Design QoS-Aware Multi-Path Provisioning Strategies for Efficient Cloud-Assisted SVC Video Streaming to Heterogeneous Clients

Zuqing Zhu, *Senior Member, IEEE*, Suoheng Li, *Student Member, IEEE*, and Xiaoliang Chen

***Abstract*—We layout a network infrastructure that leverages the storage and computing power of a cloud residing in the core for collecting network status and computing multi-path scalable video coding (SVC) streaming provisioning strategies. Therefore, in addition to its conventional tasks in the application layer, the cloud also gets involved in the network layer for the optimization of routing and forwarding. We call this scheme as cloud-assisted SVC streaming, and use it to further improve the performance of SVC streaming by using close cooperation between cloud and network. Compared to source-routing based provisioning, the cloud-assisted scheme can provide more cost-effective provisioning strategies by utilizing better knowledge of network environment together with more powerful computation power. We then propose several multi-path provisioning algorithms for cloud-assisted SVC streaming in heterogeneous networks. To the best of our knowledge, these are the first proposals to work on the problem of adaptive multi-path SVC streaming under the bandwidth, delay and differential delay constraints. Our design of the provisioning algorithms starts from an approach that is based on *Max Flow* and an *Auxiliary Graph*. Several extensions are then made based on this approach to address the situations such as provisioning from multiple sources and provisioning in dynamic network environments with rapid background traffic fluctuations. Simulations in both static and dynamic network environments show that the proposed algorithms can achieve effective performance improvements in terms of request blocking probability, bandwidth utilization, packet delay, packet loss rate, and video playback quality.**

***Index Terms*—Cloud-assisted video streaming, label switching, multi-path provisioning, scalable video coding.**

I. Introduction

OVER the last decade, various network applications have emerged on the Internet and pushed the network traffic to grow at an annual rate of more than 30% [1]. In order to accommodate the increase of applications and bandwidth demands, the Internet service needs to migrate from the best-effort model to an integrated one that considers different quality-of-service (QoS) requirements from data, voice and video applications [2]. However, due to the heterogeneity of networks and clients, fluctuations of bandwidth, and many other challenges, service providers still have difficulties to deliver QoS-guaranteed service to anywhere, at any time, especially when they target for high-bandwidth applications such as video-on-demand and teleconference. Therefore, it has been a consensus that the routing and forwarding protocols have to be revisited for supporting arbitrary set of end-to-end QoS requirements [2]–[7].

In the early days of Internet, routing and forwarding primarily focused on how to deliver data packets over a single path with the shortest length, and the routing tables were built based on a single cost metric, such as hop-count or propagation delay. Multimedia applications, such as video streaming, usually depend on a combination of QoS parameters (e.g., delay, bandwidth, differential delay and etc.) to offer good user experience. Very often, a single routing path in today's bandwidth-limited Internet cannot satisfy these QoS parameters, especially the high bandwidth requirement [3], [5]. Research works have been trying to resolve this issue from both the networking and video coding perspectives. From the networking point of view, path diversity has been proposed as an effective mechanism that aggregates network resources efficiently [2], [8], [9]. Delivering media traffic over multiple paths usually provides a more cost-effective solution than setting up a single path that can support the total streaming capacity [5]. Moreover, path diversity offers the advantages of better protection and faster restoration. As shown in Fig. 1, the streaming client can collect aggregated media traffic from multiple transmission flows for improved user experience. This scheme becomes extremely useful when the client is in a wireless system, where some of the connections are more bandwidth-limited and unstable compared to the wired ones [2]. On the video coding side, multiple description coding (MDC) and scalable video coding (SVC) have been designed to generate several substreams such that different levels of reconstruction qualities can be achieved with different subsets of them [10], [11]. This property correlates well with multi-path provisioning, and their combination can dramatically improve the QoS of video streaming [12].

A. Scalable Video Coding (SVC)

SVC [11], [13] is the scalable extension of H.264/AVC that supports spatial, temporal and quality scalability. Using a layered video codec, SVC encodes a video as one base layer (BL) and several enhancement layers (ELs). Each of these layers can be transmitted and received independently. With the BL, a streaming client can recover the video with low quality,

Manuscript received February 16, 2012; revised June 27, 2012; accepted November 13, 2012. Date of publication January 11, 2013; date of current version May 13, 2013. This work was supported by the Program for New Century Excellent Talents in University (NCET) under Project NCET-11-0884. The associate editor coordinating the review of this manuscript and approving it for publication was Joel Rodrigues.

The authors are with the School of Information Science and Technology, University of Science and Technology of China, Hefei, Anhui 230027, China (e-mail: zqzhu@ieee.org; lzzitc@mail.ustc.edu.cn; arabus@mail.ustc.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMM.2013.2238908

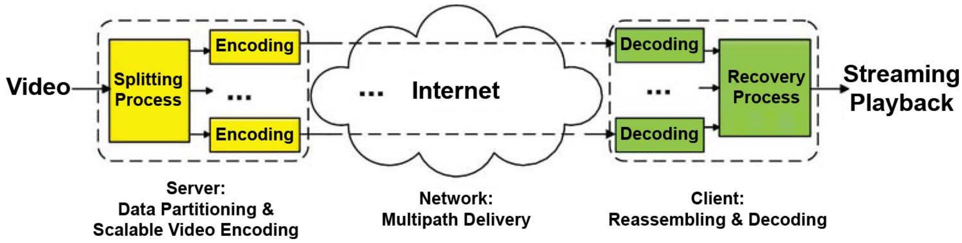


Fig. 1. Video delivery framework by integrating SVC and multi-path provisioning.

while the more ELs it collects, the better the playback quality will be. This property, together with a multi-path provisioning strategy, provides an intuitive and promising solution to the problem of how to stream a video to heterogeneous clients with various QoS requirements efficiently [14].

