Journal of Network and Computer Applications 157 (2020) 102574

Contents lists available at ScienceDirect

Journal of Network and Computer Applications

journal homepage: www.elsevier.com/locate/jnca

Decentralized asynchronous optimization for dynamic adaptive multimedia streaming over information centric networking

Mu Wang [a], Changqiao Xu [a,*], Xingyan Chen [a], Lujie Zhong [b], Gabriel-Miro Muntean [c]

[a] State Key Laboratory of Networking and Switching Technology, Beijing University of Posts and Telecommunications, Beijing, 100876, PR China
[b] Information Engineering College, Capital Normal University, Beijing, 100048, PR China
[c] Performance Engineering Laboratory, School of Electronic Engineering, Dublin City University, Dublin, Ireland

ARTICLE INFO

Keywords:
Information centric networking (ICN)
Dynamic adaptive video streaming(DAS)
Distributed concave optimization
Stochastic optimization

ABSTRACT

By the envision of combing smooth viewing experience with high-efficiency content distribution, dynamic adaptive streaming (DAS) over information-centric networking (ICN) is becoming a promising trend for the future video services. However, optimizations of DAS flow transmission control and rate adaptation need to be revisited for better adopting the ICN with multicast, multi-rate forwarding and decentralized framework. In this paper, we propose a decentralized asynchronous method for ICN-DAS. We first formulate the problem as a two-stage optimization, wherein the first stage's objective is to optimize the transmission rate within network capacity constraints, and the second is adapting the video bitrate for the long-term viewing utility. A distributed asynchronous optimization algorithm (DAOA) is then proposed for solving the two-stage problem iteratively by a novel distributed switching mirror descent and virtual queue-based iterations. Analytic results including convergence, computation complexity and time-varying adaptation are provided to validate theoretically the DAOA's performance. Simulation-based testing has also been conducted for evaluating DAOA's performance and assess its viewing experience, in comparison with state-of-the-art solutions.

1. Introduction

Following the proliferation of smart devices and increase in rich media content demand (Xu et al., 2015a; Costa etal., 2019, Cao et al., 2019), it is foreseen that diverse video-based applications will be responsible for the traffic which will dominate the future Internet. According to Cisco, over 80% of the traffic in the future Internet will be generated by video services after 2020 (Costa et al.). This traffic increase is caused by both growing popularity of video applications and increasingly high bandwidth requirements of the latest evolving video (i.e., virtual/augmented reality (Guna et al., 2019; Rashid et al., 2017)) and challenges significantly the existing network capacity. Apart from the effects of network capacity shortage, video clients have also affected their smooth content playback by the mismatch between the dynamic fluctuation of the available bandwidth and constant video encoding bitrate.

Instead of patching the current TCP/IP paradigm (Xu et al., 2015b; Tang et al., 2019) when trying to address the network capacity issue, the emerging information-centric networking (ICN) employs a different approach (Xu et al., 2017; Abdullahi et al., 2015). ICN improves the network resource (re)usage by shifting the network design concern from host to content and network operation focus from host management to content distribution. In this context, ICN enables name-based content delivery (i.e., based on sending of *Interest* packets (Zhang et al., 2014)) and in-network caching (Xu et al., 2018), which inherently create opportunities for nearby data fetching and multicast-oriented delivery (Stais et al., 2015), thereby improving network capacity.

Recently, the dynamic adaptive video streaming (DAS) (El Essaili et al., 2015; Rainer et al., 2017), designed for heterogeneous devices and networks, is widely used. DAS relies on different video versions (termed *representations*) and on multiplexing the various video encoding bitrates. By employing dynamic selection of video representations in a process of flexible adaptation of the video bitrate according to network bandwidth variation, DAS makes possible smooth remote video playback. In order to support provision of improved performance of video services, a natural emerging trend was to deploy DAS in ICN, which has gained

* Corresponding author.
E-mail addresses: wangmu@bupt.edu.cn (M. Wang), cqxu@bupt.edu.cn (C. Xu), chenxingyan@bupt.edu.cn (X. Chen), zhonglj@cnu.edu.cn (L. Zhong), gabriel.muntean@dcu.ie (G.-M. Muntean).

https://doi.org/10.1016/j.jnca.2020.102574
Received 12 April 2019; Received in revised form 4 January 2020; Accepted 8 February 2020
Available online 13 February 2020

important attention (Rainer et al., 2016; Samain et al., 2017). In a ICN DAS solution, crucial for resulting service quality is how to fully utilize the network capacity and optimally request the appropriate video representations. Existing research avenues, including (Lederer et al., 2013; Jmal et al., 2017), follow an end-to-end rate adaptation design principle to control the ICN DAS delivery. However, this approach does not achieve best performance mostly due to neglecting ICN features, such as inherent multicast and multi-rate delivery provided by in-network caching and *Interest* packet aggregation.

The operation of ICN DAS has two main phases:

1) ICN clients firstly determine how fast to send out the *Interest* packets (termed as the sending rate of requests) according to the network capacity. This phase can be considered as a flow control problem which aims to distributively maximize the overall transmission rate within the bandwidth limits. Due to the multicast, multi-rate feature of ICN DAS, one provider may simultaneously serve multiple clients with different requesting representations. Existing solutions (Karami, 2015; Liu and Wei, 2016) which consider the unicast scenarios only would benefit if they accommodate this salient feature. In addition, using a Lagrangian method for solving this problem as in solution (Carofiglio et al., 2016) also becomes difficult due to the non-differential aspect of the problem. Instead, a lightweight flow control algorithm to avail from for multicast and multi-rate features of ICN DAS should be considered.
2) In the second phase, the clients determine the encoding rate of the requested video according to the *Interest* packet sending rate. Different from current studies which formulate encoding bitrate adaptation as a deterministic optimization problem (Rainer et al., 2016), here the requested video bitrate varies frequently, mostly due to the dynamic ICN characteristics. In this phase, the focus should be on the algorithm for selection of the video representations that optimizes client utility.

Addressing the above-mentioned challenges, this paper introduces and describes DAOA, a distributed asynchronous optimization algorithm for ICN DAS. DAOA includes a lightweight decentralized flow control algorithm which optimizes the *Interest* sending rate of each client. DAOA also includes an algorithm for video representation selection that maximizes the long term client utility. The major contributions of this paper are summarized as follows:

(1) ICN DAS delivery is formulated as a two stage optimization problem. In the first stage, a generic flow control problem considering the multi-cast and multi-rate features of ICN DAS is formulated. The problem optimizes the transmission rate for a given network capacity in each time slot. In the second stage, the video representation selection problem is formulated as a stochastic optimization problem, focusing on long term utility optimization according to the feasible transmission rate.
(2) A distributed asynchronous optimization algorithm (DAOA) for ICN DAS is proposed to solve the two stage optimization problem. As solution to the flow control problem, a Distributed Switching Mirror Descent Algorithm (DSMDA) is introduced. DSMDA enables each client self-determine the transmission rate by negotiating with the on-path links. As solution to the second stage issue, the Virtual Queue-based Iteration Algorithm (VQIA) is proposed to determine the video representations to be requested in each time slot so individual client utility are optimized.
(3) In order to assess the performance of the proposed algorithm from both theoretical and practical points of view, comprehensive analysis on algorithms' convergence, computation complexity and time varying adaptation characteristics was performed. Additionally, DAOA was tested via simulations with three different topologies. Results verify the fast convergence and time varying adaptation, and also show how DAOA outperforms another state-of-art solution in terms of throughput, playback stalling and quality.

The rest of the paper is organized as follows: section 2 surveys related works and section 3 describes the scenarios of focus. Section 4 provides the system model and problem formulation. Sections 5 and 6 include the detail design of the proposed DAOA for ICN DAS and present the main theoretical results. Sections 7 and 8 present the simulation-based evaluations and draw conclusions.

2. Related work

2.1. ICN flow control

In order to maximize communication resource utilization and overall throughput while avoiding network congestion, several solutions for ICN flow control have been proposed. Among the existing attempts, Zhang et al. (2015) proposed an explicit congestion control mechanism for content-centric networks named Chunk-switched Hop Pull Control Protocol (CHoPCoP). The main idea of CHoPCoP is that routers schedule sending the *Interest* requests according to one-hop congestion estimations, and end users adjust their sending rate based on the conventional AIMD method. However, users adjust the sending rate according to the marking packets of upstream routers instead of the congestion information of the whole delivery path, which may result in ineffective congestion control. An integrated method to improve the transmission control is proposed by Li et al. (2017). This method employs both a flow-aware congestion estimation scheme to predict congestion according to historical information and a mechanism that dynamically sets the eviction time of PIT entries according to round trip time (RTT). However, both the congestion prediction and RTT estimation may become inaccurate in ICN scenarios due to the high dynamics of data flows (such as flash crowd) and on-path caching.

Carofiglio *et al.* proposed a multipath control method (Carofiglio et al., 2013) where RTT and window-based control info are processed at each router in order to improve the accuracy of rate control. To further smoothen sending window variation, a window decrease method named remote adaptive active queue management (RAAQAM) has been proposed. The transmission dynamic under RAAQAM has also been modeled as a fluid-based model and the stability of RAAQAM has also been proved. However, whether this method can achieve or not optimum bandwidth utilization is not discussed. Besides, the proposed fluid-based model does not consider the *Interest* aggregation feature of ICN, whose performances may be impaired when delivering content with multicast.

Karami (2015) proposed ACCPndn, a machine learning-based control method for NDN. Specifically, ACCPndn uses a neural network architecture to forecast the network congestion and degree of congestion. A heuristic congestion avoidance method is then proposed, which leverages a fuzzy inference system to estimate the interface load and adjust the sending rate according to the load. However, the accuracy of forecasting by employing neural networks highly relies on the training set and the dynamics of traffic pattern may result in inaccurate rate forecasting, which in turn influences the control efficiency. ACCPndn requires implementation of a central controller for congestion forecasting, which has scalability issues.

Carofiglio et al. (2016) formulated an ICN flow control problem with two objectives: throughput maximization and network cost minimization. The problem is furthered divided into two sub-problems focused on rate control and *Interest* forwarding, which are solved separately. However, the problem formulated in Carofiglio et al. (2016) treats each flow individually and does not consider the multicast feature of ICN, which lead to low bandwidth utilization. Besides, the computation and communication overhead brought by Lagrangian calculation may also reduce the algorithm performance.

2.2. Representation adaptation

In DAS, for example, MPEG-DASH based on scalable video coding, the video content is encoded into a base layer and several enhancement layers. Video stream can be decoded from the base layer or base layer plus a single or multiple enhancement layers. The more enhancement layers involved, the higher the video quality obtained is, but also the higher consumption of bandwidth resources. Hence, the challenge is to select suitable video representations that ensure smooth video playback at high quality of experience level in given network conditions. In this area, most studies concentrate on caching and forwarding design for ICN DAS, whereas limited number of works focus on optimally selecting video representations. In Lederer et al. (2014), a DASH-enabled content-centric network (CCN) architecture was designed. The associated representation selection is based on the end-to-end bandwidth measurement, and ignores the involvement of data forwarders during the transmission. Unfortunately, this solution may suffer from high inaccuracy in terms of bandwidth measurement due to the multicast and flow dynamic caused by *Interest* aggregation and in-network caching.

A network-assisted CCN DAS solution, based on measurement of the available bandwidth from intermediate nodes to both server and client, was proposed in Jmal et al. (2017). A rate adaptation algorithm is then proposed which decides the representations to be requested based on both estimated bandwidth and buffer level. In Liu and Wei (2016), a hop-by-hop based rate control for ICN DAS was proposed, which enables each ICN router shape independently the rate according to the local traffic status. Benefiting from the hop-by-hop design, the rate of the requested video can quickly follow bandwidth variation. However, the associated heuristic control mechanism yields a suboptimal transmission rate, which affects delivery performance.

In Rainer et al. (2016), the representation selection problem is formulated as a multi-commodity flow problem in order to derive the upper bound of throughput gains. However, due to highly dynamic network conditions and preference on entire playback quality at the client side, formulation of the rate representation problem as a deterministic optimization is unsuitable.

Another literature (Hu et al., 2019) related with our work proposed a framework of joint optimizing the caching, transcoding and routing decisions for adaptive video streaming over ICN. Unlike (Hu et al., 2019) whose objective is to minimize the access delay and maximize the cache hit ratio, our work mainly focuses on maximize the overall transmission rate and quality of viewing representations of each client. In addition, we solve the transmission control problem in first stage by proposing a novel distributed switching mirror descent algorithm. We also consider and solve the representation adaption problem via stochastic perspective that is able to maximize the long term viewing quality and smooth video playback according to the available transmission rate derived from first stage problem.

3. Scenario description

The design targets primarily ICN DAS and a typical video delivery scenario is considered. For simplicity, it is assumed that all the requests in this scenario are made for adaptive video content.[1] Let us consider a simple ICN scenario as in Fig. 1, where a media server distributes video content to three ICN clients A, B, C via ICN router R. An SVC based MPEG-DASH video application is considered and video content is encoded into one base layer L and two enhancement layers H_1 and H_2, with bitrates l, h_1, and h_2, respectively. Thus, three possible types of requesting data rate, l, $l + h1$, $l + h1 + h2$, can be selected by clients. Let the $B_{(R,A)}$, $B_{(R,B)}$, $B_{(R,C)}$, $B_{(S,R)}$ denotes the access link capacities of A, B, C, and server bandwidth, respectively. Let following inequality eq. (1) hold.

$$\begin{aligned} & l < B_{(R,A)} < l + h_1 < B_{(R,B)} \\ & < l + h_1 + h_2 < B_{(R,C)} < 3l < B_{(S,R)} < 3l + h_1 \end{aligned} \tag{1}$$

Inequality (1) indicates that client A can only request the lowest representation, namely, video with base layer L. Clients B and C can make requests for the video with one and both enhancement layers, respectively. However, the server capacity can only concurrently support three clients with the lowest representation.

