and D_0 are the parameters of the R-D model.

When receiver t accesses to a new layer m, its receiving rate increases from R to $R + \Delta R$. By Taylor expansion, we approximate $D_e(R + \Delta R)$ by the first two terms of its Taylor series, the corresponding quality variation between layers goes as follows:

$$\begin{aligned} \Delta D_e &= D_e(R + \Delta R) - D_e(R) \\ &= D_e'(R) \cdot \Delta R + \frac{1}{2} D_e''(R) \cdot \Delta R^2 + o(\Delta R^2) \\ &\approx -\frac{\theta}{(R - R_0)^2} \cdot \Delta R + \frac{\theta}{(R - R_0)^3} \cdot \Delta R^2 \end{aligned}$$

Namely, for any receiver t with flow rate R_t^m on layer m, its distortion decrement can be described as a strictly convex function of R_t^m:

$$\Delta D_e(R_t^m) = -\frac{\theta}{(\sum_{i=0}^{m-1} R_t^i - R_0)^2} \cdot R_t^m + \frac{\theta}{(\sum_{i=0}^{m-1} R_t^i - R_0)^3} \cdot (R_t^m)^2 \qquad (1)$$

Generally, receiver t has multiple alternative paths to join the multicast session m, but not all these paths are optimal ones. Analogous to practical routing, the optimal

paths can be chosen in a variety ways based on different considerations, such as delay, resource usage or commercial charge. Inspired by the generic cost function definition [4], we propose the following path cost function:

$$\rho(R_{t,k}^m) = \sum_{(i,j)\in E} z_{k,ij}^{tm} \cdot \frac{R_{t,k}^m}{c_{ij} - R_{t,k}^m} + d_k^t \cdot R_{t,k}^m \tag{2}$$

According to [4], receiver t's congestion in terms of queuing delay on each link in layer m is a function of ongoing information flow rate $R_{t,k}^m$ and the capacity c_{ij} of that link. Using M/M/1 queuing model [16], the average queuing delay on each link can be expressed by $1/(c_{ij} - R_{t,k}^m)$, and the total queuing delay on that link becomes $R_{t,k}^m/(c_{ij} - R_{t,k}^m)$. Consequently, the first part of (2) represents the sum of queuing delay at links that belongs to that path. In the second term, d_k^t is a parameter corresponding to the average propagation delay over path k normalized by the average packet size. Therefore, the second term, $d_k^t \cdot R_{t,k}^m$ denotes the propagation delay on path k. With this definition, $\rho(R_{t,k}^m)$ denotes the end to end delay of information flow within layer m transmitting to receiver t along its k-th path and is a differentiable and convex function.

3.2 Optimization problem

For a given SVC streaming multicast network, we aim at maximizing the overall video quality (i.e., minimizing the total video distortion) of all receivers, while satisfying content priority of the base layer and minimum bandwidth utilization at all the layers. Mathematically, it can be formulated as:

P1: minimize $O(\mathbf{R}, \mathbf{f})$

$$= \sum_{t\in T}\sum_{m\in M} \Delta D_e \left(\sum_{k\in P(t)} R_{t,k}^m \right) + \sum_{t\in T}\sum_{k\in P(t)} \rho(R_{t,k}^1) + \sum_{m\in M}\sum_{(i,j)\in E} f_{ij}^m$$

subject to

1) $\sum_{k\in P(t)} \left(Z_{k,ij}^{tm} \cdot R_{t,k}^m \right) \le f_{ij}^m,\ \forall (i,j) \in E,\ \forall m \in M,\ \forall t \in T;$

2) $\sum_{m\in M} f_{ij}^m \le c_{ij},\ \forall (i,j) \in E;$

3) $0 \le \sum_{k\in P(t)} R_{t,k}^m \le B_m,\ \forall m \in M,\ \forall t \in T;$

The objective function $O(\mathbf{R}, \mathbf{f})$ consists of three parts. The first term represents the total quality variation between layers. The second term defines the overall end-to-end latency for the base layer dissemination. As the base layer makes predominated contribution in video data reconstruction, we emphasize on minimizing the delay of the base layer to guarantee a basic quality for all the receivers. The last term denotes the bandwidth assigned at all the layers. Clearly, it should be diminished as much as possible on the premise that all the receivers could successfully receive

their desired contents. For an optimal overall video quality, we attempt to seek an aggregate minimization solution that takes into account these three factors.

Constraint 1) represents the relationship between information flow rate and actual bandwidth consumption within each layer on each link, where network coding is applied to information flows of the same video layer. With network coding, different receivers will not compete for link bandwidth within the same session. Therefore, the actual bandwidth consumption on link (i, j) for layer m is equal to the largest information flow rate of all the receivers.

Constraint 2) ensures that the total bandwidth consumption of each link on different layers do not exceed the link capacity. Constraint 3) gives the upper bound of the information flow rate allocated to each receiver at each layer, i.e. for each receiver, the sum of information flow rate for transmitting layer m over all $P(t)$ paths cannot exceed the expected transmission rate B_m.

Define $\mathbf{R}_t = [R^1_{t,1}, \cdots, R^1_{t,|P(t)|}, R^2_{t,1}, \cdots, R^2_{t,|P(t)|}, \cdots, R^M_{t,1}, \cdots, R^M_{t,|P(t)|}]$ and $\mathbf{R} = [\mathbf{R}_1, \cdots, \mathbf{R}_T]^T$. Also let $\mathbb{R}_t = \left\{ \mathbf{R}_t \;\middle|\; 0 \le \sum_{k \in P(t)} R^m_{t,k} \le B_m \text{ , for all } m \text{ and } k \right\}$, $t \in T$, and $\mathbb{R}$ denote the Cartesian product of $\mathbb{R}_t$ $(t \in T)$, then Problem **P1** can be rewritten as:

$$\textbf{P2:} \quad \underset{\mathbf{R} \in \mathbb{R}}{\text{minimize}} \quad \sum_{t \in T} \sum_{m \in M} O(\mathbf{R}, \mathbf{f})$$

subject to

$$1) \quad \sum_{k \in P(t)} \left(Z^{tm}_{k,ij} \cdot R^m_{t,k} \right) \le f^m_{ij}, \ \forall (i, j) \in E, \ \forall m \in M, \ \forall t \in T;$$

$$2) \quad \sum_{m \in M} f^m_{ij} \le c_{ij}, \ \forall (i, j) \in E. \tag{3}$$

It can be verified that the objective function and the constraint set in **P2** are all convex [6]. Thus, there exists an unique optimal solution to **P2** which can be easily obtained by the centralized algorithms. However, the drawback of a centralized solution is that it requires a central node to collect global information such as the assigned flow rates on all links, and to perform all the computations. Such solution could be very costly and sometimes infeasible in practice. As the network size grows, it is preferable to solve the problem in a distributed manner.