B. Related Works

SVC provides service providers a remarkable opportunity to stream videos to heterogeneous clients cost-effectively, especially in the cases where multi-path routing are integrated. To maximize the benefits associated with this integrated scheme, previous works have investigated a few detailed subjects, such as multi-path routing under multiple constraints [3], [5], [15], media-specific path selection [16], media rate allocation and packet scheduling [7], [17], [18], peer-to-peer and multi-source streaming [6], [10], [19], protection and restoration with path switching [20], and data partition and video coding frameworks [21], [34]. From the system perspective, multi-path provisioning strategy (e.g., routing path selection and rate allocation) plays an important role in the optimization of network resource allocation for efficient SVC video streaming, and it has yet been fully investigated [2].

It is well-known that routing under multiple constraints is an NP-hard problem in general [2]. Chen *et al.* proposed maximum flow based multi-path routing algorithms for both unicast and multicast, and utilized them to satisfy the bandwidth requirement while minimizing the start-up delay [3]. Later work in [15] had proposed a fully polynomial time approximation routing scheme for better delay performance. Ma *et al.* proposed multi-path selection scheme that considered multiple network metrics such as bandwidth, delay and packet loss [24]. However, none of these works specifically considered the differential delay or jitter when making routing decision. When the SVC sub-streams are delivered over multiple paths, they may experience different delay and thus reach the clients at jittered time points. The consequent differential delay may require a significant increase of memory in the client's playback device and degrade the streaming quality [5]. Therefore, it has to be well-controlled to save network element cost and to improve QoS and quality-of-experience (QoE). Recently, Huang *et al.* and Zhang *et al.* [5], [25] have investigated multi-path routing with differential delay constraint for network restoration and highly-available provisioning. Nevertheless, the problem of adaptive multi-path routing for SVC streaming with differential delay constraint is still under-explored, even though a few multi-path SVC streaming frameworks [26], [27] have been proposed based on an overlay network infrastructure. Moreover, since multi-path routing can make the routing and forwarding protocols more complicated, the additional protocol overheads from the overlay-based approaches in [26], [27] may further limit the system's efficiency and scalability. While multi-path routing is also feasible in the network layer [9], it is desired to realize the provisioning strategies with such a label-switching mechanism (i.e., multi-protocol label switching (MPLS) [28]), for minimizing protocol overheads.

Nowadays, wireless technology has emerged as a key enabler for providing flexible network support to various media applications. SVC based single-path video streaming in wireless IP networks has been investigated in [14]. Chen *et al.* has proposed a perception-aware single-path streaming strategy that could transmit multiple SVC videos over IEEE 802.11e wireless-LAN with high-efficiency [29]. Multi-path SVC streaming has been investigated for wireless sensor networks [30], wireless ad hoc networks [31], and indoor 60 GHz radio networks [32]. Compared to wired counterparts, wireless connections are usually more dynamic and unstable.

C. Contributions of Our Work

In this paper, we first layout a network infrastructure that leverages the storage and computing power of a cloud residing in the core for collecting network status and computing multi-path SVC streaming provisioning strategies. Therefore, in addition to its conventional tasks in the application layer, our cloud also gets involved in the network layer optimization of routing and forwarding. We call this scheme as cloud-assisted SVC streaming, and use it to further improve the performance of provisioning. Recently, we have proposed to perform multi-path SVC streaming based on a K-shortest-path routing algorithm, and the rate-allocation was done by mapping the SVC layers to routing paths one-by-one [33]. However, since this approach treated the routing of each SVC layer as an atomic operation, it only has limited flexibility for providing SVC streaming to heterogeneous clients. Moreover, the K-shortest-path routing may bring difficulties to rate-allocation when two or more routing paths share links. In this paper, to extend our previous work in [33], we propose several multi-path provisioning algorithms for cloud-assisted SVC streaming to heterogeneous clients. To the best of our knowledge, these are the first proposals to work on the problem of adaptive multi-path routing for SVC streaming with bandwidth, delay and differential delay constraints. The proposed multi-path provisioning strategies are able to deliver SVC video to heterogeneous clients through large-scale networks, with joint optimization of multiple metrics. Based on the nature of SVC encoding, we define a bandwidth allocation granularity and then design the first provisioning algorithm based on *Max Flow* and an *Auxiliary Graph*. We then prove that with

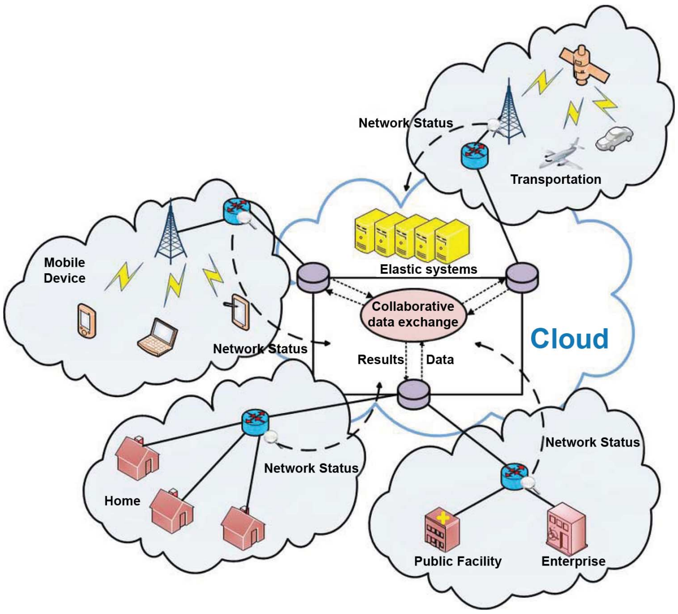


Fig. 2. Cloud-assisted video streaming to heterogeneous clients.