Three clients request the same video at different times. It is assumed the Least Frequently Used (LFU) caching policy is enabled at R. At t_1, requests from A and B arrive asynchronously, but within the *Interest* aggregation time window. At t_2, C sends its request, where $t_2 - t_1$ is less than the caching eviction time. In such cases, a conventional IP-based DAS treats the requests from the three different clients separately, and consequently each client will experience the lowest representation.

Thanks to the *Interest* aggregation and on-path caching, ICN provides better performance in such cases by inherently enabling multicast with multi-rate delivery. The requests from A and B arriving within aggregation time window are combined at R. In particular, only the requests for $L + H_1$ will be forwarded to the server. After accessing the data associated with $L + H_1$, R forwards the $L + H_1$ to B and L to A, respectively. In addition, ICN router R will also cache the content with representation $L + H_1$. Thus, for client C, the content of $L + H_1$ can be provided by R, while the server only needs to deliver the enhancement layer H_2 to C. Under such circumstances, all clients A, B and C achieve the highest representations within their bandwidth limitations. From streaming perspective, the server fully utilizes its capacity and simultaneously serves three clients with a single video flow with rate $l + h_1 + h_2$.

Table 1
Notifications used in problem formulation.

Symbol	Description
$\mathcal{V}, \mathcal{C}$	Universe of clients and providers
$\mathcal{E}$	Universe of links
x_{ij}	Transmission rate of client i to provider j
$\boldsymbol{x}$	Transmission rate configuration of network
$g(x_{ij})$	Utility for client i with rate x_{ij}
$f(\boldsymbol{x})$	Objective of problem **P1**
F_j	Video flows initiated by provider j
c_l	Capacity of link l
$F(t)$	Universe of flows in network at t
$d_{\min}, d_{\max}$	The lowest and highest bitrate of DAS
$p(v_i(t))$	Utility with video bitrate $v_i(t)$
$\overline{p}_i$	Objective of problem **P2** for client i
$v_i(t)$	Bitrate of requested video for i at time t
$\mathcal{T}$	Set of time slots
$x_i^*(\tau)$	The optimal transmission rate of i at τ
D	Set of bitrates of selectable video representations
$\|\|\cdot\|\|$	Cardinality of set

4. Problem formulation

As already mentioned, the ICN DAS transmission control is considered as a two-stage optimization problem. In the first stage, a generic flow control problem considering the multicast, multi-rate features of ICN is formulated. The goal is to optimize the overall transmission rate in ICN. Based on the optimized transmission rate, a representation adaptation problem is formulated in the second stage. This optimization focuses on long term user viewing experience and is performed by selecting the optimal encoding bitrate to request. Table 1 includes the notations used in problem formulation.

[1] without loss of generality, delivering DAS in ICN can be considered generic, and thereby the same design method can also be applied to other applications.

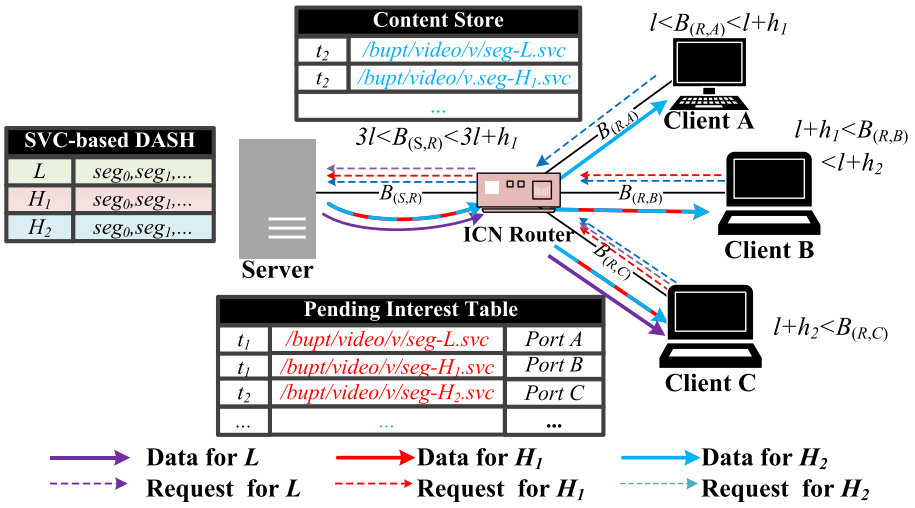


Fig. 1. Multicast, multi-rate delivery of ICN DAS.

4.1. First stage: ICN flow control optimization

The previous section illustrates how by using ICN can benefit DAS. However, these multicast and multi-rate features also challenge formulating the flow control problem that describe exactly DAS delivery in ICN.

Given a ICN network denoted by $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V}$ and $\mathcal{E}$ indicate the network nodes and links, respectively. $\mathcal{V}$ consists of clients, routers and servers. Consider the routers in ICN not only acting as data forwarders, but also as providers thanks to the in-network caching design. Hence, for simplification, $\mathcal{U}$ denotes the set of providers and routers in the network. Given the universe of clients $\boldsymbol{C}$, $\mathcal{V}$ is thereby equal to the union set of $\mathcal{U}$ and $\boldsymbol{C}$, namely:

$$\mathcal{V} = \boldsymbol{C} \cup \mathcal{U}$$

Let $e_{xy} \in \mathcal{E}$ denote the link between neighboring nodes $x \in \mathcal{V}$ and $y \in \mathcal{U}$. For any of consumer-provider pair $(i,j), i \in \mathcal{U}, j \in \boldsymbol{S}$ in network, we define the path between i and j by the set of links between them:

$$p(i,j) \triangleq \{e_{i,s_1}, e_{s_1,s_2}, e_{s_2,s_3}, \ldots, e_{s_k,j}\}$$

where $s_l \in \mathcal{U}$ $(l = 1, 2, 3, \ldots, n)$. Assuming the network $\mathcal{G}$ is fully connected, namely, for any consumer-provider pair (i,j) in $\mathcal{G}$, there always exists at least one path between them.

Considering network dynamics, it is assumed the time is slotted, such that $\mathcal{T} = \{1, 2, \ldots, t \ldots\}$. Let x_{ij} denote the data transmission rate of client $i \in \boldsymbol{C}$ accessing video content from provider $j \in \mathcal{U}$. We assume the utility function is concave and differential, such as logarithmic forms in Huang et al. (2018) and $4.75 - 4.5e^{-0.77x_{ij}}$ in Liu and Lee (2016). Reasons for this assumption are two folded: (1) The concave function ensures the existence and uniqueness of optimum (Boyd and Vandenberghe, 2004), which is important for optimality analysis on algorithm design; (2) From the practical perspective, concave function, especially with the logarithmic forms, are very suitable for capturing the user experiences of various network applications (Reichl et al., 2013) including the video streaming.

As discussed, single provider in ICN can potentially serve multiple clients with identical requests and heterogeneous transmission rates via a single data flow. By considering all the clients associated with one video flow, the overall utility of the clients served via the flow f_j initiated from j can be denoted by $\sum_{i \in c_{f_j}} g(x_{ij})$, where c_{f_j} denotes the clients requesting f_j. For any link $l \in \mathcal{E}$, its corresponding capacity is denoted by c_l. At every t, the main purpose of the first stage optimization is to maximize the overall utilities of flows in ICN as well as avoid violating the limitations of link capacities. Let the vector $\boldsymbol{x} \triangleq \{x_{ij}\}_{i \in \boldsymbol{C}, j \in \mathcal{U}}$ denote the transmission rate configuration. Thus, the first stage problem can be formulated as follows:

P1: For each slot t:

$$\text{Maximize} \quad f(\boldsymbol{x}) = \sum_{j \in \mathcal{U}} \sum_{f_j \in F_j} \sum_{i \in c_{f_j}} g(x_{ij}) \tag{2}$$

$$\text{Subject to} \quad \sum_{f_j \in s_l} \max_{i \in c_{f_j}(l)} x_{ij} \le c_l \quad l \in \mathcal{E} \tag{3}$$

$$\boldsymbol{x} \in \left[d_{\min}, d_{\max}\right]^{|C|} \tag{4}$$

where F_j denotes the video flows from j, s_l is the set of flows using l, $c_{j_f}(l)$ denotes the set of clients served simultaneously by j with flow f, while sharing link l. $\mathcal{F}(t)$ denotes the universe of video flows in the network at t, and $d_{\min}$ and $d_{\max}$ are the rates of lowest and highest representations, respectively. The objective indicated in eq. (2) is to maximize the utility of all clients in ICN DAS. The constraint from eq. (3) indicates that for any of the link $l \in \mathcal{E}$, the entire traffic handled by l should not exceed its capacity. Note that due to the multicast multi-rate feature of ICN, the sending data rate of each provider i over l is equal to the largest receiving rate of users accessing f from i via l. Hence, considering the left term of eq. (3), the total handling traffic can be derived by summing the maximum x_{ij} associated with each clients in $c_{f_j}(l)$. The constraint from eq. (4) ensures the flow configuration satisfies the viewing quality requirement. Accordingly, the following theorem ensures the concavity of problem **P1**.

Theorem 1. *Given the utility $g(.)$ of any video client $i \in \boldsymbol{C}$ is concave, problem **P1** can be considered as a non-smooth concave optimization problem, namely, a unique optimal flow configuration exists that maximizes* eq. (2) *under the constraints from* eq. (3) *and* eq. (4).

Proof. Since the theorem assumes each client's utility is concave, according to the concavity preservation of linear addition, the objective from eq. (2) is also concave. Hence, we only need to prove the convexity of constraints from eq. (3) and eq. (4). The constraint from eq. (3) is convex but non-smooth due to the convexity and piecewise of the maximum function. For each i, let us have two arbitrary selected data rates $x_i, y_i \in D$ with $x_i \le y_i$. The following inequality holds:

$$d_{\min} \le x_i \le \theta x_i + (1-\theta) y_j \le y \le d_{\max}, \forall \theta \in [0,1]$$

Hence, for the corresponding network flow vectors x and y, we have $\theta x + (1-\theta) y \in \left[d_{\min}, d_{\max}\right]^{|C|}$, namely $\left[d_{\min}, d_{\max}\right]^{|C|}$ is a convex set. This proves the theorem. □

For the purpose of formulating a generic flow control problem for ICN, the following discusses how **P1** can easily adopt different ICN delivery scenarios, after with minor modifications.

(1) *Multipath Scenario*: Recent studies exploiting the multipath features of ICN for enhancing the delivery performance, which enable ICN clients simultaneously access content from multiple difference sources. For such scenarios, assuming client i accesses content from multiple $[o_1, o_2, \ldots, o_M]$ interfaces. Let the corresponding rate of o_k be $x_{i,j_{o_k}}$. Hence, the total delivery rate of user i $x_i = \sum_{k=1}^{M} x_{i,j_{o_k}}$. Accordingly, the QoE function of u will be rephrased as $g\left(\sum_{i=1}^{M} x_{i,j_{o_k}}\right)$.

(2) *Unicast Scenario*: In this case, providers deliver the video flow to one user only. Applying our formulation to unicast requires setting the sum $\sum_{i\in l(s)} \max_{j\in s_i(u)_l} x_{i,j}$ in constraint from eq. (3) to $\sum_{s\in l(s)} x_s$, where x_i denotes the delivery rate of provider i.

4.2. Second stage: requesting BitRate adaptation problem

After the data transmission rate is determined, the clients will adapt the requested representations of video content according to the available network capacity. At each time slot t, let the maximum transmission data rate $x_i^*(t)$ be derived by solving t's **P1**. Due to the fact that $x_i^*(t)$ varies mainly because of the network dynamic behavior, the second stage mainly focuses on the long term optimization of user viewing experience. Given the utility of client i: $p(v_i(t))$, where $v_i(t)$ denotes the requested video bitrate at time t, the corresponding long term average utility is defined as in eq. (5).

$$\overline{p}_i \triangleq \lim_{t\to\infty} \frac{1}{t} \sum_{\tau=1}^{t} p(v_i(t)) \tag{5}$$

If $\overline{p}_i$ is the objective, the second stage optimization can be formulated as the following stochastic optimization:

P2: For each client i

$$\text{Maximize} \quad \overline{p}_i \tag{6}$$

$$\text{Subject to} \quad \lim_{t\to\infty} \frac{1}{t} \sum_{\tau=1}^{t} \frac{x_i^*(\tau)}{v_i(\tau)} \geq 1 \tag{7}$$

$$v_i(t) \in D \tag{8}$$

where D includes all possible video rate of all representations. Constraint from eq. (7) limits the average of the selected requested bitrate to less than the maximum transmission rate. The reason for introducing eq. (7) is explained next. During each processing interval $[t-1, t]$, the client consumes 1, and receives $x_i^*(\tau)/v_i(\tau)$ time units of content, respectively. When $x_i^*(\tau)/v_i(\tau) < 1$, the buffer level decreases. When the video buffer becomes empty, the playback is stalled. In contrast, if $x_i^*(\tau)/v_i(\tau) > 1$, the buffer level can be maintained at a non-empty level and hence ensuring the smooth playback.

5. Distributed asynchronous optimization algorithm design

In order to solve the formulated two stage optimization problem, a Distributed Asynchronous Optimization Algorithm (DAOA) for ICN DAS is proposed. DAOA derives the optimum in each of the two stages individually. For each time slot t in DAOA, the network first determines the optimal flow control configuration by solving **P1**, then, each client optimizes the requested video representation by solving problem **P2**.

5.1. Algorithm for P1

When analyzing eq. (2), eq. (3) and eq. (4), two major obstacles to solving problem **P1** are noted.

A) **Maximum Functions**: The constraint in eq. (3) contains the *maximum* function which is non-differentiable. Using traditional methods such as the Lagrangian method (Boyd and Vandenberghe, 2004) needs to solve the inverse of lagrangian's gradient, hence, requiring to consider all possible linear combinations of different x_{ij} in constraints. For example, let client 3,4 simultaneously access video flows from provider 1 using link l, while client 5 receives video from 2 via l. Thus, the capacity constraints for l can be denoted as $\max\{x_{31}, x_{41}\} + x_{52} \leq c_l$ which is non-differential. Conventional methods such as Lagrangian need to convert $\max\{x_{31}, x_{41}\} + x_{52} \leq c_l$ into two linear constraints: $x_{31} + x_{52} \leq c_l$ and $x_{41} + x_{52} \leq c_l$, hence, complicating the solving of Lagrangian function.