4 Distributed algorithm

4.1 Primal decomposition

It is difficult to directly solve the problem **P2** with Lagrange duality, because of the interaction between variables f^m_{ij} and $R^m_{t,k}$ in Constraint 2). If we fix the variables f^m_{ij}, **P2** can be decoupled with respect to the variables $R^m_{t,k}$. Following this assumption,

we adopt the primal decomposition approach [23] and solve **P2** by a two-level optimization procedure:

$$\textbf{P2a}: \quad \underset{\mathbf{R}\in\mathbb{R}}{\text{minimize}} \quad \sum_{t\in T}\sum_{m\in M} O(\mathbf{R},\mathbf{f})$$

$$\text{subject to}: \sum_{k\in P(t)}\left(Z_{k,ij}^{tm}\cdot R_{t,k}^{m}\right)\leq f_{ij}^{m},\ \forall(i,j)\in E,\ \forall m\in M,\ \forall t\in T; \tag{4}$$

$$\textbf{P2b}: \quad \underset{\mathbf{R}\in\mathbb{R}}{\text{minimize}} \quad \sum_{t\in T}\sum_{m\in M} \widehat{O}(\mathbf{f})$$

$$\text{subject to}: \sum_{m\in M} f_{ij}^{m}\leq c_{ij},\ \forall(i,j)\in E. \tag{5}$$

Problem **P2a** performs a low-level optimization, which can be further decomposed into a set of sub-problems under the condition that **f** is fixed. Problem **P2b** performs a high-level optimization, which fulfills the update of variable **f**. The optimal value of the objective function of the low-level optimization is locally optimal. It approximates to the global optimality by using the results of the high-level optimization.

4.2 Two-level optimization update

(1) Low-leveloptimization update The Lagrangian dual of Problem **P2a** is defined as:

$$\mathbf{L}(\mathbf{R},\boldsymbol{\lambda})=\sum_{t\in T}\sum_{m\in M} O(\mathbf{R},\mathbf{f})+\sum_{t\in T}\sum_{m\in M}\sum_{(i,j)\in E}\lambda_{ij}^{tm}\left[\sum_{k\in P(t)}(Z_{k,ij}^{tm}\cdot R_{t,k}^{m})-f_{ij}^{m}\right] \tag{6}$$

where λ_{ij}^{tm} is the Lagrange multiplier.

The Lagrange dual function $\mathbf{L}(\mathbf{R},\boldsymbol{\lambda})$ is the maximum value of the Lagrangian over primal variable $\boldsymbol{\lambda}$, and it is given by: $\mathbf{g}(\boldsymbol{\lambda})=\sup_{\mathbf{R}}\mathbf{L}(\mathbf{R},\boldsymbol{\lambda})$.

The Lagrange dual problem is then formulated as: $\underset{\boldsymbol{\lambda}\geq 0}{\text{maximize}}\ \mathbf{g}(\boldsymbol{\lambda})$. Note that **P2a** is equivalent to the above dual problem when the following Karush-Kuhn-Tucker (KTT) conditions [6] are satisfied:

(1) $\left.\frac{\partial\mathbf{L}(\mathbf{R},\hat{\boldsymbol{\lambda}})}{\partial R_{t,k}^{m}}\right|_{R_{t,k}^{m}=\hat{R}_{t,k}^{m}}=0,\ \forall k\in P(t),\ \forall m\in M,\ \forall t\in T;$

(2) $\sum_{(i,j)\in E}\hat{\lambda}_{ij}^{tm}\left[\sum_{k\in P(t)}(Z_{k,ij}^{tm}\cdot\hat{R}_{t,k}^{m})-f_{ij}^{m}\right]=0,\ \forall(i,j)\in E,\ \forall m\in M,\ \forall t\in T;$

(3) $\sum_{k\in P(t)}\left(Z_{k,ij}^{tm}\cdot\hat{R}_{t,k}^{m}\right)-f_{ij}^{m}\leq 0,\ \forall(i,j)\in E,\ \forall m\in M,\ \forall t\in T;$

(4) $\hat{\lambda}_{ij}^{tm}\geq 0,\ \forall(i,j)\in E,\ \forall m\in M,\ \forall t\in T;$

where $\hat{\mathbf{R}}$ and $\hat{\boldsymbol{\lambda}}$ represent the primal and dual optimal point, respectively.

We now propose the following primal-dual algorithm [22] to solve the low-level optimization problem. It updates the primal and the dual variables simultaneously, and moves together towards the optimal points asymptotically.

$$R^m_{t,k}(n+1) = \left[R^m_{t,k}(n) + \dot{R}^m_{t,k} \right]^+ = \left[R^m_{t,k}(n) + \alpha(n) \cdot \frac{\partial \mathbf{L}(\mathbf{R}, \boldsymbol{\lambda})}{\partial R^m_{t,k}} \left(R^m_{t,k}(n) \right) \right]^+ \tag{7}$$

$$\lambda^{tm}_{ij}(n+1) = \left[\lambda^{tm}_{ij}(n) + \dot{\lambda}^{tm}_{ij} \right]^+ = \left[\lambda^{tm}_{ij}(n) - \beta(n) \cdot \frac{\partial \mathbf{L}(\mathbf{R}, \boldsymbol{\lambda})}{\partial \lambda^{tm}_{ij}} \left(\lambda^{tm}_{ij}(n) \right) \right]^+ \tag{8}$$

where n is the iteration index, $\alpha(n)$ and $\beta(n)$ are positive step sizes, and $[z]^+ = \max\{z, 0\}$. The partial derivatives of $\mathbf{R}$ and $\boldsymbol{\lambda}$ are given by:

$$\dot{R}^m_{t,k} = \alpha(R^m_{t,k}) \left[\frac{\partial O(R^m_{t,k}, f^m_{ij})}{\partial R^m_{t,k}} + \sum_{(i,j) \in E} \left(Z^{tm}_{k,ij} \cdot \lambda^{tm}_{ij} \right) \right] \tag{9}$$

$$\dot{\lambda}^{tm}_{ij} = \beta(\lambda^{tm}_{ij}) \left[\sum_{k \in P(t)} \left(Z^{tm}_{k,ij} \cdot R^m_{t,k} \right) - f^m_{ij} \right] \tag{10}$$

Here λ^{tm}_{ij} can be viewed as the congestion price at link (i, j) for the bandwidth requirement of receiver t in layer m. It can be seen from (8) and (10) that if the demand $\sum_{k \in P(t)} (Z^{tm}_{k,ij} \cdot R^m_{t,k})$ at link (i, j) for the information flow exceeds the supply f^m_{ij}, the price λ^{tm}_{ij} will rise, and decrease otherwise. Also, it is notable that all the updating steps are distributed and can be implemented at individual links using only local information.

(2) High-level optimization update As mentioned above, the low-level optimization is operated under the assumption that the value of $\mathbf{f}$ is fixed. In this section, we discuss how to adjust $\mathbf{f}$ to solve the high-level optimization problem.