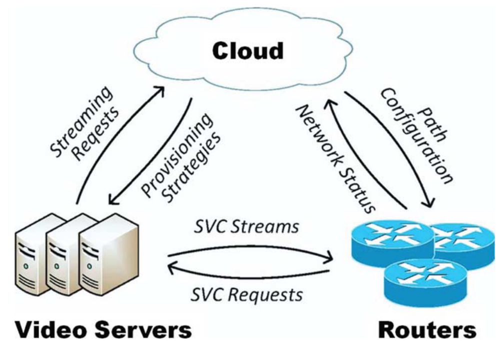


Fig. 3. Communications between network elements for setting up and maintaining SVC streaming sessions with cloud assistance.

minor modifications, the proposed algorithm can also solve the problem of provisioning from multiple sources. To further improve the efficiency of the provisioning algorithm and to make it suitable for dynamic network environments with rapid background traffic fluctuations, we develop an extended algorithm that uses path stacks to allocate just-enough bandwidth to SVC streaming requests. The simulations of the proposed algorithms in both static and dynamic network environments show that they can achieve effective performance improvements in terms of blocking probability, bandwidth utilization, packet delay, packet loss rate, and video play-back quality.

The rest of the paper is organized as follows. Section II layouts the network infrastructure and protocols we will discuss in this paper. Section III formulates the problem of multi-path provisioning under bandwidth, delay and differential delay constraints. The QoS-aware multi-path provisioning algorithm that is based on *Max Flow* and *Auxiliary Graph* is discussed in Section IV. Section V describes several algorithm extensions, and Section VI shows the simulation setups and results for evaluating the proposed algorithms. Finally, Section VII summarizes the paper.

II. Cloud-Assisted SVC Video Streaming in Heterogeneous Networks

Fig. 2 shows the network infrastructure we discuss in this paper. The cloud resides in the core of the service provider's network, and heterogeneous clients can connect to the streaming services through various access scenarios. It is well known that one of the most significant benefits of cloud is its elasticity such that storage resources and computing power can be allocated to tasks in a dynamic and cost-effective manner. We assume that the elastic systems of the cloud has the capability of collecting both video resources and network status proactively. To adapt to the heterogeneity of clients, the cloud encodes video contents with SVC and stores the results in one or more appropriate servers. As the video coding frameworks have already been investigated intensively [21]–[23], we will not discuss them. By leveraging its numerous computation and storage resources, the cloud collects and analyzes network status to compute an optimized QoS-aware multi-path provisioning strategy for each request. With a centralized source-routing mechanism, the SVC video streaming service is then set up in the network-layer using multi-path label-switching [9]. Specifically, the cloud assigns specific labels to the content servers for encoding SVC packets, and instructs the routers to forward the packets on multiple label-switching paths. When network status changes, the cloud can re-calculate the paths and update them. As the path computation is done by the cloud but not the routers, the re-design of routers can be avoided and significant protocol overheads can be saved.

Fig. 3 shows communications between network elements for setting up and maintaining SVC streaming sessions with cloud assistance. Here, we assume that the cloud and the network are owned by the same operator. During operation time, the cloud collects network status, such as delay and available bandwidth of links, which is disseminated with an existing routing protocol (e.g., OSPF-TE). Note that since advertising network status information too frequently may cause significant overheads and limit network scalability, we need to make a tradeoff between precise networking monitoring and low control-plane overheads in the designs of algorithms and protocols. When a video server receives a streaming request, it forwards the request to the cloud for suggestions on provisioning strategy. The cloud then computes a multi-path provisioning strategy for the request, allocates labels to streaming paths, and returns the strategy to one or multiple servers. When a server receives the provisioning strategy, it starts to pack the video content with the labels provided by the cloud. Meanwhile, the cloud configures the routers along the label switching paths (LSPs) with MPLS messages. When all these tasks are done, a streaming session is setup. The cloud then keeps monitoring the network and performs necessary LSP adjustments by reconfiguring the routers, when there are changes on the network status. It can be seen that one of the major technical difficulties of this cloud-assisted multi-path streaming scheme is *How to efficiently compute multiple routing paths and allocate bandwidth to them?* We will address this problem in the following sessions and verify our proposals with numerical simulations.

III. Problem Formulation

We consider the network under investigation as a weighted directed graph $G(V, E, B, D)$, where V is the node set, E is the network link set. $b(e) \geq 0 \in B$ and $d(e) > 0 \in D$ represent the bandwidth and delay on the link $e = (u, v) \in E$ from u to v, respectively. Note that the delay we consider in this work is the combination of propagation delay and node processing delay. We do not consider the delay caused by identifying, searching and locating resources in the cloud. We define $R_{s,d}$ as the set of feasible paths from s to d, and $r_{s,d}^{(k)} \in R_{s,d}$ is the k-th path in the set. Then, the available bandwidth of $r_{s,d}^{(k)}$ is

$$BW(r_{s,d}^{(k)}) = \min(b(e)),\ \forall e \in r_{s,d}^{(k)} \quad (1)$$

and the delay of it is

$$DL(r_{s,d}^{(k)}) = \sum_{e \in r_{s,d}^{(k)}} d(e). \quad (2)$$

The video contents are identified with a global ID i. For the i-th content, $c_{i,j}$ represents the minimal bandwidth requirement of the j-th SVC layer. Therefore, $c_{i,1}$ is the bandwidth requirement of the base layer (BL), and $c_{i,j} (j > 1)$ is the requirement of the $(j-1)$-th enhancement layer (EL).