B) **Coupling of clients**: Eq. (2) is a linear summation of users' utility functions, thus eq. (2) can be calculated by enabling each client to calculate their own utility function individually. Namely, eq. (2) is separable at $x_{i,j}$. However, solving the constraints from eq. (3) requires coordination among all the clients using link l, which means data rate of clients are coupled by constraints. This makes the problem difficult to solve by distributed methods.

To overcome the above challenges, we propose the distributed switching mirror descent algorithm (DSMDA) which extends the switching mirror descent (SMD) (Beck and Teboulle, 2003) to the distributed scenarios. The switching of SMD indicates there are two different types of iterations, i.e., feasible/infeasible step, which perform alternatively according to whether the decision variable is within the feasible set or not. Each step applies the mirror descent paradigm which is a combination of a *Bregman Distance* and objective's subgradient. With the above design, SMD approaches the optimum without requiring the smoothness of constraints, namely, the first issue of maximum function is solved. Besides, it also avoids the complex computation of deriving the inverse differential forms as in Lagrangian. For more details on SMD, readers can refer to Appendix A.

However, original SMD is inapplicable to the distributed implementation of ICN flow control, since it requires to calculate the optimizer $\boldsymbol{x}$ centrally. Thus, directly applying SMD for transmission control needs a central coordinator for collecting the global status of the network and synchronously optimizes the entire clients' transmission rate. To accommodate the decentralization of ICN flow control, we propose DSMDA to enable each client to perform the switching mirror descent individually without coordinating with each other.

For this design purpose, we first rephrase the **P1** (2)-(4) as following convex optimization:

$$\text{Maximize} \quad \hbar(\boldsymbol{x}) = -\sum_{j\in\mathcal{U}} \sum_{f_j\in F_j} \sum_{i\in c_{f_j}} g(x_{ij}) \tag{9}$$

$$\text{Subject to} \quad (3)(4) \tag{10}$$

We introduce the *entropic distance* as eq. (11)

$$\|\boldsymbol{x}\|_e \coloneqq \sum_{o=1}^{n} \|x_o\| \tag{11}$$

and a *proxy function* in eq. (12)

$$d_e(\boldsymbol{x}) = \ln n + \sum_{o=1}^{n} x_o \ln x_o \tag{12}$$

where x_o is the o-th component of $\boldsymbol{x}$. Following proposition presents the important mathematical features of $d_e(\boldsymbol{x})$.

Proposition 1. *For all* $\boldsymbol{x} \in Q$, *function* $d_e(\boldsymbol{x})$ *defined in* eq. (12) *is differential and strongly convex.*

Proof. See Appendix B. □

According to Beck and Teboulle (2003), the *Bregman Distance* is defined as following:

$$V[(\boldsymbol{x}(k))](x) \triangleq d_e(\boldsymbol{x}) - d_e(\boldsymbol{x}(k)) - \langle \boldsymbol{x} - \boldsymbol{x}(k), \nabla d_e(\boldsymbol{x}(k)) \rangle \tag{13}$$

by substituting the (12) into (13), we have the *Bregman Distance* of *proxy function* (12)

$$V[\mathbf{y}](\boldsymbol{x}) = \sum_{o=1}^{n} (x_o(\ln x_o - \ln y_o) + (x_o - y_o))$$

For the o-th component of $\boldsymbol{x}$, we define the $V[\mathbf{y}](\boldsymbol{x})_o = x_o(\ln x_o - \ln y_o) + (x_o - y_o)$. Let $\mathcal{B}_h^e(\boldsymbol{y}, \boldsymbol{g}) \triangleq \arg\min_{\boldsymbol{x} \in Q} \{h\langle \boldsymbol{x}, \mathbf{g} \rangle + V[\mathbf{y}](\boldsymbol{x})\}$. By the optimal condition (Boyd and Vandenberghe, 2004) of $\mathcal{B}_h^e(\boldsymbol{y}, \boldsymbol{g})$, we have

$$0 \in h\mathbf{g} + \nabla_{\boldsymbol{x}} V[\mathbf{y}](x) \tag{14}$$

Where $\nabla_{\boldsymbol{x}}$ is the subgradient operator for $\boldsymbol{x}$, h is the step size. Solving the (14), we derive

$$\mathcal{B}_h^e(\boldsymbol{y}, \boldsymbol{g})_o = y_o e^{-hg_o - 2} \tag{15}$$

where $\mathcal{B}_h^e(\boldsymbol{y}, \boldsymbol{g})_o$ and g_o are the k-th component of $\mathcal{B}_h^e(\boldsymbol{y}, \boldsymbol{g})$, $\boldsymbol{g}$, respectively.

We let $f_l(\boldsymbol{x}) = \sum_{f_j \in s_l} \max_{i \in c_{f_j}(l)} x_{ij} - c_l$ and p_{ij} denote the set of links of the client i delivery path to j. The Lipschitz constant of the objective $\hbar(\boldsymbol{x})$ is denoted as M. The set $I_{p_{ij}}(t) = \{e_j \in p_{ij} | f_j(\boldsymbol{x}(t)) > h \parallel \nabla f_j(\boldsymbol{x}(t)) \parallel\}$, where $\nabla f_l(\boldsymbol{x}(t))$ is the subgradient for $f_l(\boldsymbol{x}(t))$. Let $T = 2\sigma/h^2$ is the upper limit in terms of the number of iterations,[2] where $\sigma = \max_{x \in D^{|C|}} r_0(\boldsymbol{x})$. Then, for client $i \in \boldsymbol{C}$ in ICN, DSDMA performs the following iterations:

Feasible step at t:

- each link l in p_{ij} returns $f_l(\boldsymbol{x}(k))$ and $\parallel \nabla f_l(\boldsymbol{x}(k)) \parallel_{E^*}$;
- If $k+1 < T$, $I_{p_{ij}}(k) = \emptyset$:
- $x_{ij}(k) = \mathcal{B}_h^e\left(\boldsymbol{x}(k), \frac{\nabla \hbar(\boldsymbol{x}(k))}{M}\right)_i$.

Infeasible step at t:

- each link l in p_{ij} returns $f_l(\boldsymbol{x}(t))$ and $\parallel \nabla f_l(\boldsymbol{x}(k)) \parallel_{E^*}$;
- If $i+1 < T$, $I_{p_{ij}}(t) \neq \emptyset$:
- Let $\zeta_i(\boldsymbol{x}(k)) = \max_{j \in I_{p_{ij}}(k)} f_j(\boldsymbol{x}(k))$;
- Let[3] $h_{k,i}(k) = \frac{\zeta_i(\boldsymbol{x})}{\|\nabla \zeta_k(\boldsymbol{x}(k))\|^2_{E^*}}$
- $x_{ij}(k+1) = \mathcal{B}^e_{h_{k,i}}(\boldsymbol{x}(k), \nabla \zeta_i(\boldsymbol{x}(k)))_i$.

Comparing with original SMD, the switching condition of feasible step has been modified to accommodate the distributed implementation. Besides, at infeasible step, our proposed DSMDA using the Bregman distance based on local maximum subgraident instead of the global maximum. Thus, calculation of eq. (15) for any client i requires only the information of $\zeta_k(\boldsymbol{x}(t))$ over i's delivery path p_{ij}. Therefore, in each iteration, for any of $k \in \boldsymbol{C}$, $x_k(t)$ can be individually derived by client k instead of coordinating with other clients using links over p_k. Thus, yielding the **coupled** challenge of ICN flow control problem.

5.2. Algorithm for P2

Recall that **P2** is a stochastic optimization problem. We propose the Virtual Queue-based Iteration Algorithm (VQIA) based on the min-drift-minus-plenty policy (Neely, 2010). We first introduce the virtual queue $H_i(t)$ for **P2**, which is updated at each t, as follows:

$$H_i(t) = \lceil H_i(t-1) - 1 + \frac{x_i^*(t)}{v_i(t)} \rceil^+ \tag{16}$$

[2] The reason of setting T is explained by Theorem 1.

[3] According to maximum theorem, the $\nabla \zeta_i(\boldsymbol{x}(k)) = \max_{j \in I_{p_{ij}}(k)} \nabla f_j(\boldsymbol{x}(t))$.

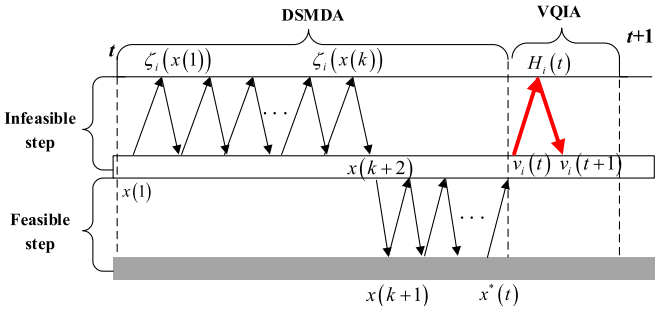


Fig. 2. Illustrations of asynchronous iteration in DAOA.

where $\lceil x \rceil^+$ denotes $\max\{x, 0\}$. By summation eq. (16) for all t, the inequality from eq. (17) holds when $H_i(0) = 0$.

$$\sum_{\tau=1}^{t} \left(\frac{x_i^*(\tau)}{v_i(\tau)} \right) - 1 \leq H_i(t) \tag{17}$$

Thus, when $H_i(t)$ is *stable*, i.e., $1/t \lim \sup_{t \to \infty} H_i(t) = 0$, (Neely, 2010), the constraints from eq. (7) hold. The equivalency between $H_i(t)$'s stability and holding of constraints from eq. (7) can also be explained from a physical perspective. By observing eq. (16), the physical meaning of $H_i(t)$ is the accumulative playback stalling time at t. Thus, *stable* for $H_i(t)$ indicates the accumulative playback stalling time is sublinear to the t. When t approximates the infinity, the expectation of $H_i(t)$ also close to the constant, namely, no increasing of stalling time and thereby confirms the smooth playback in long term of eq. (7), as discussed in Section 4.

According to Neely (2010), the optimum of **P2** can be derived by solving the following min-drift-minus-penalty expression in each time slot t.

$$\text{Minimize} \quad -\mathcal{V}p(v_i(t)) + H_i(t)\left(\frac{x_i^*(t)}{v_i(t)} - 1\right) \tag{18}$$

$$\text{Subject to} \quad v_i(t) \in D \tag{19}$$

where $\mathcal{V}$ is the penalty parameter.

To solve the above problem, we design VQIA as follows. At each time t, the VQIA of each client i collects $x_i^*(t)$ derived by DSMDA, and then the following iterations are performed:

$$H_i(t) = \lceil H_i(t-1) - 1 + \frac{x_i^*(t)}{v_i(t)} \rceil^+ \tag{20}$$

$$v_i(t+1) = \arg\min_{v_i \in D} \left(H_i(t)\left(\frac{x_i^*(t)}{v_i} - 1\right) - \mathcal{V}p(v_i) \right) \tag{21}$$

As the representation universe D is a discrete and finite set, eq. (21) can be easily derived by scanning D. The pseudo-code of VQIA is shown in **Algorithm 2**. Based on the two algorithms, the asynchronous optimization algorithm for ICN DAS can be illustrated as in Fig. 2. At each time slot t, the optimization procedure is divided into two stages. First, clients communicate with on-path links and determine $x_i^*(t)$ via DSMDA. Then, $x_i^*(t)$ will be delivered to VQIA and used to calculate the requested representation $v_i(t+1)$ and virtual queue $H_i(t)$.

6. Main theoretical results

6.1. Convergence analysis

This section discusses the main theoretical results of the proposed algorithms. It starts with the convergence of DSMDA that solves the ICN flow control problem. To state the theorem, first a gap function is defined as follows:

$$\rho_k = \frac{1}{S_k} \sum_{k \in \mathcal{F}} \frac{\hbar(\boldsymbol{x}(k))}{M} - L(\lambda) \tag{22}$$

where $M \triangleq \| \nabla \hbar(\boldsymbol{x}(k)) \|_{E^*}$, $\mathcal{F} = \{k | I_{p_{ij}}(k) = \emptyset, \forall i \in \mathcal{C}, k \in \{0, \ldots, T\}\}$, $S_k = \sum_{k \in \mathcal{F}} \frac{1}{M}$, and $L(\lambda)$ the Lagrangian of eq. (9) and eq. (10), which is given by:

$$L(\lambda) = \inf_{\boldsymbol{x} \in [d_{\min}, d_{\max}]^{|C|}} \{\hbar(\boldsymbol{x}) + \sum_{l \in \mathcal{E}} \lambda_l f_l(\boldsymbol{x})\}$$

According to Boyd and Vandenberghe (2004), the Lagrangian can be considered as a lowerbound of the optimum. Thereby, ρ_k indicates the average distance between solutions generated by DSMDA and theoretical optimal solution when the number of iterations reaches k. The lower value of ρ_k is, the higher the accuracy of DSMDA has. The following theorem ensures the upperbound ρ_k generated by the proposed DSMDA.

Theorem 2. *Given that $d_e(\boldsymbol{x})$ is bounded and $B_h^e(\mathbf{y}, \mathbf{g})$ has the formula described in* eq. (15)*, when $k > 2\sigma/h^2$, DSMDA yields a ρ_k such that*

$$\rho_k \leq \Theta h$$

where $\sigma = \max_{\boldsymbol{x} \in D^{|C|}} r_0(\boldsymbol{x})$, and $\Theta = \max_{t \in \boldsymbol{I}, l \in \boldsymbol{C}} \| \nabla f_l(\boldsymbol{x}(t)) \|_E^*$. Hence, $\boldsymbol{x}(k)$ generated by DSMDA converges to the optimal value.