Suppose $\hat{\lambda}^{tm}_{ij}$ is the optimal Lagrange multiplier corresponding to the constraint in **P2a**. Similar to $\mathbf{R}_t$, we define $\mathbf{f}_{ij} = [f^1_{ij}, \cdots, f^M_{ij}]$ and $\mathbf{f} = [\mathbf{f}_{11}, \cdots, \mathbf{f}_E]^T$. Also let $\mathbb{F}_{ij} = \left\{ \mathbf{f}_{ij} \;\middle|\; f^m_{ij} \geq 0 \text{ for all } m \text{ and } \sum_{m \in M} f^m_{ij} \leq c_{ij} \right\}$, $(i, j) \in E$, and $\mathbb{F}$ denotes the Cartesian product of $\mathbb{F}_{ij}\big((i, j) \in E\big)$. Then the Lagrangian dual and the primal-dual algorithm of **P2b** are proposed as follows:

$$\mathbf{L}'(\mathbf{f}, \boldsymbol{\eta}) = \hat{O}(f) + \sum_{(i,j) \in E} \eta_{ij} \left(\sum_{m \in M} f^m_{ij} - c_{ij} \right) \tag{11}$$

$$f^m_{ij}(n'+1) = \left[f^m_{ij}(n') + \dot{f}^m_{ij} \right]^+ = \left[f^m_{ij}(n') + a(n') \cdot \frac{\partial \mathbf{L}'(\mathbf{f}, \boldsymbol{\eta})}{\partial f^m_{ij}} \left(f^m_{ij}(n') \right) \right]^+ \tag{12}$$

$$\eta_{ij}(n'+1) = \left[\eta_{ij}(n') + \dot{\eta}_{ij} \right]^+ = \left[\eta_{ij}(n') - b\,(n') \cdot \frac{\partial \mathbf{L}'(\mathbf{f}, \boldsymbol{\eta})}{\partial \eta_{ij}} \left(\eta_{ij}(n') \right) \right]^+ \tag{13}$$

where n' denotes the iteration index, and $a(n')$, $b(n')$ are positive step sizes. Through mathematic deduction, the partial derivatives of $\mathbf{f}$ and $\boldsymbol{\eta}$ are given by:

$$\dot{f}_{ij}^{m} = \frac{\partial \mathbf{L}'(\mathbf{f}, \boldsymbol{\eta})}{\partial f_{ij}^{m}}\left(f_{ij}^{m}(n')\right) = a\left(f_{ij}^{m}\right)\left[2 \cdot f_{ij}^{m} + \sum_{t \in T} \lambda_{ij}^{tm} + \eta_{ij}\right] \tag{14}$$

$$\dot{\eta}_{ij} = \frac{\partial \mathbf{L}'(\mathbf{f}, \boldsymbol{\eta})}{\partial \eta_{ij}}\left(\eta_{ij}(n')\right) = b\left(\eta_{ij}\right)\left[\sum_{m \in M} f_{ij}^{m} - c_{ij}\right] \tag{15}$$

Actually, η_{ij} can be regarded as the aggregate congestion price of link (i, j). If the consumed bandwidth f_{ij}^{m} on link (i, j) in layer m cannot meet the actual requirement of all receivers, the f_{ij}^{m} will increase in the next step, or else, it will decrease. Also, the iterations of η_{ij} and f_{ij}^{m} can be implemented in a decentralized manner.

5 Practical implementation of the distributed algorithm

When implementing the proposed distributed algorithm, each link (i, j) and each receiver t is treated as an entity capable of processing, storing and communicating information in a distributed computation system. Assume that the processor for link (i, j) keeps track of variables λ_{ij}^{tm} and f_{ij}^{m}, while the processor for receiver t keeps track of variable $R_{t,k}^{m}$. A decentralized version of the proposed algorithm is summarized in Table 1.

Note that the low-level and high-level algorithms operate at different time scales. The former is an inner loop and operates at a fast time scale, while the latter is an outer loop and performs at a low time scale. More specifically, the high-level

Table 1 Implementation of the proposed distributed algorithm

Initialization

sets $n = 0$, $n' = 0$ and $\lambda_{ij}^{tm}(0)$, $R_{t,k}^{m}(0)$, $f_{ij}^{m}(0)$, $\eta_{ij}(0)$ respectively to some non-negative values for all t, m, (i, j) and k.

Repeat

Updating at link (i,j) in Low-level Implementation:

Receives $R_{t,k}^{m}(n)$ from all receivers $\{t | t \in T$, and $Z_{k,ij}^{tm} = 1\}$.

Updates the congestion price $\lambda_{ij}^{tm}(n)$ according to (8) and (10).

Broadcasts the new price $\lambda_{ij}^{tm}(n+1)$ to all receivers $\{t | t \in T$, and $Z_{k,ij}^{tm} = 1\}$.

Updating at receiver t in Low-level Implementation:

Receives from the network the aggregate congestion price $\sum_{k \in P(t)}\left(Z_{k,ij}^{tm} \cdot R_{t,k}^{m}\right)$.

Updates the rate $R_{t,k}^{m}(n)$ with (7) and (9).

Broadcasts the rate $R_{t,k}^{m}(n+1)$ to all links $\{(i, j) | (i, j) \in E$, and $Z_{k,ij}^{tm} = 1\}$.

Updating at link (i,j) in High-level Implementation:

Calculates the sum $\hat{\lambda}_{ij}^{tm}\left(f_{ij}^{m}(n')\right) = \sum_{t \in T} \hat{\lambda}_{ij}^{tm}\left(f_{ij}^{m}(n')\right)$.

Updates a new $f_{ij}^{m}(n')$ with (12) and (14).

Updates the aggregate congestion price according to (13) and (15).

Broadcast the new $f_{ij}^{m}(n'+1)$ to all receivers $\{t | t \in T$, and $Z_{k,ij}^{tm} = 1\}$.

Until

All variables converge to the optimums.

algorithm will not move to its step until $\hat{\lambda}$ at the low-level converges to its optimum value. When the algorithm converges, the generated solution will jointly optimize the rate allocation and the transmission structure.

When the communication overhead issue [15] is taken into account, all the update operations at both low-level and high-level iterations can utilize those variables stored in the local node or link, except the information of the updated rate $R^m_{t,k}(n+1)$, $f^m_{ij}(n'+1)$ and the updated price $\lambda^{tm}_{ij}(n+1)$ that are needed to be transmitted by extra packets. For example, according to (8), to update the Lagrange price $\lambda^{tm}_{ij}(n)$, the Rate Packet (RP) of receiver t carrying the rate information of $R^m_{t,k}(n)$ is only required to transmit upward along t's paths to the subset of links $\{(i,j)|(i,j) \in E$, and $Z^{tm}_{k,ij} = 1\}$. Similarly, on the basis of (7), to update the rate $R^m_{t,k}(n)$, the Control Packet (CP) containing link (i,j)'s Lagrange price $\lambda^{tm}_{ij}(n)$ is only to be sent downward to the subset of receivers $\{t|t \in T$, and $Z^{tm}_{k,ij} = 1\}$ along the paths that link belongs to. If we adopt the float type in implementation, each rate or Lagrange price takes up only 4 bytes, thus is negligible compared to the main video streaming traffic. Roughly estimated, the time spent by the whole network to reach the stability is equal to the number of iterations required for convergence multiplying the update time interval of each iteration. It is found in [9] that an update interval which is about 2 to 3 times the one way propagation delay of the particular receiver is sufficient. Therefore, the entire overhead of the proposed distributed algorithm is quite small.