Definition 1 (Bandwidth Allocation Granularity): For a content i, we say g_i is its bandwidth allocation granularity such that:

$$c_{i,j} \leq N \cdot g_i,\ N \in \mathbb{Z}^+,\ \forall j. \quad (3)$$

Note that choosing a proper g_i can reduce the complexity of the multi-path provisioning. When g_i is determined, we have

$$N_i = \sum_j \left\lceil \frac{c_{i,j}}{g_i} \right\rceil \quad (4)$$

as the number of sub-layers we need to assign to multiple paths.

Definition 2 (Path Selection): For the streaming request of content i from s to d, the path selection is represented by the flag $f_{i,n}^{(k)}$ when the routing path set $R_{s,d}$ is known. If a sub-layer $n(n = 1, \ldots, N_i)$ of content i will be delivered over $r_{s,d}^{(k)} \in R_{s,d}$, $f_{i,n}^{(k)}$ is 1, otherwise, it is 0.

Definition 3 (Differential Delay): Let $R_{s,d,i}$ denotes the set of selected paths, the differential delay of the path selection is defined as:

$$DD(R_{s,d,i}) = \max(DL(r_{s,d}^{(k)})) - \min(DL(r_{s,d}^{(k)})), \times \{k : r_{s,d}^{(k)} \in R_{s,d,i}\}. \quad (5)$$

With the definitions above, the multi-path provisioning becomes an optimization problem: Given $G(V, E, B, D)$, for each streaming request from s to d for content i, the cloud needs to find a delivery strategy that can maximize deliverable SVC layers M with $R_{s,d,i}$ under multiple constraints. Then, the mathematical formulations of the problem is:

Objective:

$$\max_{R_{s,d,i}, f_{i,n}^k} \quad M.$$

Subject to:

Bandwidth Constraint:

$$\sum_{n=1}^{N_i} g_i f_{i,n}^{(k)} \leq BW(r_{s,d}^{(k)}),\ \forall k. \quad (6)$$

Delay Constraint:

$$\max(DL(r_{s,d}^{(k)})) \leq D_{\max},\ \{k : r_{s,d}^{(k)} \in R_{s,d,i}\}. \quad (7)$$

Differential Delay Constraint:

$$DD(R_{s,d,i}) \leq J_{\max}. \quad (8)$$

Decodable Constraint:

$$\sum_{k=1}^{|R_{s,d}|} f_{i,n}^{(k)} = 1,\ \left\{ n : n \leq \sum_{j=1}^{M} \left\lceil \frac{c_{i,j}}{g_i} \right\rceil \right\} \quad (9)$$

where $D_{\max}$ denotes the maximum start-up delay, and $J_{\max}$ denotes the maximum differential delay.

IV. QoS-Aware Multi-Path Provisioning Algorithm

We decompose the optimization formulated in the previous section into three steps and develop a multi-path heuristic to solve it. Algorithm 1 shows the overall algorithm, while Algorithms 2 and 3 show the details of "Phase II: Path Computation and Selection" and "Phase III: Bandwidth Allocation" of Algorithm 1. For the graph $G(V, E, B, D)$ and the s-d pair of a streaming request, we first find the maximum aggregated bandwidth from s to d by computing the max-flow and then transform the graph to the max-flow graph $G'(V', E', B', D')$ [3]. The deliverable streaming capacity from s to d is upper-bounded by the bandwidth of the max-flow $BW_{\max}$.

Algorithm 1 QoS-Aware Multi-Path Provisioning Algorithm

Inputs: $G(V, E, B, D)$; $c_{i,j}$; s; d; g_i; $D_{\max}$; $J_{\max}$.

Output: $f_{i,n}^{(k)}$; $R_{s,d,i}$.

{Phase I: Graph Transformation}

1:compute the max-flow from s to d;

2:combine all augmenting paths to form G';

{Phase II: Path Computation and Selection}

3:compute multiple paths in G' to get $R_{s,d}$;

4:map paths in $R_{s,d}$ as vertices in auxiliary graph G'';

5:connect vertices in G'' based on $J_{\max}$;

6:find the maximum clique in G'';

7:select paths as the vertices in the maximum clique;

{Phase III: Bandwidth Allocation}

8:allocate bandwidth based on g_i to the selected paths;

9:output $f_{i,n}^{(k)}$ and $R_{s,d,i}$;

Algorithm 2 Path Computation and Selection Algorithm

$R_{s,d} \leftarrow \varnothing$;
$delay = 0$;
$k = 1$;
while $delay \leq D_{\max}$ **do**
  $r_{s,d}^{(k)} = Dijkstra(G', s, d)$;
  $delay = DL(r_{s,d}^{(k)})$;
  if $delay \leq D_{\max}$ **then**
    $R_{s,d} \leftarrow r_{s,d}^{(k)}$;
    if $BW(r_{s,d}^{(k)}) \geq g_i$ **then**
      insert $\lceil BW(r_{s,d}^{(k)})/g_i \rceil$ vertices into G'';
    end if
    $k = k + 1$;
  end if
  modify G' by subtracting $BW(r_{s,d}^{(k)})$ from the bandwidth of each link along $r_{s,d}^{(k)}$;
  remove all links in G' that have 0 bandwidth;
end while
connect two vertices in G'' if the paths they represent have a differential delay $DD \leq J_{\max}$;
find the maximum clique in G'';
combine all vertices in the maximum clique as $R_{s,d,i}$;