Proof. See Appendix C. □

The above theorem reveals that the optimal convergence is highly related to the step h; the lower the h value is, the smaller gap to the optimum is. However, according to the stop criterion in Theorem 2, smaller gap also yields a high number of iterations. Thus, a trade-off between number of iterations and gap to the optimum should also be considered.

Unlike **P1** deriving the optimum in each time slot, **P2** focuses on the long term optimal rate adaption. Thus, to measure the performance of VQIA, we define the following gap function:

$$Gap_t = \sum_{\tau=1}^{t} \left(p\left(v_i^*\right) - p(v_i(t)) \right)$$

where v_i^* is the optimal solution of **P1**.[4] Gap_t indicates the distance between the solution derived by VQIA and the optimum. The following theorem bounds the Gap_t yielded by VQIA.

Theorem 3. *Given $p(.)$ is concave, iterations from* eq. (20) *and* eq. (21) *yield a Gap_t which satisfies the following inequality:*

$$Gap_t \leq \frac{B_i t}{\mathcal{V}}$$

where B_i is a constant satisfying $B_i \geq \left(x_i^*(t)/v_i(t)\right)^2 + 1, \forall t$.

Proof. See Appendix D. □

Therefore, according to the above discussion, the proposed DAOA converges to the optimum in both stages.

6.2. Complexity analysis

According to the description of Algorithm 1, the complexity of first stage optimization (DSMDA algorithm) depends on that of its main two components: links and clients, respectively. The processing complexity of the links component is determined by computation of the subgradient, which depends on the cardinality of set s_l. Hence, the complexity of link processing is $O(|s_l|)$, which is linear and ensures high flexibility of our algorithm. The client processing complexity is mainly determined by the products between number of links in the path and complexity of $\max_{j \in l_{p_k}(t)} f_j(\boldsymbol{x}(t))$, namely, $O\left(|p_k|^2\right)$. Normally, the length of the delivery path is relatively small, hence the processing load at clients is also light. This enables our algorithm run even on small devices (e.g. mobile phone, sensors). In the second stage, the client only needs to calculate $H_i(t)$ and $v_i(t+1)$, where $v_i(t+1)$ requires traversing the discrete set D. Hence, the complexity is bounded by $O(|D|)$.

Therefore, it can be concluded that DAOA is simple and scalable and can be used in a wide range of ICN scenarios.

6.3. Time varying adaptation

As in the second stage the stochastic optimization already accommodates the randomness of network variation, the time varying adaptability of DAOA is mainly determined by the first stage optimization. When formulating the flow control problem in eq. (2), eq. (3)and eq. (4), we assume that the objective function, video providers and routes are given and constant during every t. Yet, our algorithm can be easily extended to an environment with time variable features such as dynamic caching and routing, and a time-dependent objective function. Namely, our algorithm still can converge to the optimal solution under dynamic network conditions.

To cope with the time varying scenarios, the objective function of **P1** can be re-formulated as in eq. (23) by replacing the static $\mathcal{U}$, F_j and c_{f_j} with their time dependent form.

$$f(\boldsymbol{x}, t) = \sum_{i \in \mathcal{U}(t)} \sum_{f_j \in F_j(t)} \sum_{i \in c_{f_j}(t)} g\left(x_{i,j}(t)\right) \tag{23}$$

$l(s)$ in constraint from eq. (3) is replaced by $l(s,t)$, which is the time variant provider set that uses link l. Based on these changes, each end user still executes the same client algorithm as described in Algorithm 1, except for replacing the $\zeta_k(\boldsymbol{x}(k))$ by $\zeta_k(\boldsymbol{x}(k), t)$, which is time varying with the link capacity and number of clients. Intuitively, if the change in link routings and providers is relative slower than the convergence rate of algorithm $2\sigma/h^2$ (see Proof of Theorem 2), the algorithm still can converge to the optimal rates $\boldsymbol{x}^*$. This aspect is further illustrated in the experimental tests presented in Section 7.

7. Performance evaluation

In order to evaluate the performance of ICN DAS using the proposed DAOA, we model our algorithm by using *ndnSIM 2.0* (ndnsim in ns-3), an ICN simulation tool based on Network Simulator 3 (NS-3). First, we present the simulation setup. Then, we consider three scenarios with different network topologies: Forest-based, Content Delivery Network (CDN)-oriented, and Backbone network. In the first two topologies, we perform comparison between experimental and theoretical results of our proposed DAOA. In the third scenario, we test the delivery rate, playback stalling and viewing quality of DAOA over large scale scenarios by comparing with a state-of-art solution: ACCPndn (Karami, 2015).

7.1. Parameter settings

To implement our algorithm in ndnSIM, we add a new function named `Link-Subgradient` to `Net-Device-Face` class, which enables the links calculate $\nabla f_l(\boldsymbol{x})$ by collecting the delivery rate of the clients. We also create a new `DSMDA-Module` at each client to enable them deploy DAOA. To test video delivery quality, we install a DASH video application at each client in order to access the MPEG-DASH multimedia streaming with SVC-encoded format (Liu and Lee, 2016). Each of video segments contain a base layer with basic information for video decoding and several enhancement layers for improved representation. Therefore, the video can be adaptively delivered with a single base layer or with the base and one or more enhancement layers, according to the network conditions.

[4] Assume optimal solution is derived by an oracle that knows network dynamics at all t.

Algorithm 1 DAOA for ICN DAS.

Input: $\boldsymbol{x_0} = \arg\min_{\boldsymbol{x}\in D^{|C|}} d_e(\boldsymbol{x})$, h, T, $t = 0$;
Output: optimal rate configuration $\boldsymbol{x}^*$ at each t; $v_i(t)$

while $!t$ **do**
  //*DSMDA iterations*//
  while $\exists g,\ l \in p_{ij},\ k \le 2\sigma/h^2$ **do**
    receive rate $x_i(k)$ from all clients in set $l(s)$;
    compute subgradient $f_l(\boldsymbol{x}(k))$ and $\| \nabla f_l(\boldsymbol{x}) \|_E^*$;
    deliver $\nabla f_l(\boldsymbol{x}(k))$ to clients in $l(s)$;
    $k++$;
  end
  while $k \le 2\sigma/h^2$ **do**
    $I_{p_{ij}}(k) = \emptyset$;
    foreach $l \in p_{ij}$ **do**
      receive $f_l(\boldsymbol{x}(k))$ and $\| \nabla f_l(\boldsymbol{x}) \|_E^*$;
      calculate $f_l(\boldsymbol{x}) - h \| \nabla f_l(\boldsymbol{x}) \|_E^*$;
      if $f_l(\boldsymbol{x}) - h \| \nabla f_l(\boldsymbol{x}) \|_E^* \ge 0$ **then**
        add l to $I_{p_{ij}}$;
      end
    end
    if $I_{p_{ij}} \neq \emptyset$ **then**
      $\zeta_i(\boldsymbol{x}(k)) = \max_{j\in I_{p_{ij}}(t)} f_j(\boldsymbol{x}(k))$;
      calculate $h_{i,k} = \zeta_i(\boldsymbol{x}) / \| \nabla \zeta_i(\boldsymbol{x}(k)) \|_{E^*}^2$;
      $x_i(k) = \mathcal{B}^e_{h_{i,k}}(\boldsymbol{x}(k-1), \nabla\zeta_i(\boldsymbol{x}(k)))_i$;
    else
      $x_i(t) = \mathcal{B}^e_h\left(\boldsymbol{x}(t), \frac{\nabla\hbar(\boldsymbol{x}(t))}{M}\right)_i$;
    end
    communicate $x_i(t)$ to links $l \in p_k$;
  end
  if $k \le 2/h^2\sigma$ **then**
    $x_i^*(t) = x_i(k)$;
  end
  $k = k + 1$;
  //*VQIA iterations*//
  Collect $x_i^*(t)$ from DSMDA;
  Compute $H_i(t)$ by (20) and $v_i(t+1)$ by (21);
end
final ;

The utility functions in both **P1** and **P2** are set to $10\log\boldsymbol{x}$. The step size h is set to 10^{-3}, the penalty parameter V to 150, and the time interval between any t and $t+1$ is 2s, which is consistent with the length of a video segment. The link channel delay is fixed to 10 ms during the simulation. For the general error model randomly corrupting the packets, we use the BurstErrorModel in (ndnsim in ns-3), determining which burst of packets will be dropped according to an underlying distribution.

The testing video segment has one base layer and three enhancement layers. The base layer B has a bitrate of 2200 kbps, and enhancement layers $L1$, $L2$ and $L3$ have 1700, 1400 and 2700 kbps, respectively. Furthermore, to benefit data delivery from the ICN multicast feature, we also redesign the naming scheme that names the layers separately and hence the requests for identical layers of any given segment are aggregated. For example, assuming one client requests B and $L1$ and another requests B, $L1$ and $L2$, by separately naming each layer, the provider can serve these two flows simultaneously by delivering B, $L1$ and $L2$. The length of each video segment is 2s. 8 videos exist in the network and each of them is 240s long.

7.2. Experimental results

7.2.1. Forest-based topology

To evaluate the performance in terms of convergence at link side and time varying adaptation, we implement the proposed DAOA over a forest-based topology, which consists of 14 ICN routers and 13 clients. The link state and bandwidth of the forest-based topology are shown in Fig. 3. To simulate the network heterogeneous characteristics, the leaf ICN routers act as access points (AP) with different communication technologies. For instance, AP1 and AP5 act as edge routers in wired networks, where four different access bandwidth types are provided: 1 Mbps, 3 Mbps, 5 Mbps and 10 Mbps; AP2 and AP4 are wireless access points using the IEEE 802.11a protocol with 5 Mbps shared bandwidth. AP3 is a LTE network base station to simulate the cellular network environment which provides 4 Mbps access bandwidth to each end user. In this topology, the client generates video requesting flows following two different patterns:

(1) *Multicast access*, where the clients simultaneously access videos 1, 2 and 3 from S1 at 0s, 200s and 400s, respectively, while requesting videos 4, 5, and 6 at 100s, 300s and 500s from S2,

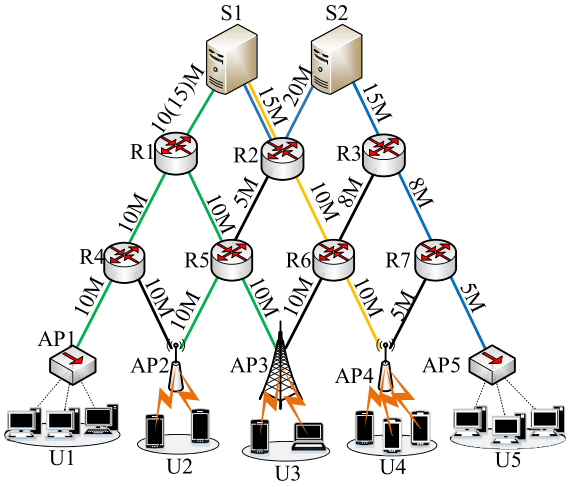


Fig. 3. Testing scenario with the forest-based topology.

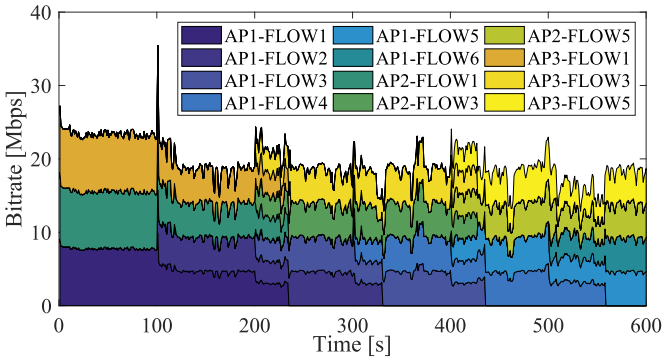


Fig. 5. Transmission rate of flows under multicast access.

respectively.

(2) *Random access*, where the clients randomly issue requests for videos during the simulation.

Multicast Access: Fig. 5 illustrates the transmission rates of flows at AP 1, 2, 3. Thanks to the multicast access of content, the server capacity during the simulation is fully utilized. Fig. 4(a)(b) shows a comparison between theoretical optimum and experimental data of several links. As the figures show, all curves corresponding to the experimental results converge well to those of theoretical computations, supporting the optimality of DSMDA. In addition, the figures also reveal that when theoretical optimum varies, the experimental results still converge fast to the updated optimum, hence, verifying the time adaptability of the algorithm at link sides.

At the beginning of simulation, all flows at three APs achieve the maximum transmission rate of 8 Mbps as shown in Fig. 5. This is also verified by the fact that transmission rates of (S1,R1), (R1,R4), (R1,R5), (R5,AP3) and (R4,AP1) are 8 Mbps in Fig. 4(a)(b). After 100s, new flows have been added to the network, fact which reduces the average transmission rate to 5 Mbps. However, the total transmission rate at (S1,R1) increases to 10 Mbps, since (S1,R1) currently delivers two different video flows (video 1 and 4) to end users. Therefore, each video flows has 5 Mbps. Due to the fact that the downstream link capacity is 5 Mbps, the transmission rate of (S2,R3) is limited to 5 Mbps. With increasing number of flows in network, the average flow transmission rate decreases, but links are kept fully utilized, as shown in Fig. 4(a)(b). This verifies the optimal performance of the DSMDA flow control.