6 Results and discussion

In this section, we present numerical and simulation results to show the performance of the proposed algorithm. We conduct numerical experiments on classical butterfly network topology which has been extensively used in network coding-based simulation studies [1, 19, 28]. The purpose of numerical solution is to evaluate the convergence behavior of the proposed distributed algorithm. Also, we present simulation results for a packet-level simulation with a general network topology, and show that our algorithm achieves an overall balanced throughput and better video quality over all receivers.

6.1 Numerical simulation results

The classical butterfly network topology, shown in Fig. 3a, consists of source S, relay nodes R_i, and receivers T_i. The capacity and random propagation delay (between 0 and 1) of each link are marked as capacity/delay on each link. Assume the source generates an SVC stream into three layers, with rate of 2.5 (data units/second) on both the base layer and the first enhancement layer, and a rate of 1 on the second enhancement layer.

Convergence behavior Figure 4 shows the assigned data rate for each receiver at each layer during the low-level optimization, where we adopt constant step sizes with $\alpha(n) = 0.0631$, $\beta(n) = 0.01733$, $a(n) = 0.51$ and $b(n) = 0.0155$. It can be seen that all data rates converge after 100 iterations. For instance, the total rates achieved by T_1 reach within 10 % of its optimal value after 37 iterations and converge to 5.00038

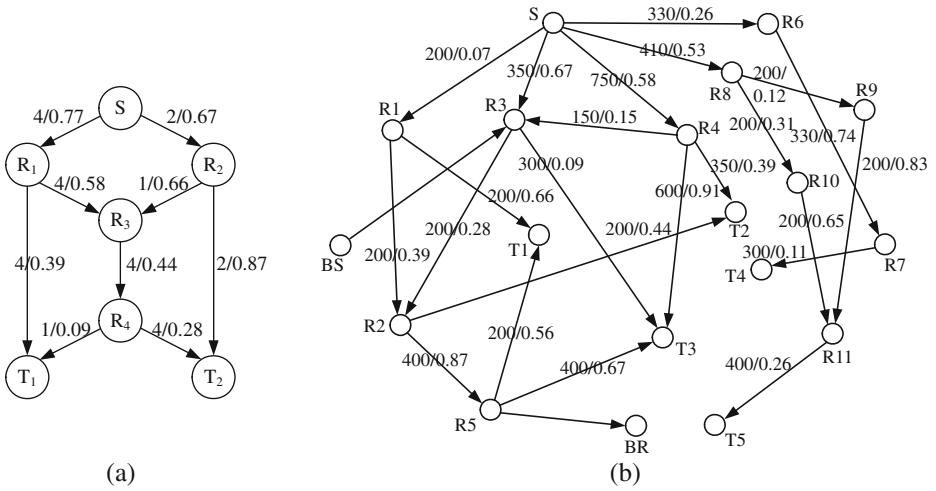


Fig. 3 Network topology associated with link capacity/delay, where **a** is a butterfly topology for numerical experiment, **b** is a general network topology for SVC streaming based simulation

after 80 iterations. The rates achieved by T_2 of two layers reach within 10 % of its optimum after 46 iterations and converge to 4.995936 after 92 iterations.

Figure 5 shows the convergence behavior of the high-level optimization. Due to space limit, we only show the rate evolutions of links (R_1, R_3), (R_2, R_3) and (R_3, R_4) at the first enhancement layer, while other links have similar outcomes. It is observed that the flow rates on these three links converge after 250 iterations. In addition, due to the implementation of network coding on link (R_3, R_4), the sum of the flow rates on links (R_1, R_3) and (R_2, R_3) is almost equal to the flow rate on link (R_3, R_4).

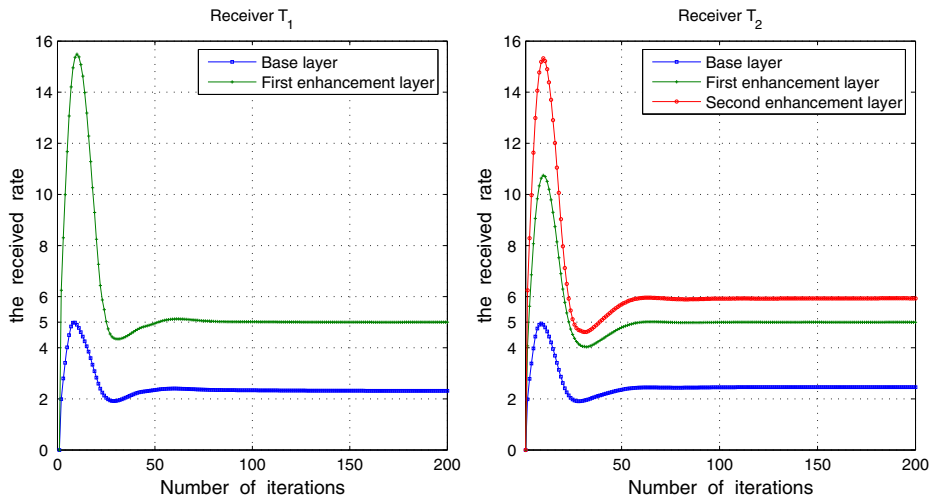


Fig. 4 Evolution of the assigned rate for each receiver in the low-level optimization

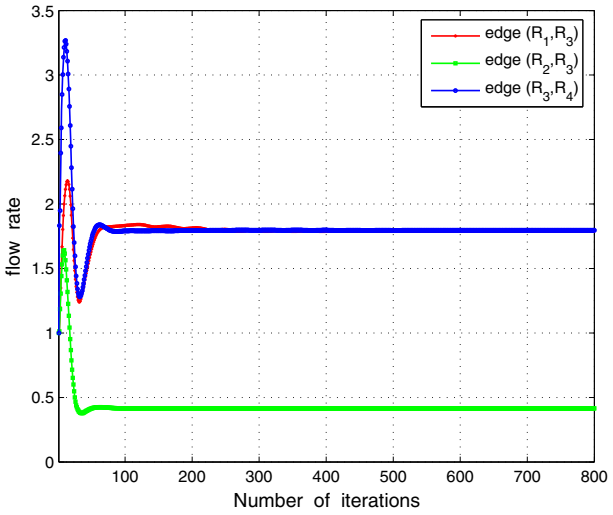


Fig. 5 The performance of the high-level optimization

Impact of step size The Lagrange multipliers $\boldsymbol{\lambda}$ and $\boldsymbol{\eta}$ signal the congestion status of the entire network. By iteratively modifying their values, the distributed algorithm gradually reaches an optimal rate allocation solution. We now investigate the impact of the step sizes $\boldsymbol{\lambda}$ and $\boldsymbol{\eta}$ on the convergence speed. In contrast to the aforementioned experiment, we adjust $\beta(n)$ to 0.018, and $b(n)$ to 0.025, with $\alpha(n)$ and $a(n)$ unchangeable. As seen in Fig. 6, in this case, the receiving rate of T_2 does not converge to the optimal point. Instead, it converges to some suboptimal solution within a quite small neighborhood around the optimum. Since such phenomenon is likely to happen when the constant step size is used [5], the diminishing step size becomes a better alternative.