Algorithm 3 Bandwidth Allocation Algorithm

change the bandwidth of $r_{s,d}^{(k)}$ in $R_{s,d,i}$ to $\lceil BW(r_{s,d}^{(k)})/g_i \rceil$;
for all SVC layers of content i from the BL **do**
  $N_j = \lceil c_{i,j}/g_i \rceil$;
  if allocate N_j to shortest paths in $R_{s,d,i}$ is successful **then**
    turn on all related $f_{i,n}^{(k)}$;
    subtract allocated bandwidth from the paths in $R_{s,d,i}$;
    remove all paths with 0 bandwidth;
  else
    break;
  end if
end for

In the second phase, we compute multiple routing paths to get $R_{s,d}$ and use an auxiliary graph based mechanism to select paths that can satisfy the QoS requirements. Algorithm 2 illustrates the details of path computation and selection. The paths are computed with a shortest path search algorithm, such as the famous Dijkstra algorithm, in iterations. In each iteration, a path $r_{s,d}^{(k)}$ is first validated to make sure $DL(r_{s,d}^{(k)}) \leq D_{\max}$ and $BW(r_{s,d}^{(k)}) \geq g_i$, for satisfying the QoS requirements on the start-up delay and bandwidth. If the path is valid, $\lceil BW(r_{s,d}^{(k)})/g_i \rceil$ vertices are inserted into the auxiliary graph G''. The bandwidth that is associated with these paths is then removed from G'. After inserting vertices into the auxiliary graph G'', we connect two vertices, if the differential delay between two paths can satisfy the QoS requirement on differential delay. A clique of G'' is a complete subgraph in it, i.e., sets of vertices that any two of them are connected. Therefore, by searching for the maximum clique in G'', we obtain $R_{s,d,i}$ from $R_{s,d}$, which can maximize the streaming bandwidth under both the delay and differential delay constraints. Finally, we allocate bandwidth based on g_i to the selected paths as illustrated in Algorithm 3. Figs. 4–5 show an intuitive example of our multi-path provisioning algorithm.

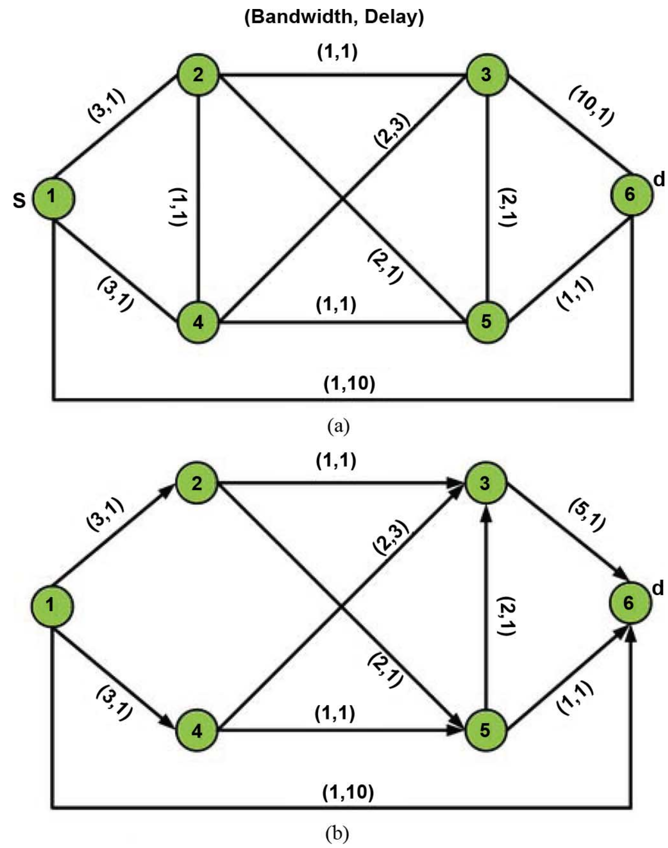


Fig. 4. Transform (a) a network topology $G(V, E, B, D)$ to (b) a max-flow topology $G'(V', E', B', D')$ (a) Network Topology G(V,E,B,D) (b) Max-Flow Topology G'(V',E',B',D').

Algorithm Extensions

A. Multi-Path Provisioning From Multiple Sources

In the previous section, we develop a multi-path provisioning scheme for SVC streaming from a single source. However, in practical cloud-assisted SVC streaming systems, the multi-source scenario can be used to improve streaming efficiency [6], [10], [19]. As shown in Fig. 6, this multi-source provisioning problem can be easily transformed to a single-source one by adding a virtual source s_0 into $G(V, E, B, D)$. Specifically, before *Phase I* in Algorithm 1, we add s_0 and connect it to all sources $s_1, \ldots, s_m$ using links e that $b(e) = +\infty$ and $d(e) = 0$.