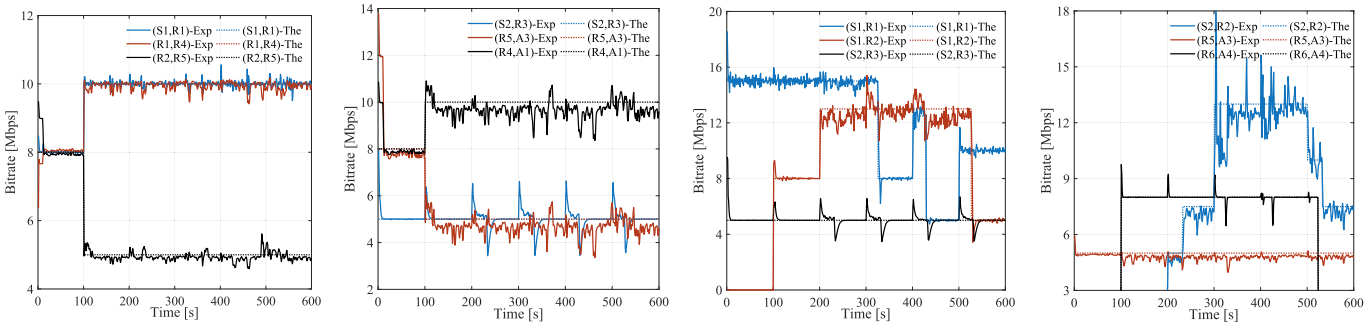
Fig. 6 illustrates the buffer level of flows during simulation. As shown in the figure, due to the multicast feature, the variations of buffer levels at different APs have similar variation trends. Additionally, the buffer level remains stable at a relative high level during the simulation. These verify that the algorithm proposed in the second stage ensures smooth playback at the video clients.

Random Access: In this case, we increase the (S1,R1) link capacity to 15 Mbps. Fig. 7 shows that the total transmission rate of flows at AP1, AP2 and AP3 equals the (S1,R1) link capacity (i.e. S1 serves the flows from AP1, AP2 and AP3). The average transmission rate in random access case is lower than that in the multicast case, especially when the flow increases after 100s. Additionally, according to Fig. 4(c)(d), the link capacity utilization is lower than that of multicast access cases. This can be explained by the fact that the transmission rate in unicast relies heavily on the bottleneck link. Thus, when the number of flows over the bottleneck links increases, the average transmission rate decreases, which in turn reduces the utilization of other links. Following the comparison of the two different cases, a routing policy in ICN which aggregates as many *Interests* requests as possible is suggested for improving the network capacity.

Fig. 8 illustrates the buffer level of flows in the simulation. According to the figure, although the transmission rate is lower than that of the multicast cases, the buffer levels of all flows are maintained at high levels. This demonstrates that our algorithm can ensure smooth playback even in highly loaded bandwidth scenarios.

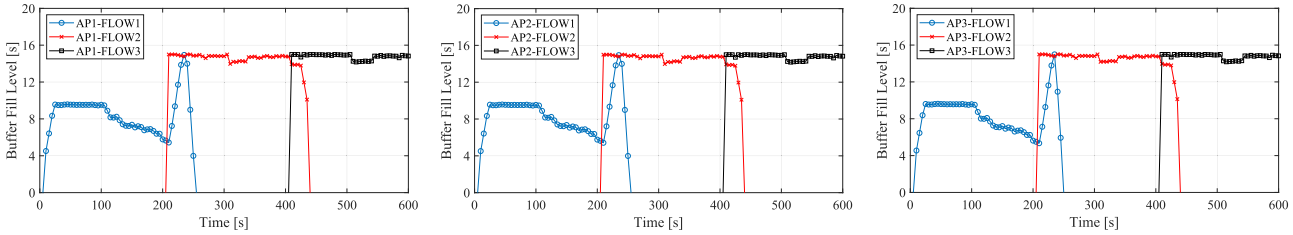
7.2.2. CDN-based topology

In the CDN-based topology, three servers deliver content to four nodes acting as ICN clients clusters. Each client accesses the content by a pre-set forwarding path in FIB. The corresponding topology is shown in Fig. 9. The main purpose of this topology is to test the performance of algorithm convergence at client side and bitrate adaptation. We consider two different types of content access pattern:



(a) Multicast-1 (b) Multicast-2 (c) Unicast-1 (d) Unicast-2

Fig. 4. Theoretical and experimental link loads in Forest-based topology.



(a) Buffer Level-AP1 (b) Buffer Level-AP2 (c) Buffer Level-AP3

Fig. 6. Transmission rate of flows under multicast access.

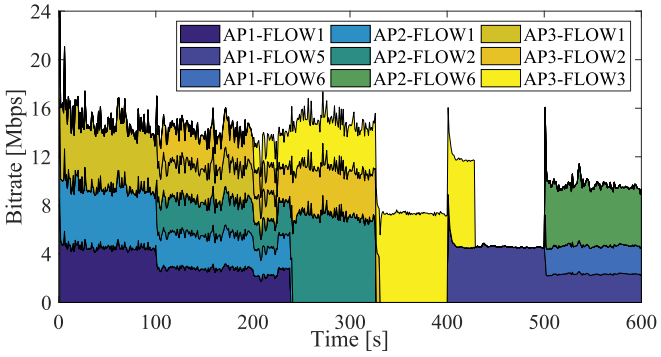
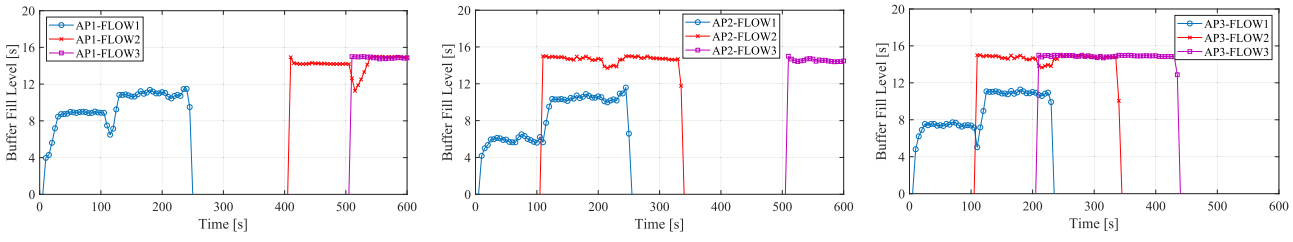


Fig. 7. Transmission rate of flows under multicast access.

(1) *Multicast access*, where client 1 and client 2 access simultaneously video 1 and client 3 and client 4 transfer simultaneously video 2 during the simulation. The identical requests will be aggregated at routers to provide multicast-oriented delivery.
2) *Random access*, where each client randomly requests a video during the simulation. In this case, multicast still exists when identical *Interest* of content arrive within a small time window, yet as most requests arrive asynchronously, they are responded to individually.

Performance of Multicast Access: Client 1 and client 2 reveal similar trends since they are concurrently served via $R10$. Client 3 and client 4 have also similar behavior as they are concurrently served via $R11$. By observing Fig. 10, all clients' requested bitrates converge well to the transmission rate. Especially, at the 300s, a large data flow has been added to link $(R5, R8)$, which results in a quick decrease of the transmission rates of client 1 and client 2. The requested bitrates of client 1 and client 2 also fast follow the decrease in the transmission rate, hence, proving the time-varying adaptation of proposed algorithm. Additionally, the buffer levels corresponding to the blue curves in figures are maintained at good values (at 10s on average) and ensure smooth playback. Especially, when the requested bitrate switches, the buffer level first decreases and then fast recovers. This confirms that the proposed algorithm supports smooth video playback.

Performance of Random access: Similar to the multicast case, the requested client bitrate in unicast, shown in Fig. 11, also converges to the transmission rate. Each client is served individually due to the random access, hence, their transmission rates are diverse. Although, the average requested bitrate is lower than that of multicast access, the buffer levels at each client still ensure smooth playback at clients.



(a) Buffer Level-AP1 (b) Buffer Level-AP2 (c) Buffer Level-AP3

Fig. 8. Buffer level of flows under unicast case.

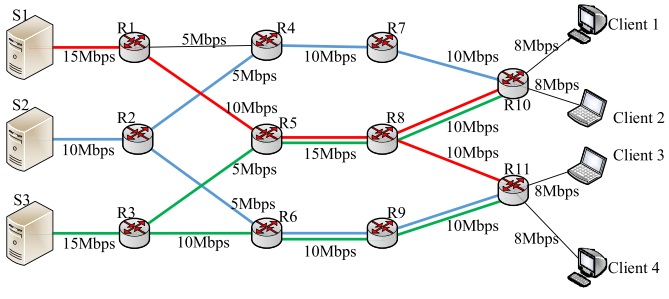


Fig. 9. Topology of CDN-based scenario.

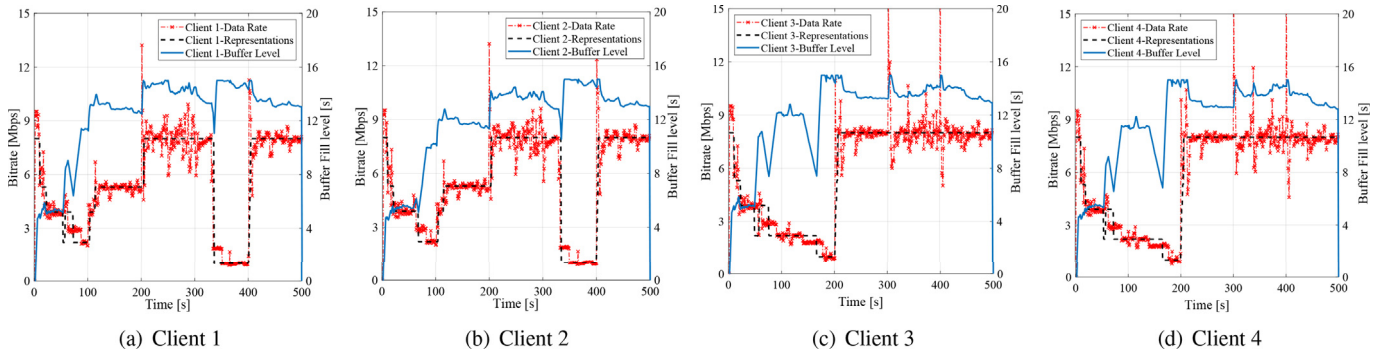


Fig. 10. Transmission rate, Video Bitrate, Buffer Level vs Simulation Time(Multicast Case).

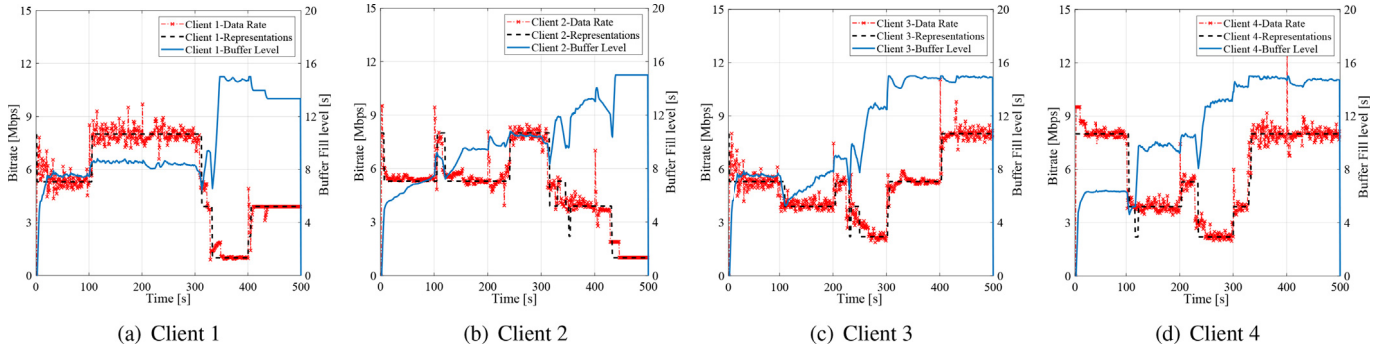


Fig. 11. Transmission rate, Video Bitrate, Buffer Level vs Simulation Time(Unicast Case).

Table 2
Large scale topology parameters.

Parameter	Value
Number of Clients	3000
Number of ICN Routers	500
Number of DAS content server	10
Network Link Bandwidth Range	[5,10] Mbps
Server Link Capacity	100 Mbps
Video Request Pattern	Poisson ($\lambda = 0.1$)
Link Degree Distribution	Power Law

Therefore, the results illustrated in Figs. 10 and 11 confirm that our proposed algorithm not only achieves optimal bitrate adaptation, but also ensures smooth playback.

7.2.3. Backbone network topology

To further evaluate the performance in a large scale scenario with heavy load traffic, we build a backbone network topology for implementing DAOA and two learning-based schemes: ACCPndn (Karami, 2015) and ACCP (Liu et al., 2019) with buffer-based bitrate adaptation. Table 2 shows the topology settings. The arrival rate of video requests at each edge router follows the Poisson distribution with parameter 0.1. ACCPndn employs a time-lagged feedforward network (TLFN) to predict the network congestion degree, and a non-linear fuzzy logic-based control system to regulate the transmission rate at each router. Similarly, ACCP also consists of congestion forecasting and transmission control. The difference is that ACCP forecasts the congestion via a deep learning framework and regulates the sending rate by estimating the average queue length of Interest. To investigate the performance effect of routing, we further consider two different routing policies

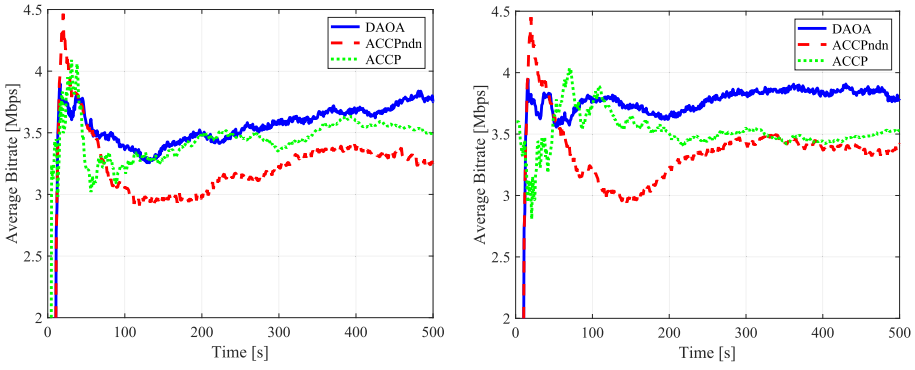


Fig. 12. Averge bitrate vs. simulation time.