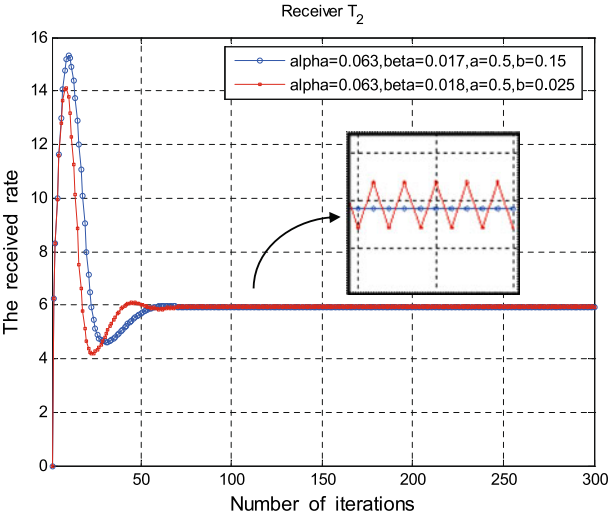


Fig. 6 Impact of different fixed step sizes on convergence behavior

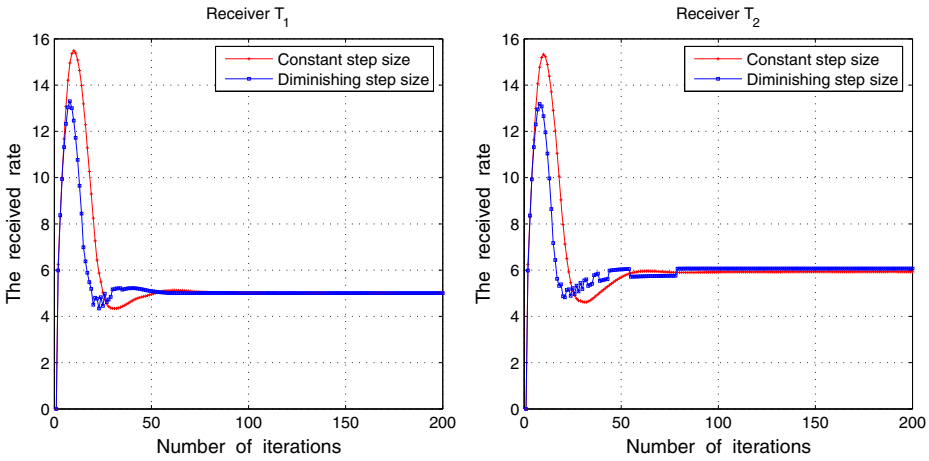


Fig. 7 Performance comparison for constant and diminishing step size

Here we let $\beta(t) = \frac{1}{t}$, satisfying $\lim_{t\to\infty} \beta(t) = 0$ and $\sum_{t=0}^{\infty} \beta(t) = \infty$. Compared with a constant step size, we can find in Fig. 7, that the receiving rates with a diminishing step size vary smoother but converge more slowly than its fixed counterpart. Although a fixed step size is more convenient for distributed implementation, a diminishing step size is recommended in practice, for the rate with low and smooth fluctuation is crucial for video quality smoothness.

Throughput performance Figure 8 compares the achievable throughput of two receivers by the shortest path(SP) distribution tree, the LION algorithm and the proposed algorithm. It is seen that the proposed algorithm outperforms both the shortest path and LION algorithms. As the shortest path scheme constructs video distribution tree with single path and does not use network coding, in contrast, LION and our method have introduced network coding based multipath routing and achieved significant gains in network throughput. Furthermore, for receiver T_1, both multipath algorithms can realize its max-flow capacity of 5, while for receiver T_2, only

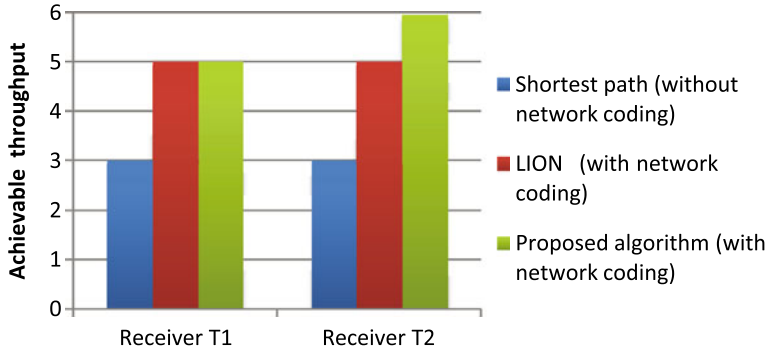


Fig. 8 Comparison of achievable throughput

our distributed algorithm can achieve a rate of 5.942 and approximate to its max-flow capacity 6.

6.2 Packet-level simulation results

To evaluate the received video quality using the proposed distributed algorithm, we also conduct packet-level simulations with a general network topology, as shown in Fig. 3b. It contains a source S, 11 relay nodes $R_1 \sim R_{11}$ and 5 receivers $T_1 \sim T_5$. The capacity (Kbps)/propagation delay (per Kbit) is marked on each link. The numbers of alternative paths for 5 receivers are 4, 4, 6, 1, 2, and their max-flow rates are 400, 550, 1300, 300, 400 Kbps, respectively. The configuration of parameters are shown in Table 2.

In the packet level simulations, we use the practical random network coding [8] to distribute the source packets of each layer. Here we assume intra-session network coding is implemented within each layer to ensure easy operation. During data transmission, each relay node (as well as the source node) combines its received packets belonging to the same generation from different upstream links (or video source packets encoded by the source node) with random linear operations over a large Galois Field and then sends the coded packets to its downstream links. Each destination node can correctly decode the original packets if it receives enough coded packets. To cope with asynchronous transmission, we use the buffer model [8] to synchronize the packet arrivals and departures. In the buffer model, packets that arrive at a node on any of the incoming links are put into a single buffer sorted by layer. Then, whenever there is a transmission opportunity at an outgoing link, the number of packets of every layer in the buffer is checked and a packet is generated containing a random linear combination of all the packets that belong to the layer with the largest number of packets. After the generated packet is transmitted to the outgoing link, certain old packets are flushed from the buffer according to the flushing policy. Specially, if two layers have the same number of packets in the buffer, the lower layer is prioritized to generate a packet for transmission.

We use four standard test sequences “Bus”, “Coastguard”, “Foreman” and “Mobile” with a frame rate of 30 fps, CIF (352×288) resolution, and a GOP-length of 32 frames with IBBP... structure. The streams are generated using the Joint Scalable Video Model 9_10 reference codec of H.264/AVC scalable extension, with 256 Kbps on the base layer and 384 Kps, 512 Kps and 1024 Kps on the enhancement layers by fine granularity scalability (FGS) encoding. Figure 9 shows the rate-distortion performance, measured in average peak signal-to-noise ratio (PSNR), for the four CIF video sequences.