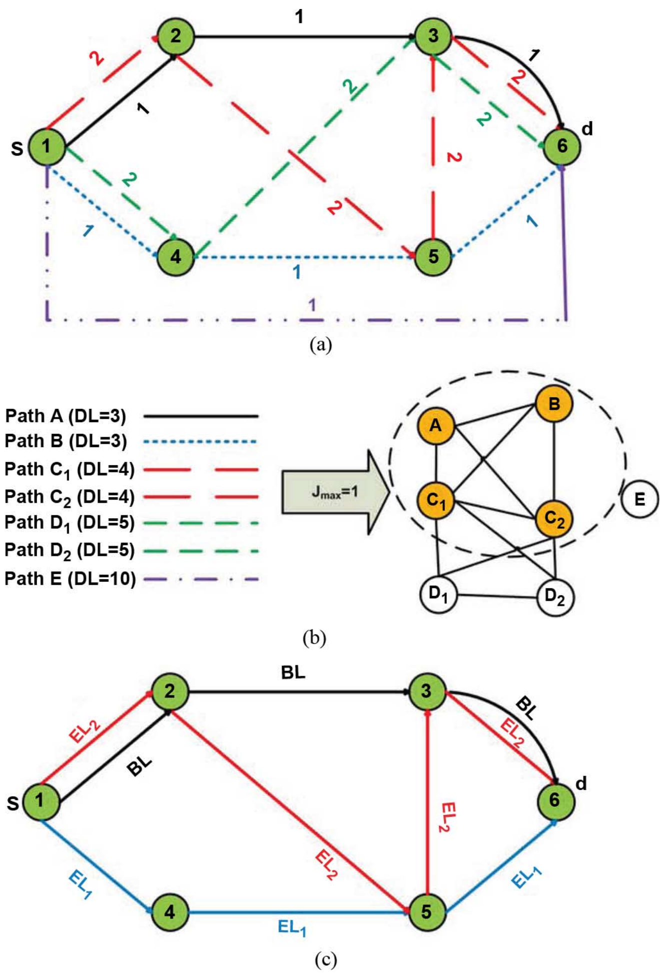


Fig. 5. (a) Path computation based on $G'(V', E', B', D')$; (b) construct the auxiliary graph G'' based on $J_{\max}$; (c) bandwidth allocation for a 3-layer SVC streaming (a) Path Computation Results from G'(V',E',B',D') (b) Construct the Auxiliary Graph G" and Find the Maximum Clique (c) Multipath Provisioning Scheme for a SVC Streaming with 3 Layers, BL $(\mathrm{BW}=1)$ and EL_1 $(\mathrm{BW}=1)$, EL_2 $(\mathrm{BW}=2)$.

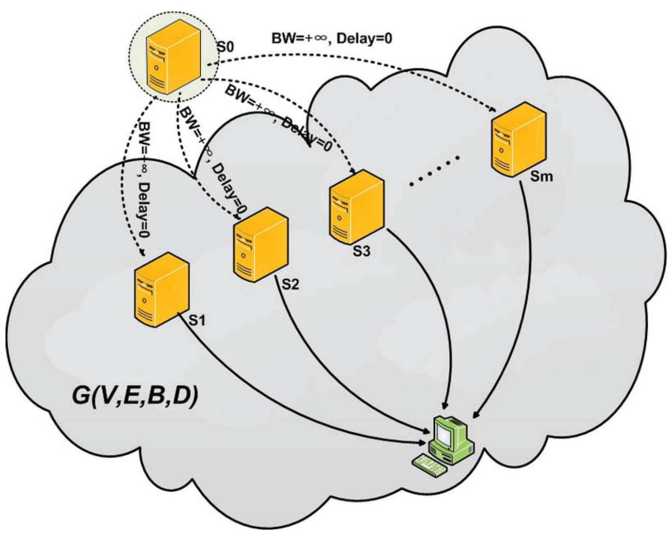


Fig. 6. Transformation of a multi-source streaming problem to a single-source one.

We define this modified graph as $G^*(V^*, E^*, B^*, D^*)$. By inputting G^* instead of G into Algorithm 1, we can obtain the solution of QoS-aware multi-path provisioning from multiple sources.

B. Multi-Path Provisioning Using Path Stacks

The max-flow based multi-path heuristic may lead to a semi-optimal solution for certain network topologies [15]. Moreover, when the streaming bandwidth requirement $\sum_j c_{i,j}$ is much less than the max-flow bandwidth $BW_{\max}$, a lot of computations may become unnecessary and the performance of dynamic provisioning may be affected. To overcome these drawbacks, we develop an extended algorithm that uses path stacks to allocate just-enough bandwidth to a SVC streaming request.

Algorithm 4 and Algorithm 5 illustrate the path selection procedures for a sub-layer n with bandwidth g_i, from s to d. In *Phase I*, we remove links whose available bandwidth is insufficient from $G(V, E, B, D)$ by setting their delays $d(e)$ to $+\infty$. The shortest paths from all nodes in V (except d) to d are then calculated by applying the Dijkstra algorithm. We push s in a path stack $HopList$, calculate the lower bound $DL|_{\min}$ of the path delay, and enqueue a structure that contains the combination of $DL|_{\min}$, total delay $delay = DL(r^{(1)}_{s,s}) = 0$ from s, and the current $HopList$ in a priority queue Q. *Phase II* searches for K shortest loop-less paths and stores them in set $R_{s,d}$ as candidates of the path selection in *Phase III*. For each intermediate node v when moving from s to d, we add the minimum delay from v to d to path's current delay $DL(s, v)$ as the new lower bound $\Delta DL'|_{\min}$ of path delay. The $(DL_{i,n})_{\max}, n = 1, \ldots, N_i$ is the upper bound of the routing path delay for delivering sub-layer n, obtained by considering $D_{\max}$, $J_{\max}$, and current path selections. In *Phase III*, when the candidate paths are obtained, we select the one with the largest available bandwidth for delivering sub-layer n. After assigning the routing path, we update the available bandwidth B in $G(V, E, B, D)$ accordingly.