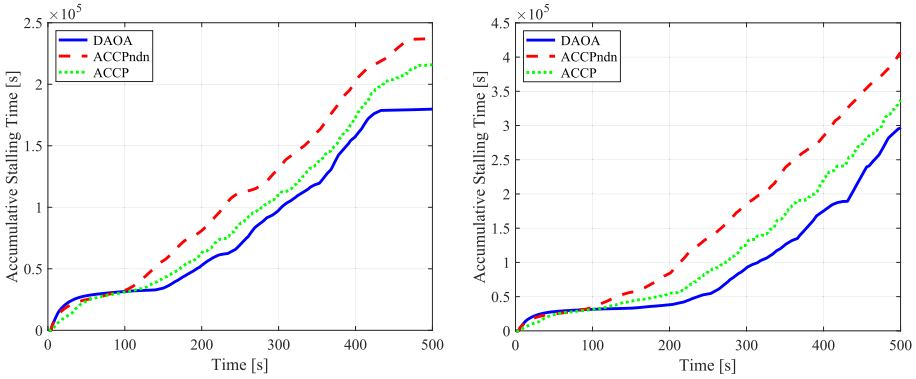


Fig. 13. Averge stalling time vs. simulation time.

(ndnsim in ns-3): *BestRoute* and *Multicast*. *BestRoute* tries to discover the shortest path to the source for each requested flow, whereas the *Multicast* routing tries to aggregate as many identical requests as possible. We measure two video quality-related delivery parameters during this simulation: **Average bitrate** and **Accumulative playback stalling time**.

Average BitRate (ABR): We define ABR at time T as the arithmetic mean of video bitrate average for all flows. Fig. 12(a) and (b) show the ABR of three solutions under *BestRoute* and *Multicast*, respectively. From the figures we observe that after the 20s, all curves experience a decreasing trend because of the increasing number of users, and are stabilized after 300s (because the rate of arrival is averagely equal to that of departure).

As expected, both solutions under the *BestRoute* outperform the *Multicast*, given a high bandwidth utilization achieved by the multicast delivery. The curves corresponding to ACCPndn decrease sharply and maintained at a relatively low level. Two solutions of ACCP perform better than that of ACCPndn, respectively. DAOA under *Multicast/BestRoute* both experience a slight decrease and enter the stable phase in the latter half of the simulation. In the stable phase, DAOA achieves about 20% (13%) and 22% (10%) increase of ABR in comparison with ACCPndn (ACCP) in *Multicast* and *BestRoute*, respectively.

The accuracy of the congestion prediction determines the performance of two learning-based solutions. ACCP applies deep learning to provide more accurate congestion forecasting and thereby results in a higher ABR than that of ACCPndn. However, both of them rely on heuristic rate control methods, whose rate configuration are suboptimality. ABR in DAOA is optimized distributively by the proposed two-stage optimization which tends to the theoretical optimal bound, hence, providing the best performance.

Accumulative playback stalling time (APST): In DAS, the viewing process stalls when the buffer is empty and restarts when enough content is buffered. The time interval between playback stalling and restarting is defined as the playback stalling time. The longer APST is, the worse the video quality of experience perceives. We define the APST

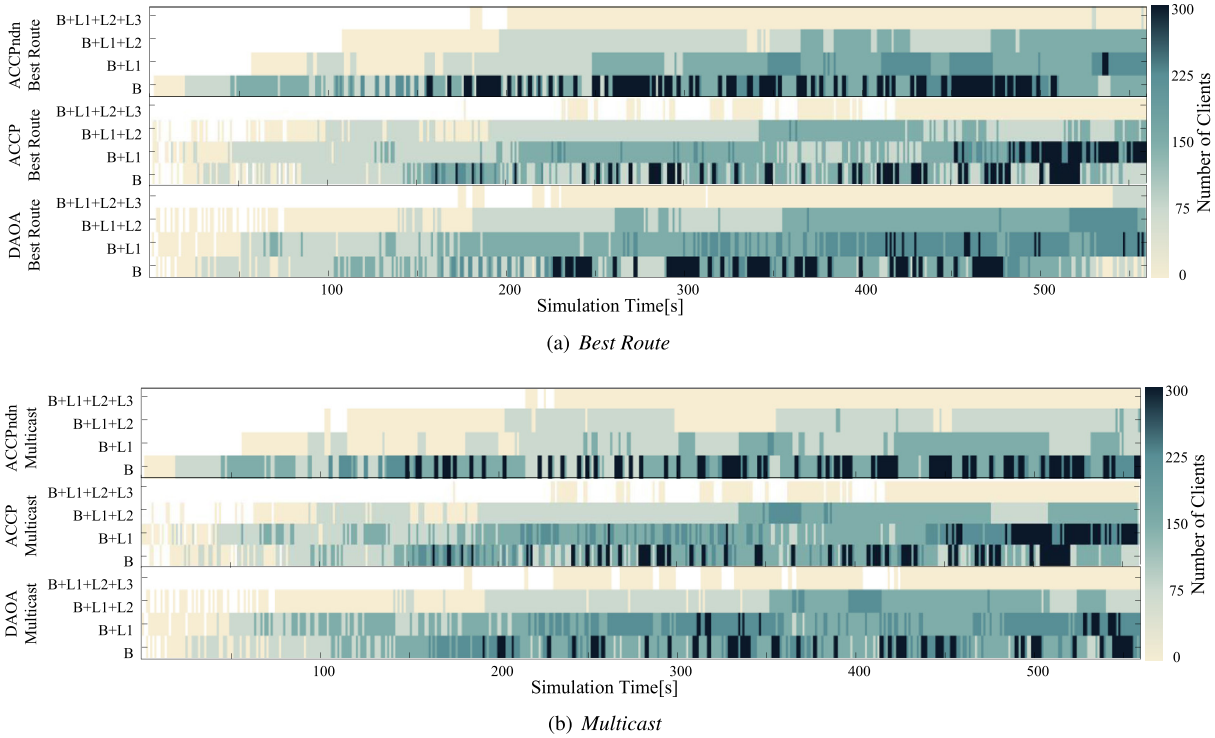


Fig. 14. The number of clients at each representation of retrieved segments during the simulation.

as the sum of playback stalling times during the simulation. As Fig. 13 shows, all curves experience a fast increasing trend because of the increment of stalling frequency. In both routing cases, DAOA achieve better CPST than ACCPndn and ACCP. Particularly, DAOA of *BestRoute* performs better than that of *Multicast*, as *BestRoute*'s priority is to shorten the transmission delay.

Fig. 14(a) and (b) show the number of clients that retrieve certain quality of each segment within the three solutions under *BestRoute* and *Multicast*. The deeper the color is, the higher the number of clients access the video with this representation. Clearly, in both ACCPndn and ACCP cases, most users are requesting the video with two bottom layers. Besides, the large bright area in these two also implies the unsmooth playback of clients using ACCP and ACCPndn, which confirms the result of high CPST for ACCP and ACCPndn illustrated in Fig. 13. In two solutions of DAOA, more users can retrieve the video content with higher bitrate given the deeper color in the space for B + L1+L2 and B + L1+L2+L3.

The reason that DAOA outperforms the ACCPndn and ACCP in terms of the number of clients at each representation are as follows: ACCPndn and ACCP predict the network congestion via machine learning-based methods whose accuracy is not guaranteed. Inaccurate forecasting may either result in frequent playback freeze (more bright area in the figure) or unsatisfied viewing experience (fewer requests for higher bitrate), which are both undesirable to the ICN DAS. ACCPndn and ACCP also underutilize the link capacity since their heuristic control algorithm yields a suboptimal of rate configuration. Instead, DAOA theoretically ensures the optimality of transmission control and thereby delivers video with a higher data rate. Besides, the proposed stochastic optimization based rate adaption timely selects the optimal bitrate of the video while ensuring the smooth playback in the long-term perspective. Hence, comparing with other two solutions, DAOA not only smoothen the viewing process but also enables clients to access higher representations.

8. Conclusion

In this paper, we focus on a joint optimization of flow control and bitrate adaptation in ICN DAS. The target problem is formulated as a two stage optimization problem. The ICN flow control with multicast multi-rate features is formulated as a non-smooth concave optimization in the first stage. While the playback representation adaption is formulated as a stochastic optimization, which aims to provide long term optimization for user viewing experience over a random dynamic network status. To solve the first stage, the proposed DSMDA enables the DAS clients individually optimize transmission rate at each time slot. Benefiting from the mirror descent and divided feasible/infeasible iterations, DSMDA provides a lightweight implementation with relatively low computation and communication overhead. The VQIA is further employed at second stage, which dynamically adapts the requesting representations with long term utility optimization while also stabilizing the increment of playback stalling time.

Main theoretical results, including the convergence Proof, computation complexity and time varying adaption, are provided. We also conduct a series of simulation tests under different network scenarios. The results not only validate the optimal convergence of DAOA in both flow control and bitrate adaption, but also illustrate how DAOA outperform state-of-art solutions in terms of quality of viewing experience. Future work will include joint consideration of routing optimization and mobility adaptation.

CRediT authorship contribution statement

Mu Wang: Conceptualization, Formal analysis, Methodology, Writing - original draft. **Changqiao Xu:** Investigation, Funding acquisition, Project administration, Resources, Writing - review & editing, Conceptualization, Supervision, Methodology. **Xingyan Chen:** Data curation, Software, Formal analysis. **Lujie Zhong:** Supervision, Investigation. **Gabriel-Miro Muntean:** Writing - review & editing.

Acknowledgment

This work is supported by the National Natural Science Foundation of China (NSFC) under Grant Nos. 61871048 and 61872253; Supported by National Key R&D Program of China (2018YFE0205502).

Appendix A. Principle of Switching Mirror Descent Method

The proposed DSMDA extends SMD (Beck and Teboulle, 2003), which considered a combination between the mirror descent (Nesterov, 2009) and switching subgradient methods, designed to perform functional optimization. SMD's basic design principles are described next. *Bregman Distance* is firstly introduced: $V[(\boldsymbol{x}(k))](x)$:

$$V[(\boldsymbol{x}(k))](x) \triangleq \varphi(\boldsymbol{x}) - \varphi(\mathbf{x}(\mathbf{k})) - \langle \boldsymbol{x} - \mathbf{x}(\mathbf{k}), \nabla \boldsymbol{\varphi}(\boldsymbol{x}(\mathbf{k}))\rangle$$

where $\varphi(\boldsymbol{x})$ is the *proxy function* with continuous and strongly convex, and ∇ is the sub-gradient operator. The mirror decent expression is then derived by:

$$\boldsymbol{\mathcal{B}}_{h_{k,i}}(x(k), \mathbf{g}) \triangleq \arg\min_{\boldsymbol{x}\in Q} \{h_{k,i}\langle \boldsymbol{x}, \mathbf{g}\rangle + V[(\boldsymbol{x}(k))](x)\}$$

$h_{k,i}$ is considered as the stepsize. Given the following functional constraints optimization:

$$\min f_0(\boldsymbol{x}), \mathbf{s.t} \quad \mathbf{F} \leq 0$$

where $F \triangleq \{f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \ldots, f_n(\boldsymbol{x})\}$, SMD for the above problem performs the following iterations:

1. **feasible step**: Given h, if $\forall f_i(\boldsymbol{x}) \in \mathbf{F}, f_i(x(t+1)) < h \parallel \nabla f_i(\boldsymbol{x}(t)) \parallel_{E^*}$,

   $\boldsymbol{x}(k) = \boldsymbol{\mathcal{B}}_h(\boldsymbol{x}, \nabla f_0(x(t)) / \parallel \nabla f_0(\boldsymbol{x}(k)) \parallel_{E^*})$

2. **infeasible step**: else,

   $\boldsymbol{x}(k) = \boldsymbol{\mathcal{B}}_{h_{k,i}}(\boldsymbol{x}, \nabla f_i(x(k)))$

   where $h_{k,i}$ is equal to $f_i(x(k)) / \parallel \nabla f_i(x(k)) \parallel_{E^*}$.

Appendix B. Proof of Proposition 1

Proof. As d_e is a sum of $x_i \ln x_i, i = 1, \ldots, n$, which are all differential, d_e is also differential.

For all i, the corresponding first partial derivative is:

$$\nabla_i d_e(\boldsymbol{x}) = \ln x_i + 1$$

and the second partial derivative for all i, j is:

$$\nabla_{ij} d_e(\boldsymbol{x}) = \begin{cases} \frac{1}{x_i}, i = j \\ 0, \text{otherwise} \end{cases}$$

The Hessian matrix $\mathbf{H}_d$ of d_e is a diagonal matrix with diagonal elements $\frac{1}{x_i}, i = 1, \ldots, n$. Because $x_i > 0$, we have $\mathbf{y}^T \mathbf{H}_d \mathbf{y} > 0$ for all $\mathbf{y} \in \mathcal{R}^n$, namely, matrix $\mathbf{H}_d$ is positive definite. Therefore, according to Boyd and Vandenberghe (2004), d_e is strongly convex.

□

Appendix C. Proof of the Theorem 2

We first introduce the following lemma.

Lemma 1. *Recall that $d_e(\boldsymbol{x}) = \ln n + \sum_{i \in C} x_i \ln x_i$, given the $B_h^e(\mathbf{y}, \mathbf{g})$, positive constant h and $\mathbf{g}_i$ subgradient of $f(\boldsymbol{x}_i)$, for all $x_i, y_i \in D$ the i-th component of $\boldsymbol{x}, \boldsymbol{y}$, respectively. Let $x_i^* = B_h^e(\mathbf{y}, \mathbf{g})_i$, then,*

$$\left(h\mathbf{g}_i + \nabla_i d_e\left(\boldsymbol{x}_i^*\right) - \nabla_i d_e(\boldsymbol{x})\right)\left(y_i - x_i^*\right) > 0$$

where $\nabla_i d_e(.)$ is the i-th components of subgradient $\nabla d_e(.)$.