Table 2 Configuration of parameters

Parameter description	Value
Step size $\alpha(n)$	0.05
Step size $a(n)$	0.05
Step size $\beta(n)$	0.01
Step size $b(n)$	0.01
Galois field size of network coding	8
Generation size of network coding	50
Number of iterations	400
Update interval	5.4 ms

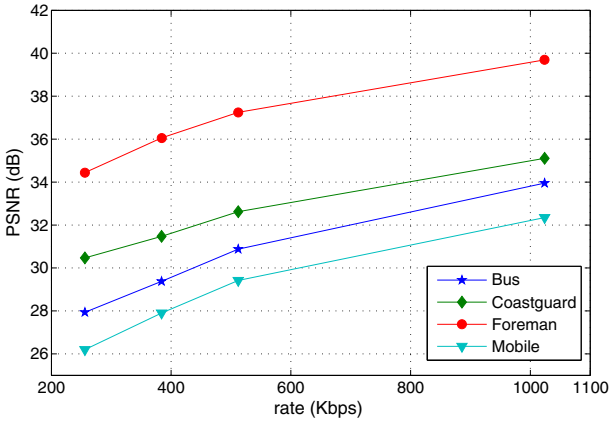


Fig. 9 PSNR performance achieved for four CIF sequences with frame rate of 30 fps and GOP length of 32

Throughput and transmission cost comparison Table 3 presents the number of layers received at each receiver with different algorithms. The transmission cost for each receiver in the base layer is shown in Fig. 10, which is the sum of each path's cost calculated by (2) for the base layer distribution. It can be seen that the shortest path scheme not only achieves the lowest throughput, but brings the undesired cost for the base layer transmission. The LION algorithm builds the distribution meshes with a heuristic scheme and achieves a suboptimal throughput. Similar to the shortest path algorithm, the LION also does not consider the layer synchronization of SVC streams. Therefore, these two algorithms are not efficient for practical SVC multirate multicast. Conversely, the proposed algorithm makes a joint optimization on the throughput and the transmission cost. As a result, it achieves the best throughput performance over all receivers, meanwhile, maintains the smallest cost for the base layer transmission. In addition, if network coding is not used in our algorithm, the overall throughput will distinctly decrease for the prohibition of the bandwidth share at the same layer.

Relationship between cost and delay According to the definition of the cost function in (2), the path cost can be used to depict the end-to-end path delay. To verify their linear relationship, we vary the playback deadline for "Bus", "Coastguard", "Foreman" and "Mobile" streams from 400 ms to 500 ms. Note that we only consider broadcasting stored video, and 'live' videos are beyond the scope of this paper. Here, we suppose that packets are dropped if they do not arrive at the

Table 3 Number of layers received at each receiver

	T_1	T_2	T_3	T_4	T_5	Total
Shortest path tree (without network coding)	2	2	3	1	2	10
LION (with network coding)	2	3	3	1	2	11
Proposed algorithm (with network coding)	2	3	4	1	2	12
Proposed algorithm (without network coding)	2	2	3	1	2	10

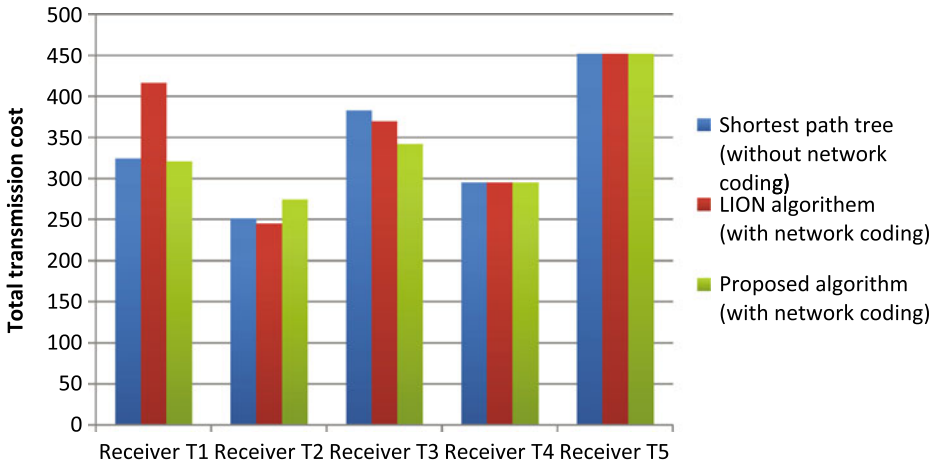


Fig. 10 The transmission cost of the base layer for each receiver

receiver by the playback deadline. In Table 4, we show the average video quality (in PSNR) at receiver T_1 as an example. Clearly, the proposed algorithm achieves better video quality. Note that in the shortest path or the LION scheme, the base layer packets for T_1 are dropped when the playback deadline is small, i.e. 400 ms. Although T_1 can receive higher layer packets at lower cost, it still cannot decode any video information. As the playback deadline increases, larger packet delays can be tolerated. When the playback deadline increases to 500 ms, the video quality of the shortest path and LION schemes is similar to that of the proposed algorithm.

Influence of continuous achievable rate region Figure 11 shows the average video quality measured as PSNR for "Mobile" stream at T_2 and T_3, where the aggregate rate allocated over the network, i.e., the total rate allocated on the output links of source node S, varies from 200 Kbps to 1.3 Mbps. Within the context of SVC, the achievable set of layer bandwidths could be continuous with FGS. Different receivers are able to receive data at different rates by join different multicast groups and video streaming layers. The LION algorithm adopts a discrete layer rate control, where a receiver should receive either a layer in whole or nothing, even if there remains

Table 4 Received average video quality of T_1 measured as PSNR for four sequences

	Playback deadline = 400 ms				Playback deadline = 500 ms			
	Bus	Coastguard	Foreman	Mobile	Bus	Coastguard	Foreman	Mobile
Shortest path	0	0	0	0	29.38	31.47	36.06	27.9
LION	0	0	0	0	29.38	31.47	36.06	27.9
Proposed algorithm	27.94	30.47	34.44	26.2	29.38	31.47	36.06	27.9

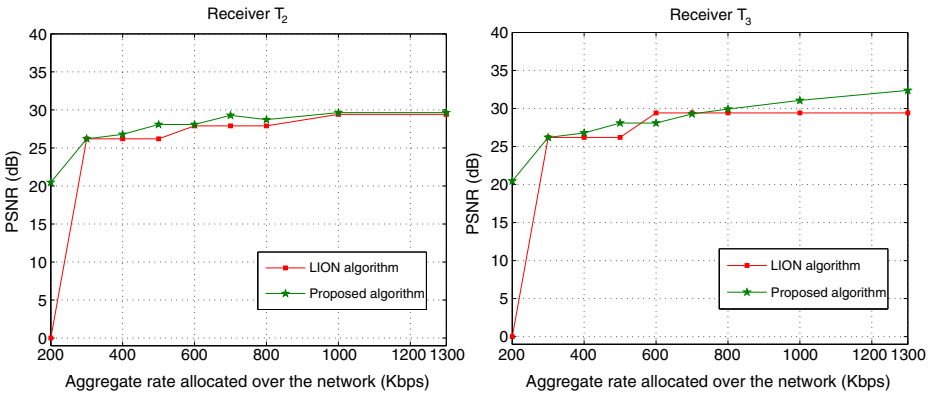


Fig. 11 Average video quality measured as PSNR for "Mobile" stream at T_2 and T_3

a lot of available bandwidth. On the contrary, the proposed algorithm supports the continuous achievable rate region and a partial subscription of the achieved highest layer. Consequently, the network resource can be fully utilized, thus achieving a better received video quality.