Algorithm 4 Stack-Based Path Selection Algorithm for a Sublayer—Part I

Input: $G(V, E, B, D)$, $D_{\max}$, $J_{\max}$, s, d, g_i, n, K

Output: $Hoplist$, $(DL_{i,n})_{\max}$, $G(V, E, B, D)$

{Phase I: Pre-Computation}

```
for all e ∈ E do
    if b(e) < g_i then
        d(e) = +∞;
    end if
end for
for all u ∈ V, u ≠ d do
    {r^(1)_{u,d}} = Dijkstra(G, u, d);
end for
DL|_min = DL(r^(1)_{s,d});
Push(HopList, s);
Q.Enqueue({DL|_min, 0, HopList});
calculate (DL_{i,n})_max based on D_max, J_max, and path
    selection of previous sub-layers;
```

Algorithm 5 Stack-Based Path Selection Algorithm for a Sublayer—Part II

Input: $G(V, E, B, D)$, $D_{\max}$, $J_{\max}$, s, d, g_i, n, K

Output: $Hoplist$, $(DL_{i,n})_{\max}$, $G(V, E, B, D)$

{Phase II: Path Computation}

$R_{s,d} \leftarrow \varnothing$;
while $Q \neq \varnothing$ **do**
  $\{DL|_{\min}, delay, HopList\} = Q.Dequeue()$ with smallest $DL|_{\min}$;
  $u = Top(HopList)$;
  for each neighbor v of u that does not in $HopList$ **do**
    $HopList' = HopList$;
    $Push(HopList', v)$;
    $delay' = delay + d(e(u, v))$;
    $DL'|_{\min} = delay' + DL(r_{v,d}^{(1)})$;
    if $v = d$ **then**
      $R_{s,d}.Enqueue(\{delay', HopList'\})$;
      if $Size(R_{s,d}) = K$ then
        go to Phase III;
      end if
    else if $DL'|_{\min} < (DL_{i,n})_{\max}$ **then**
      $Q.Enqueue(\{DL'|_{\min}, delay', HopList'\})$;
    end if
  end for
end while

{Phase III: Path Selection}

if $|R_{s,d}| = 0$ **then**
  return $\varnothing$;
else
  select $HopList$ in $R_{s,d}$ with the maximum bandwidth;
  update available bandwidth matrix B in G;
  update $(DL_{i,n})_{\max}$ if necessary;
  return $HopList$, $(DL_{i,n})_{\max}$, and $G(V, E, B, D)$;
end if

By applying Algorithms 4 and 5 to each SVC sub-layer, Algorithm 6 shows the overall provisioning procedures for serving a SVC streaming request from s to d for content i. The cloud collects the available bandwidth B of G periodically and allocates routing paths for each sub-layer of the request based on Algorithm 4 and Algorithm 5. When there is no path can be found or its QoS requirement has been reached, we stop calculating for more routing paths. Note that when the network status has been changed significantly, for example a congestion happens, re-calculation of the routing paths can be triggered to avoid performance degradation.

Algorithm 6 Overall Multi-Path Provisioning Algorithm

get the latest B in $G(V, E, B, D)$;
for each layer j starting from BL **do**
  $flag = 1$;
  for all sub-layers of this layer **do**
    apply Algorithms 4 & 5 to the sub-layer;
    if Algorithm 1 returns $\varnothing$ **then**
      $flag = 0$;
      break;
    end if
  end for
  if $flag = 0$ **then**
    revert all the path selections for the current layer j;
    break;
  else
    commit all the path selections for the current layer j;
    if QoS requirement has been reached **then**
      break;
    end if
  end if
end for

V. Performance Evaluations

In this section, we design simulations to evaluate the proposed provisioning algorithms. We first simulate a set of random requests on a randomly generated topology to evaluate the performance of the multi-path provisioning algorithm based on *Max Flow* and *Auxiliary Graph* (Algorithm 1) in a static network without background traffic variations. It can be seen that the proposed algorithm can effectively reduce the blocking probability of the streaming system. To further demonstrate the idea of cloud-assisted SVC streaming with multi-path routing, we design a simulation scheme with a dynamic network with background traffic variations and evaluate the end-to-end QoS (e.g., bandwidth, packet loss, and delay) of the multi-path provisioning algorithm using path stacks (Algorithm 6) in a 16-node grid network. It can be seen that the multi-path provisioning effectively improves the QoS performance of SVC streaming.

TABLE I
SIMULATION PARAMETERS OF STATIC NETWORK.

Number of nodes in topology	30
Number of links in topology	329
Delay on links in topology	1 - 50 units
Bandwidth on links in topology	10 - 90 Mbps
Bandwidth of SVC streaming requests	5 - 40 Mbps
Differential delay threshold	15 units
Number of request sets simulated for each data point for statistical accuracy	64

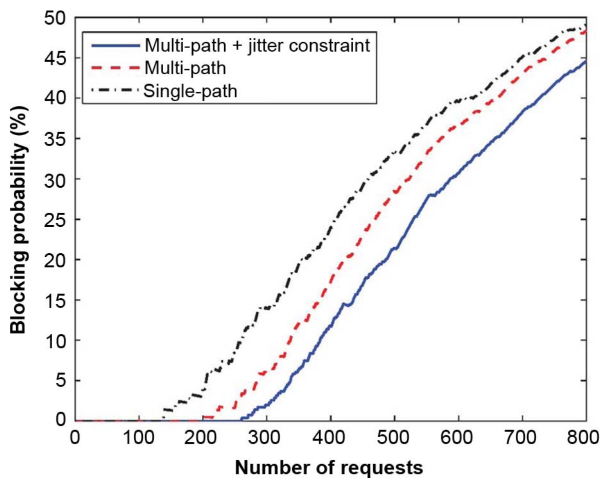


Fig. 7. Static network simulation results on blocking probability.