Proof. Let $\ell(y_i) = hf(x_i) y_i + \nabla_i d_e(\boldsymbol{y}) - d_e(\boldsymbol{x}) y_i$, then the derivatives $\nabla_i \ell(y_i) = h\mathbf{g}_i + \nabla_i d_e(\boldsymbol{y}) - \nabla_i d_e(\boldsymbol{x})$. Then the lemma holds when $\nabla_i \ell\left(x_i^*\right)\left(y_i - x_i^*\right) > 0$. We proof the lemma by contraction, namely, assume that $\nabla_i \ell\left(x_i^*\right)\left(y_i - x_i^*\right) < 0$, Let $\varphi_i(\alpha) = \ell\left(x_i^* + \alpha\left(y_i - x_i^*\right)\right)$, we have $\varphi_i(0) = \ell\left(x_i^*\right)$, and $\varphi_i'(\alpha) = \nabla_i \ell\left(x_i^* + \alpha\left(y_i - x_i^*\right)\right)\left(y_i - x_i^*\right)$. Therefore, according to the assumption, we have $\varphi_i'(0) < 0$, namely, $\varphi_i'(\alpha)$ is monotonically decreasing around 0. Thus, there exists a β, such that

$$\varphi_i(\alpha) < \varphi_i(0) = \ell\left(x_i^*\right)$$

When the distance is entropy distance, we have $B_h^e(\mathbf{y}, \mathbf{g})_i = \arg\min_* x_{i \in D} \ell(x_i)$. Thus x_i^* yields $\min_{x_i \in D} \ell(x_i)$, which duces the contradiction.

□

Now we can Proof Theorem 2.

Proof. Defining $\mathcal{F} = \{k | I_{p_{ij}}(k) = \emptyset, \forall i \in \boldsymbol{C}, k \in \{0, \ldots, T\}\}$ we let $\boldsymbol{\mathcal{I}}$ be the set $\{k | k \in \{0, \ldots, T\}\}$. Then

$$\begin{aligned} \boldsymbol{\mathcal{I}} &= \mathcal{F} \cup \{k | k \in \{0, \ldots, T\}, \exists i \in \boldsymbol{C}, I_{p_{ij}}(k) \neq \emptyset\} \\ &= \mathcal{F} \cup \{k | k \in 0, \ldots, T, \exists m, n \in \boldsymbol{C}, I_{p_{mj}}(k) \neq \emptyset, I_{p_{nj}}(k) = \emptyset\} \\ &\quad \cup \{k | k \in 0, \ldots, T, \forall i \in \boldsymbol{C}, I_{p_{ij}}(k) \neq \emptyset\} \end{aligned} \tag{C.1}$$

We define the gap function

$$\rho_k = \frac{1}{S_k} \sum_{k \in \mathcal{F}} \frac{\hbar(\boldsymbol{x}(k))}{M} - L(\lambda) \tag{C.2}$$

where $S_k = \sum_{k \in \mathcal{F}} \frac{1}{M}$, and $L(\lambda)$ the Lagrangian of (9) (10), which is given by

$$L(\lambda) = \inf_{\boldsymbol{x} \in \boldsymbol{D}^{|C|}} \{\hbar(\boldsymbol{x}) + \sum_{l \in \mathcal{E}} \lambda_l f_l(\boldsymbol{x})\}$$

Let

$$\lambda_k^{(0)} = hS_k = h \sum_{k \in \mathcal{F}} \frac{1}{M} \tag{C.3}$$

$$\lambda_l = \frac{1}{\lambda_k^{(0)}} \sum_{k \in C_l} h_{k,i} \tag{C.4}$$

where $C_l = \{k | k \in \{0, \ldots, T\}, \exists i, f_l = \max_{e \in p_{ij}} f_e(\boldsymbol{x}(k))\}$.

Recall that $\lambda_k^{(0)} = hS_k$, therefore,

$$\begin{aligned}
\lambda_k^{(0)} \rho_k &= hS_k . \rho_t \\
&= hS_k \sup_{x(k) \in D^{|C|}} \left\{ \frac{1}{S_k} \sum_{k \in \mathcal{F}} \frac{\hbar(\boldsymbol{x}(k))}{M} - \hbar(\boldsymbol{x}) - \sum_{l \in \mathcal{E}} \lambda_l f_l(\boldsymbol{x}) \right\} \\
&= \sup_{x(k) \in D^{|C|}} \left\{ h \sum_{k \in \mathcal{F}} \frac{\hbar(\boldsymbol{x}(k))}{M} - \lambda_k^{(0)} \hbar(\boldsymbol{x}) - \sum_{l \in \mathcal{E}} \left(\sum_{k \in C_l} h_{k,i} \right) f_l(\boldsymbol{x}) \right\} \\
&= \sup_{x \in D^{|C|}} \left\{ h \sum_{k \in \mathcal{F}} \frac{\hbar(\boldsymbol{x}(k)) - \hbar(\boldsymbol{x})}{M} - \sum_{k \notin \mathcal{F}} h_{k,l} f_l(\boldsymbol{x}) \right\}
\end{aligned} \tag{C.5}$$

where $h_{k,l} = h_{k,i}, k \in C_l, i \in \{i | f_l = \max_{e \in p_{ij}} f_e(\boldsymbol{x}(k))\}$, Then according to the convexity of $\hbar(\boldsymbol{x})$ and $f_l(\boldsymbol{x}), l \in \mathcal{E}$, we further have equalities,

$$\lambda_k^{(0)} \rho_k \leq \sup \left\{ h \sum_{k \in \mathcal{F}} \frac{\langle \nabla \hbar(\boldsymbol{x}(k)), \boldsymbol{x}(k) - \boldsymbol{x} \rangle}{M} + \sum_{k \notin \mathcal{F}} h_{k,l} \left(\langle \nabla f_l(\boldsymbol{x}(k)), \boldsymbol{x}(k) - \boldsymbol{x} \rangle - f_l(\boldsymbol{x}(k)) \right) \right\} \tag{C.6}$$

According to the constitution of $\mathcal{I}$, we consider three different cases:

Case I: when $k \in \mathcal{F}$ let $r_k(\boldsymbol{x}) = V[\boldsymbol{x}(k)](\boldsymbol{x}) = \sum_{i \in C} x_i (\ln x_i - \ln x_i(k))$, then

$$\begin{aligned}
& r_{k+1}(\boldsymbol{x}) - r_k(\boldsymbol{x}) \\
&= d_e(\boldsymbol{x}) - d_e(\boldsymbol{x}(k)) - \langle \nabla d_e(\boldsymbol{x}(k)), \boldsymbol{x} - \boldsymbol{x}(k+1) \rangle \\
&\quad - \left[d_e(\boldsymbol{x}) - d_e(\boldsymbol{x}(k+1)) - \langle \nabla d_e(\boldsymbol{x}(k)), \boldsymbol{x} - \boldsymbol{x}(k) \rangle \right] \\
&= \langle \nabla d_e(\boldsymbol{x}(k)), \boldsymbol{x} - \boldsymbol{x}(k+1) \rangle - d_e(\boldsymbol{x}(k+1)) \\
&\quad - \left[-d_e(\boldsymbol{x}(k)) - \langle \nabla d_e(\boldsymbol{x}(k)), \boldsymbol{x}(k+1) - \boldsymbol{x}(k) \rangle \right]
\end{aligned} \tag{C.7}$$

According to the strongly convex of $d_e(\boldsymbol{e})$(By Proposition 1), we further have,

$$\begin{aligned}
& r_{k+1}(\boldsymbol{x}) - r_k(\boldsymbol{x}) \\
&\leq \langle \nabla d_e(\boldsymbol{x}(k)) - \nabla d_e(\boldsymbol{x}(k+1)), \boldsymbol{x} - \boldsymbol{x}(k+1) \rangle \\
&\quad - \frac{\alpha}{2} \| \boldsymbol{x}(k+1) - \boldsymbol{x}(k) \|_E^2
\end{aligned} \tag{C.8}$$

Noted when $k \in \mathcal{F}$ that for each $i \in \boldsymbol{C}$, $\boldsymbol{x}_i(k+1) = \mathcal{B}_h^e \left(\boldsymbol{x}_i(k), \frac{\nabla \hbar(\boldsymbol{x}(k))}{M} \right)_i$. Because $\mathcal{B}_h^e$ is separable, and hence, we have $\boldsymbol{x}(k+1) = \mathcal{B}_h^e \left(\boldsymbol{x}_i(k), \frac{\nabla \hbar(\boldsymbol{x}(k))}{M} \right)$. In the view of optimality condition of $\mathcal{B}_h^e$,

$$\begin{aligned}
& \frac{h}{M} \langle \nabla \hbar(\boldsymbol{x}(k+1)), \boldsymbol{x}(k+1) - \boldsymbol{x} \rangle \\
&\leq \langle \nabla d_e(\boldsymbol{x}(k+1)) - \nabla d_e(\boldsymbol{x}(k)), \boldsymbol{x} - \boldsymbol{x}(k+1) \rangle
\end{aligned} \tag{C.9}$$

In this case, we have

$$\begin{aligned}
& r_{k+1}(\boldsymbol{x}) - r_k(\boldsymbol{x}) \\
&\leq -\frac{h}{M} \langle \nabla \hbar(\boldsymbol{x}(k+1)), \boldsymbol{x}(k+1) - \boldsymbol{x} \rangle \\
&\quad - \frac{1}{2} \| \boldsymbol{x}(k+1) - \boldsymbol{x}(k) \|_E^2
\end{aligned} \tag{C.10}$$

By the Cauchy inequality,

$$\begin{aligned}
& r_{k+1}(\boldsymbol{x}) - r_k(\boldsymbol{x}) \\
&\leq -\frac{h}{M} \langle \nabla \hbar(\boldsymbol{x}(k)), \boldsymbol{x}(k+1) - \boldsymbol{x} \rangle + \frac{1}{2} h^2
\end{aligned} \tag{C.11}$$

Thus,

$$\frac{h}{M} \langle \nabla \hbar(\boldsymbol{x}(k)), \boldsymbol{x}(k+1) - \boldsymbol{x} \rangle \leq r_k(\boldsymbol{x}) - r_{k+1}(\boldsymbol{x}) + \frac{1}{2} h^2 \tag{C.12}$$

Case II: when $k \in \{k|k \in \{0,\dots,T\}, \forall i \in \boldsymbol{C}, I_{p_{ij}}(k) \neq \emptyset\}$ Similar as **Case I**, according to Lemma 1, we have

$$\begin{aligned} & h_{k,i}\nabla_i f_l(\boldsymbol{x}(k))(x_i(k)-x) \\ \leq & (\nabla_i d_e(x(k+1)) - \nabla_i d_e(x(k+1)))(x_i - x_i(k+1)) \end{aligned} \tag{C.13}$$

Given the $\boldsymbol{\mathcal{B}}_h^e$ is with the form of (15), we have

$$\begin{aligned} & \sum_{i\in\boldsymbol{C}} h_{k,i}(t)\nabla_k\zeta_k(\boldsymbol{x}(t))(x(t)-x_k(t+1)) \\ \leq & \langle \nabla d_e(\boldsymbol{x}(k+1)) - \nabla d_e(\boldsymbol{x}), \boldsymbol{x}-\boldsymbol{x}(k+1)\rangle \end{aligned} \tag{C.14}$$

Hence,

$$\begin{aligned} & r_{k+1}(\boldsymbol{x}) - r_k(\boldsymbol{x}) \\ \leq & \sum_{i\in\boldsymbol{C}} -h_{k,i}\nabla_i\zeta_i(\boldsymbol{x}(k))(x_i(k+1)-x_i) - \\ & \frac{1}{2}\parallel \boldsymbol{x}-\boldsymbol{x}(k+1)\parallel_E^2 \\ \leq & \sum_{i\in\boldsymbol{C}} -h_{k,i}\nabla_i\zeta_i(\boldsymbol{x}(k))(x_i(k)-x_i) + \frac{1}{2}H(k)^2\parallel\nabla\zeta(\boldsymbol{x})\parallel_E^{*2} \end{aligned} \tag{C.15}$$

where $H(k) = \max_{i\in\boldsymbol{C}} h_{k,i}$, $\parallel\zeta(\boldsymbol{x})\parallel_E^{*2} = \max_{k\in\boldsymbol{C}} \parallel\zeta_k(\boldsymbol{x}(k))\parallel_E^{*2}$

Case III: when $k \in \{k|k \in \{0,\dots,T\}, \exists i,j \in \boldsymbol{C}, I_{p_i}(k) \neq \emptyset, I_{p_i}(k) = \emptyset\}$. According to Lemma 1, for each client i we have (C.13),

$$\begin{aligned} & h\nabla_i f_l(\boldsymbol{x}(k))(x_i(k+1)-x_i) \\ \leq & (\nabla_i d_e(x(k+1)) - \nabla_i d_e(x(k+1)))(x_i - x_i(k+1)) \end{aligned} \tag{C.16}$$

According to the algorithm, for $I_{p_{ij}}(k) = \emptyset$, $\nabla_i f_l(\boldsymbol{x}(k)) \leq \epsilon \to 0$, we have (C.17).

$$\begin{aligned} & r_{k+1}(\boldsymbol{x}) - r_k(\boldsymbol{x}) \\ \leq & -\left(\sum_{i\in\boldsymbol{\mathcal{B}}} h\nabla_i\hbar(\boldsymbol{x}(k))(x_i(k+1)-x_i) + \sum_{k\notin\boldsymbol{\mathcal{B}}} h_{k,i}\nabla_i f_l(\boldsymbol{x}(k))(x_i(k+1)-x_i)\right) + \frac{1}{2}h_i(k)^2\parallel\nabla\zeta(\boldsymbol{x})\parallel_E^2 \\ \leq & -\sum_{i\notin\boldsymbol{\mathcal{B}}} h_i(k)\nabla_i f_l(\boldsymbol{x}(k))(x_i(k+1)-x_i) + \frac{1}{2}h_{k,i}^2\parallel\nabla\zeta(\boldsymbol{x})\parallel_E^2 \\ \to & -\sum_{i\in\boldsymbol{C}} h_{k,i}\nabla_i f_l(\boldsymbol{x})(x_i(k+1)-x_i) + \frac{1}{2}h_{k,i}^2\parallel\nabla\zeta(\boldsymbol{x})\parallel_E^2 \end{aligned} \tag{C.17}$$