Influence of background traffic To simulate real environments of network traffic, we generate background traffic between nodes R_3 and R_5 by superposing 100 ON/OFF sources of pseudo nodes which have Pareto distributions. As shown in Fig. 3b, node BS serves as the background traffic source and node BR serves as the background traffic receiver, then a background traffic is constructed along $BS \rightarrow R_3 \rightarrow R_2 \rightarrow R_5 \rightarrow BR$. Since this path mainly overlaps with the paths to T_1

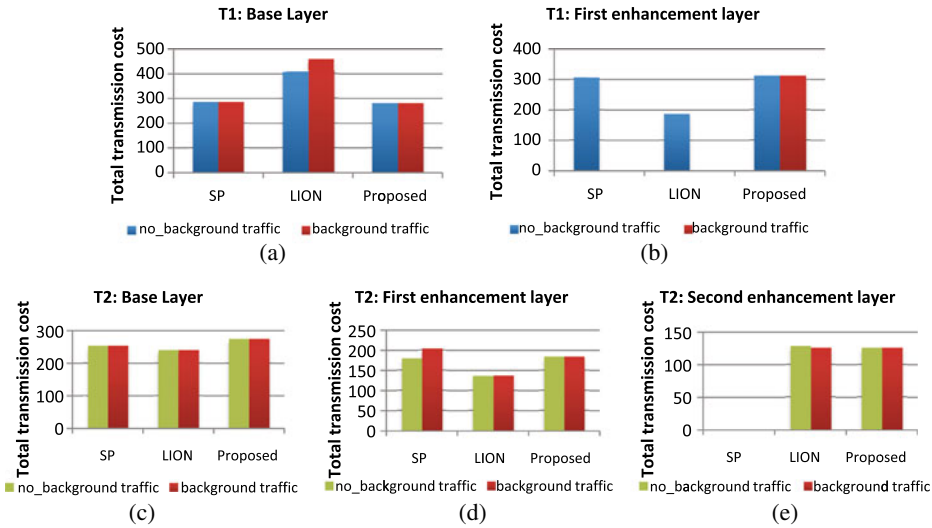


Fig. 12 Transmission cost of T_1 and T_2 when background traffic is generated in network

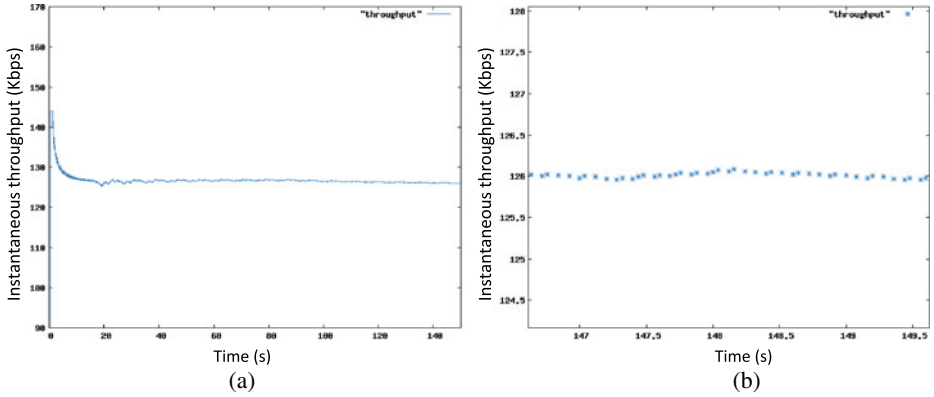


Fig. 13 Variation of throughput over path $S_1 \rightarrow R_4 \rightarrow R_3 \rightarrow R_2 \rightarrow R_5 \rightarrow T_3$

and T_2, we focus on the comparison of transmission cost and throughput of these two receivers before and after the background traffic is imposed.

As seen in Fig. 12, when the background traffic is plunged into the network, only by using the proposed algorithm, T_1 can receive two layers while T_2 can receive three layers. Moreover, the average transmission cost of the proposed algorithm for two receivers are clearly lower than other two algorithms. For the shortest path scheme, its transmission for the base layer consumes too much bandwidth, resulting in insufficient bandwidth remained for the higher layers. For the LION algorithm, it neglects the cost factor and chooses the path $R_3 \rightarrow R_2 \rightarrow R_5$ for delivering the base layer to T_1, which mostly overlaps with the path for background traffic. Consequently, a high bandwidth consumption occurs on path $R_3 \rightarrow R_2 \rightarrow R_5$ that leaves deficient bandwidth for T_1 to join the first enhancement layer.

Variation of throughput Figure 13 plots the variation of instantaneous throughput for the "Coastguard" sequence along path $S_1 \rightarrow R_4 \rightarrow R_3 \rightarrow R_2 \rightarrow R_5 \rightarrow T_3$ on the first enhancement layer when the proposed algorithm is applied. In this case, the background traffic is constructed over the path $BS \rightarrow R_3 \rightarrow R_2 \rightarrow R_5 \rightarrow BR$. As can be found in Fig. 13a, influenced by background traffic, the path throughput drops gradually from 145 Kbps and balances around the value of 126 Kbps. The rate update interval for each path in this experiment is set to 0.05 s that is comparable to the end-to-end path propagation delay [9]. It can be seen from Fig. 13b that, even within small intervals, the throughput varies smoothly around the optimum value.

7 Conclusions

In this paper, we study the prioritized optimization problem of rate-distortion control and network traffic for scalable video multicast. By coupling network coding and multipath routing with multi-rate control, we propose a convex mathematical model for constructing video distribution meshes with maximum throughput and minimum distortion. It seeks optimal paths and associated rates with a minimum bandwidth consumption scheme for each video layer, while considering the content priority of

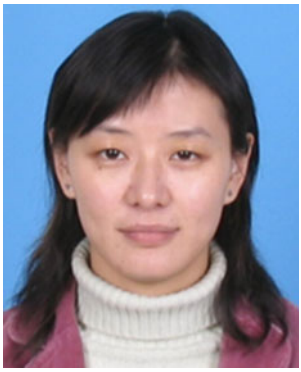
the base video layer with minimal delay. The flow control problem is formulated into a minimization programming in which the quality variation between layers, the transmission cost of the base layer, as well as an efficient resource utilization encouraging path-overlapping transmissions and allowing bandwidth sharing among different receivers for the same video layer by utilizing network coding are jointly considered. We solve the target convex optimization problem by a fully decentralized algorithm through decomposition and dual approach, and the convergence behavior and benefits of the proposed algorithm are demonstrated in extensive experiments.

In this work, we assume that the video streams are distributed through a static heterogeneous network. We propose to extend this scenario to practical peer-to-peer cases, where the dynamics of the network (e.g. peer joining and departure) should be considered on the basis of the application-layer multicast protocol. Also, with the rapid development of wireless communication techniques, we believe that scalable video streaming over various wireless networks is very important in future. We hope to study resource optimization in wireless networks by jointly designing channel competition and rate allocation in the next step.

References

1. Ahlswede R, Cai N, Li SY, Yeung RW (2000) Network information flow. IEEE Trans Inf Theory 46(4):1204–1216
2. Amon P, Rathgen T, Singer YD (2007) File format for scalable video coding. IEEE Trans Circuits Syst Video Technol 17(9):1174–1185
3. Banerjee S, Bhattacharjee B, Kommareddy C (2002) Scalable application layer multicast. In: Proc. ACM SIGCOMM. Pittsburgh, USA
4. Bertsekas DP, Tsitsiklis JN (1989) Parallel and distributed computation: numerical methods. Prentice-Hall, New Jersey
5. Bertsekas DP, Nedic A, Ozdaglar AE (2003) Convex analysis and optimization. Belmont, MA: Athena Scientific
6. Boyd S, Vandenberghe L (2004) Convex optimization. Cambridge, U.K., Cambridge Univ. Press
7. Chen L, Ho T, Low SH et al (2007) Optimization based rate control for multicast with network coding. In: Proceeding of IEEE Infocom
8. Chou PA, Wu Y, Jain K (2003) Practical network coding. In: Proc. of Allerton conference on communication, control and computing
9. Deb S, Srikant R (2004) Congestion control for fair resource allocation in networks with multicast flows. IEEE/ACM Trans Netw 12(2):274–285
10. Fazel M, Chiang M (2005) Network utility maximization with nonconcave utilities using sum-of-squares method. In: Proc. of conference on decision and control
11. Fu F, Stoenescu TM, van der Schaar M (2007) A pricing mechanism for resource allocation in wireless multimedia applications. IEEE J Sel Top Signal Process 1(2):264–279
12. Han H, Shakkottai S, Hollot CV, Srikant R, Towsley D (2006) Multipath TCP: a joint congestion control and routing scheme to exploit path diversity in the internet. IEEE/ACM Trans Netw 14(6):1260–1271
13. Jeong J, Jeon S, Jung YH, Choe Y (2009) Statistical multiplexing using scalable video coding for layered multicast. In: Proceedings of IEEE international symposium on broadband multimedia systems and broadcasting. Bilbao, Spain
14. Kar K, Sarkar S, Tassiulas L (2001) Optimization based rate control for multirate multicast sessions. In: Proc. IEEE INFOCOM
15. Kar K, Sarkar S, Tassiulas L (2002) A scalable low-overhead rate control algorithm for multirate multicast sessions. IEEE J Sel Areas Commun 20(8):1541–1557
16. Kleinrock L (1976) Queuing systems, vol II: omputer applications. Wiley Interscience, New York
17. Lai W, Pan C (2002) Achieving inter-session fairness for layered video multicast. IEEE Trans Broadcast 48(3):215–222

18. Li B, Liu J (2003) Multirate video multicast over the internet: an overview. IEEE Netw 17(1):24–29
19. Li SY, Yeung RW, Cai N (2003) Linear network coding. IEEE Trans Info Theory 49(2): 371–381
20. Low S, Lapsley DE (1999) Optimization flow control, I: basic algorithm and convergence. IEEE/ACM Trans Netw 7:861–874
21. Lun DS, Mdard M, Karger DR (2005) On the dynamic multicast problem for coded networks. In: Proc. of WINMEE, RAWNET and NETCOD 2005 workshops
22. Lun DS, Ratnakar N, Mdard M et al (2006) Minimum-cost multicast over coded packet networks. IEEE Trans Inf Theory 52(6):2608–2623
23. Palomar D, Chiang M (2006) A tutorial on decomposition methods for network utility maximization. IEEE J Sel Areas Commun 24(8):1439–1451
24. Sarkar S, Tassiulas L (2005) Fair distributed congestion control for multirate multicast networks. IEEE/ACM Trans Netw 13(1):121–133
25. Schwarz H, Marpe D, Wiegand T (2007) Overview of the scalable video coding extension of the H.264/AVC standard. IEEE Trans Circuits Syst Video Technol 17(9):1103–1120
26. Stuhlmuller K, Farber N, Link M, Girod B (2000) Analysis of video transmission over lossy channels. IEEE J Sel Areas Commun 18(6):1012–32
27. van der Schaar M, Andreopoulos Y, Hu Z (2006) Optimized scalable video streaming over IEEE 802.11 a/e HCCA wireless networks under delay constraints. IEEE Trans Mob Comput 5(6):755–768
28. Wu Y (2006) Network coding for multicasting. PhD Thesis, Princeton University
29. Zhao J, Yang F, Zhang Q et al (2006) LION: layered overlay multicast with network coding. IEEE Trans Multimed 8(5):1021–1032
30. Zhu X, Schierl T, Wiegand T, Girod B (2008) Video multicast over wireless mesh networks with scalable video coding (SVC). In: Proc. visual communication and image processing, VCIP 2008, San Jose, CA
31. Zou J, Xiong H, Li C et al (2011) Prioritized flow optimization with multi-path and network coding based routing for scalable multirate multicasting. IEEE Trans Circuits Syst Video Technol 21(3):259–273



Junni Zou received the M.S. degree and the Ph.D. degree in communication and information system from Shanghai University, Shanghai, China, in 2004 and 2006, respectively. Since then, she has been with the School of Communication and Information Engineering, Shanghai University, Shanghai, where she is an Associate Professor. From June 2011 to June 2012, she was with the Department of Electrical and Computer Engineering, University of California, San Diego (UCSD), as a visiting Professor.

Her research interests include distributed resource allocation, multimedia networking and communications, and network information theory. She has published over 50 international journal/conference papers on these topics. Dr. Zou is the recipient of Shanghai Young Rising-Star Scientist in 2011. Also, Dr. Zou acts as member of Technical Committee on Signal Processing of Shanghai Institute of Electronics.

Lu Jiang received the B.S. and M.S. degree in Communication Engineering from Shanghai University, Shanghai, China, in 2007 and 2010, respectively. Her main research interests include scalable video streaming, resource allocation and network coding.

Chenglin Li received the B.S. and M.S. degree in electronic engineering from Shanghai Jiao Tong University, Shanghai, China, in 2007 and 2009, respectively. He is currently working toward the Ph.D. degree at Shanghai Jiao Tong University. His main research interests include network oriented image/video processing and communication, and the network based optimization for video sources.