Therefore, by (C.17),

$$\begin{aligned} & \sum_{i\in\boldsymbol{C}} h_{k,i}\nabla_i f_l(\boldsymbol{x})(x_i(k+1)-x_i) \\ \leq & r_k(\boldsymbol{x}) - r_{k+1}(\boldsymbol{x}) + \frac{1}{2}H(k)^2\parallel\nabla\zeta(\boldsymbol{x}(k))\parallel_E^2 \end{aligned} \tag{C.18}$$

by Cauchy inequality, we have,

$$\begin{aligned} & \sum_{l\in\boldsymbol{\mathcal{E}}} h_{k,l}(\langle\nabla f_l(\boldsymbol{x}(k)), \boldsymbol{x}(k)-\boldsymbol{x}\rangle - f_l(\boldsymbol{x}(k))) \\ = & \sum_{i\in\boldsymbol{C}} h_{k,i}(\nabla_i f_l(\boldsymbol{x}(k))(x(k+1)-x(k)) - f_l(\boldsymbol{x}(k))) \\ \leq & r_k(\boldsymbol{x}) - r_{k+1}(\boldsymbol{x}) + \frac{\nabla\zeta(\boldsymbol{x}(k))}{2\parallel\nabla\zeta(\boldsymbol{x}(k))\parallel_E^{*2}} \\ \leq & r_k(\boldsymbol{x}) - r_{k+1}(\boldsymbol{x}) + \frac{1}{2}h^2 \end{aligned} \tag{C.19}$$

Summing up all inequalities (C.12, C.15, C.19) of $k \in \boldsymbol{\mathcal{I}}$,

$$\lambda_k^0\rho_k \leq r_0(k) + \frac{1}{2}|\boldsymbol{\mathcal{F}}|h^2 - \frac{1}{2}|\boldsymbol{\mathcal{I}}/\boldsymbol{\mathcal{F}}|h^2 \tag{C.20}$$

when $k \geq \frac{2}{h^2}\sigma$, $\rho_k \leq \Theta h$, where $\sigma = \max_{\boldsymbol{x}\in D^{|C|}} r_0(\boldsymbol{x})$, and $\Theta = \max_{k\in\boldsymbol{\mathcal{I}},l\in\boldsymbol{C}} \parallel\nabla f_l(\boldsymbol{x}(k))\parallel_E^*$. Hence, we prove the convergence of DSMDA.

□

Appendix D. Proof of Theorem 3

Proof. Recall that

$$H_i(t) = \lceil H_i(t-1) - 1 + \frac{x_i^*(t)}{v_i(t)} \rceil^+$$

by squaring the above virtual queue update, we further have,

$$\begin{aligned} & H_i^2(t) \\ \leq & H_i^2(t-1)^2 + 2H_i(t-1)\left(\frac{x_i^*(t)}{v_i(t)} - 1\right) + \left(\left(\frac{x_i^*(t)}{v_i(t)}\right)^2 - 1\right) \end{aligned} \tag{D.1}$$

Hence, by neglecting the negative terms in right side of (D.1),

$$\frac{H_i(t) - H_i(t-1)}{2} \leq B + H_i(t-1)\left(\frac{x_i^*(t)}{v_i(t)} - 1\right) \tag{D.2}$$

Let $\Delta H \triangleq (H_i(t) - H_i(t-1))/2$, by adding $\mathcal{V}p(v_i(t))$ at both sides of (D.2), we thereby,

$$\Delta H + \mathcal{V}p(v_i(t)) \leq B + \mathcal{V}p(v_i(t)) + H_i(t-1)\left(\frac{x_i^*(t)}{v_i(t)} - 1\right) \tag{D.3}$$

Since at each time slot t, according to (21), we have

$$\begin{aligned} & \mathcal{V}p(v_i(t)) + H_i(t-1)\left(\frac{x_i^*(t)}{v_i(t)} - 1\right) \\ \leq & \mathcal{V}p\left(v_i^*\right) + H_i(t-1)\left(\frac{x_i^*(t)}{v_i^*} - 1\right) \end{aligned} \tag{D.4}$$

Due to the fact that $\frac{x_i^*(t)}{v_i^*} - 1 \leq 0$, by (D.3) (D.4), we then have

$$\Delta H + \mathcal{V}p(v_i(t)) \leq B + \mathcal{V}p\left(v_i^*\right) \tag{D.5}$$

Thus, similar to Theorem 4.2 from Neely (2010), summation of all the (D.5) and neglecting the zero terms, we thereby prove the theorem. □

References

Abdullahi, I., Arif, S., Hassan, S., 2015. Survey on caching approaches in information centric networking. J. Netw. Comput. Appl. 56, 48–59.

Beck, A., Teboulle, M., 2003. Mirror descent and nonlinear projected subgradient methods for convex optimization. Oper. Res. Lett. 31 (3), 167–175.

Boyd, S., Vandenberghe, L., 2004. Convex Optimization. Cambridge University Press.

Carofiglio, G., Gallo, M., Muscariello, L., Papali, M., 2013. Multipath congestion control in content-centric networks. In: 2013 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS). IEEE, pp. 363–368.

Cao, T., Zhong, L., Xiao, H., Song, C., Yang, S., Xu, C., 2019. Credible and economic multimedia service optimization based on game theoretic in hybrid cloud networks. Trans. Emerg. Telecommun. Technol. (Wiley).

Carofiglio, G., Gallo, M., Muscariello, L., 2016. Optimal multipath congestion control and request forwarding in information-centric networks: protocol design and experimentation. Comput. Network. 110, 104–117.

Costa, F.R., da Rosa Righi, R., da Costa, C.A., Both, C.B., Cisco visual networking index: forecast and methodology, 2016-2021. http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/complete-white-paper-c11-481360.html.

Costa, F.R., da Rosa Righi, R., da Costa, C.A., Both, C.B., 2019. Nuoxus: A proactive caching model to manage multimedia content distribution on fog radio access networks. Future Generat. Comput. Syst. 93, 143–155.

El Essaili, A., Schroeder, D., Steinbach, E., Staehle, D., Shehada, M., 2015. Qoe-based traffic and resource management for adaptive http video delivery in lte. IEEE Trans. Circ. Syst. Video Technol. 25 (6), 988–1001.

Guna, J., Gerak, G., Humar, I., Song, J., Drnovek, J., Poganik, M., 2019. Influence of video content type on users' virtual reality sickness perception and physiological response. Future Generat. Comput. Syst. 91, 263–276.

Hu, H., Jin, Y., Wen, Y., Westphal, C., 2019. Orchestrating caching, transcoding and request routing for adaptive video streaming over icn. ACM Trans. Multimed Comput. Commun. Appl. (TOMM) 15 (1), 24.

Huang, W., Zhou, Y., Xie, X., Wu, D., Chen, M., Ngai, E., 2018. Buffer state is enough: simplifying the design of qoe-aware http adaptive video streaming. IEEE Trans. Broadcast. 64 (2), 590–601, https://doi.org/10.1109/TBC.2018.2789580.

Jmal, R., Simon, G., Chaari, L., 2017. Network-assisted strategy for dash over ccn. In: 2017 IEEE International Conference on Multimedia and Expo (ICME). IEEE, pp. 13–18.

Karami, A., 2015. Accpndn: adaptive congestion control protocol in named data networking by learning capacities using optimized time-lagged feedforward neural network. J. Netw. Comput. Appl. 56, 1–18.

Lederer, S., Mueller, C., Rainer, B., Timmerer, C., Hellwagner, H., 2013. Adaptive streaming over content centric networks in mobile networks using multiple links. In: 2013 IEEE International Conference on Communications Workshops (ICC). IEEE, pp. 677–681.

Lederer, S., Mueller, C., Timmerer, C., Hellwagner, H., 2014. Adaptive multimedia streaming in information-centric networks. IEEE Netw. 28 (6), 91–96.

Li, Q., Jiang, Y., Tan, Y., Xu, M., 2017. Improving the transmission control efficiency in content centric networks. Comput. Commun. 109, 76–88.

Liu, Y., Lee, J.Y., 2016. A unified framework for automatic quality-of-experience optimization in mobile video streaming. In: IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications. IEEE, pp. 1–9.

Liu, Z., Wei, Y., 2016. Hop-by-hop adaptive video streaming in content centric network. In: 2016 IEEE International Conference on Communications (ICC). IEEE, pp. 1–7.

Liu, T., Zhang, M., Zhu, J., Zheng, R., Liu, R., Wu, Q., 2019. Accp: adaptive congestion control protocol in named data networking based on deep learning. Neural Comput. Appl 31, 4675–4683.

ndnsim in ns-3, http://ndnsim.net/intro.html.

Neely, M.J., 2010. Stochastic network optimization with application to communication and queueing systems. Synth. Lect. Commun. Netw. 3 (1), 1–211.

Nesterov, Y., 2009. Primal-dual subgradient methods for convex problems. General Inf. Mathematical Programming 120 (1), 221–259.

Rainer, B., Posch, D., Hellwagner, H., 2016. Investigating the performance of pull-based dynamic adaptive streaming in ndn. IEEE J. Sel. Area. Commun. 34 (8), 2130–2140.

Rainer, B., Petscharnig, S., Timmerer, C., Hellwagner, H., 2017. Statistically indifferent quality variation: an approach for reducing multimedia distribution cost for adaptive video streaming services. IEEE Trans. Multimed. 19 (4), 849–860.
Rashid, Z., Meli-Segu, J., Pous, R., Peig, E., 2017. Using augmented reality and internet of things to improve accessibility of people with motor disabilities in the context of smart cities. Future Generat. Comput. Syst. 76, 248–261.
Reichl, P., Tuffin, B., Schatz, R., 2013. Logarithmic laws in service quality perception: where microeconomics meets psychophysics and quality of experience. Telecommun. Syst. 52 (2), 587–600.
Samain, J., Carofiglio, G., Muscariello, L., Papalini, M., Sardara, M., Tortelli, M., Rossi, D., 2017. Dynamic adaptive video streaming: towards a systematic comparison of icn and tcp/ip. IEEE Trans. Multimed. 19 (10), 2166–2181.
Stais, C., Xylomenos, G., Voulimeneas, A., 2015. A reliable multicast transport protocol for information-centric networks. J. Netw. Comput. Appl. 50, 92–100.
Tang, Y., Guo, K., Ma, J., Shen, Y., Chi, T., 2019. A smart caching mechanism for mobile multimedia in information centric networking with edge computing. Future Generat. Comput. Syst. 91, 590–600.
Xu, C., Jia, S., Zhong, L., Muntean, G., 2015a. Socially aware mobile peer-to-peer communications for community multimedia streaming services. IEEE Commun. Mag. 53 (10), 150–156, https://doi.org/10.1109/MCOM.2015.7295477.
Xu, C., Jia, S., Wang, M., Zhong, L., Zhang, H., Muntean, G.-M., 2015b. Performance-aware mobile community-based vod streaming over vehicular ad hoc networks. IEEE Trans. Veh. Technol. 64 (3), 1201–1217.
Xu, C., Zhang, P., Jia, S., Wang, M., Muntean, G., 2017. Video streaming in content-centric mobile networks: challenges and solutions. IEEE Wirel. Commun. 24 (5), 157–165, https://doi.org/10.1109/MWC.2017.1600219WC.
Xu, C., Wang, M., Chen, X., Zhong, L., Grieco, L.A., 2018. Optimal information centric caching in 5g device-to-device communications. IEEE Trans. Mobile Comput. 17 (9), 2114–2126, https://doi.org/10.1109/TMC.2018.2794970.
Zhang, L., Afanasyev, A., Burke, J., Jacobson, V., Crowley, P., Papadopoulos, C., Wang, L., Zhang, B., et al., 2014. Named data networking. ACM SIGCOMM Comput. Commun. Rev. 44 (3), 66–73.
Zhang, F., Zhang, Y., Reznik, A., Liu, H., Qian, C., Xu, C., 2015. Providing explicit congestion control and multi-homing support for content-centric networking transport. Comput. Commun. 69, 69–78.

Mu Wang received his M.S. degree in computer technology from Beijing University of Posts and Telecommunications (BUPT) in 2015. He is currently working towards the Ph.D degree with the Institute of Network Technology, BUPT. His research interests include information centric networking, wireless communications, and multimedia sharing over wireless networks.

Changqiao Xu received the Ph.D. degree from the Institute of Software, ISCAS in 2009. He is now a Professor with the State Key Laboratory of Networking and Switching Technology at BUPT. He has published over 160 technical papers in prestigious international journals and conferences. His research interests include wireless networking, multimedia communications, and future internet technology. He serves as Editor-in-Chief of Transactions on Emerging Telecommunications Technologies(Wiley). He is Senior Member of IEEE.

Xingyan Chen received the BE degree in Applied Physics from the College of Science, Beijing University of Posts and Telecommunications, in 2016. He is currently working toward the master degree under Prof. Xu at the Next generation Internet Lab. His research interests include information dissemination and content center network.

Lujie Zhong received the Ph.D. degree from the Institute of Computing Technology, Chinese Academy of Sciences, Beijing, China, in 2013. She is an Associate Professor with the Information Engineering College, Capital Normal University, Beijing. Her research interests include communication networks, computer system and architecture, mobile Internet technology.

Gabriel-Miro Muntean received his Ph.D. degree from Dublin City University (DCU), Ireland in 2003. He is Associate Professor with the School of Electronic Engineering at DCU, co-Director of the DCU Performance Engineering Laboratory. His research interests include quality-oriented and performance-related issues of adaptive multimedia delivery, performance of wired and wireless communications. He has published over 300 papers in prestigious international journals and conferences, has authored three books and 16 book chapters. He is Associate Editor for the IEEE Transactions on Broadcasting and Editor for the IEEE Communications Surveys and Tutorials. He is a Senior Member of IEEE.