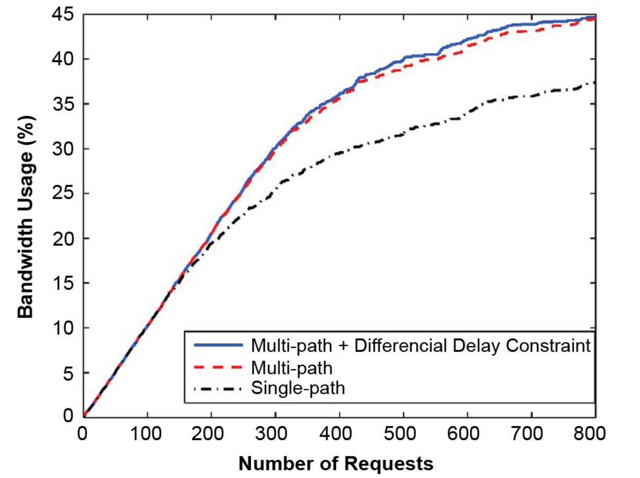


Fig. 8. Differential delay distributions of the provisioning from the two multi-path scenarios.

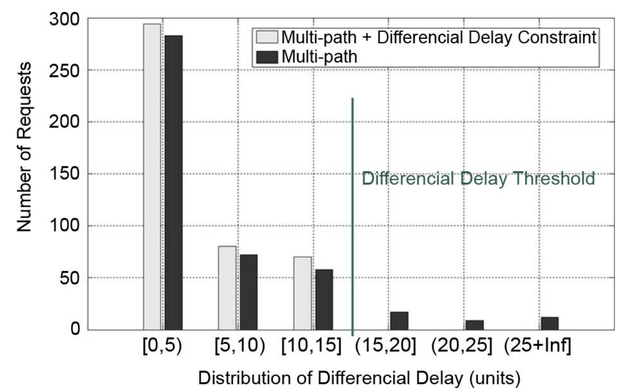


Fig. 9. Simulation setup of dynamic network.

A. Static Network Simulation

In this simulation, we compare three provisioning algorithms (e.g., single-path, multi-path without considering the differential delay constraint, and multi-path with considering the differential delay constraint) on a randomly generated mesh topology with 30 nodes. The source-destination pair of each request is randomly selected. The other parameters of each request, such as the maximum delay requirement, the differential delay requirement and the bandwidth requirement, are also randomly generated. Table I shows the simulation parameters. The single-path algorithm tries to find a single feasible path under the bandwidth and maximum delay constraints, while multi-path algorithms can allocate multiple paths to one request. To obtain sufficient statistical accuracy in the simulations, we simulated 64 requests sets and average the results to get each data point in Figs. 7–9. Figs. 7 and 8 show the results on the blocking probability and bandwidth utilization. Note that we count it as a blocking instance if the differential delay requirement cannot be satisfied in the multi-path scenarios. It can be seen that as the number of requests increases, multi-path based algorithms outperform the single-path one by yielding lower blocking probability. Between the two multi-path algorithms, our proposed algorithm provides the lowest blocking probability. From Fig. 8 we can also see that the proposed algorithm achieves the best utilization of the network bandwidth. Fig. 9 shows the detailed differential delay distributions of the provisioning from the two multi-path scenarios. It can be seen that the scenario that considers the differential delay constraint successfully controls the differential delay of the multi-path provisioning of the requests below the pre-set threshold.

B. Dynamic Network Simulation

Fig. 10 shows the setup for dynamic network simulation, and the core network topology is a 4×4 grid. The cloud resides in the core network and we assume that it has the capability to collect network status proactively and provide streaming strategies for the video provider. To emulate the practical case, we introduce a random delay to the collection process for simulating the message propagation delay in the network. At the edge of the network, a video subscriber, and several background servers and clients are presented. The background servers and clients are set up for background traffic variations. These background traffics follow the Poisson process. Other simulation parameters are shown in Table II.

Figs. 11–13 show the comparisons of the streaming bandwidth, packet delay, and packet loss rate from our proposed multi-path provisioning algorithm to those from a single-path one with the conventional shortest path strategy. A video packet will be considered to be lost if it can not be received before the decoding deadline. We use three SVC layers for the simulation, one base layer (BL) and two enhancement layers (i.e., EL1 and EL2). These simulation results show that in the single-path scenario, the three SVC layers suffer from similar congestions when bursty background traffic is present. This will lead to mosaics or freezing on subscribers' playback devices and de-

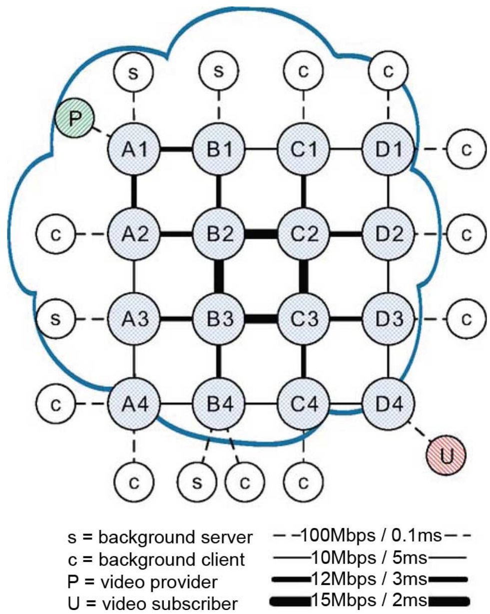


Fig. 10. Comparison of the subscriber's streaming bandwidth for SVC layers.

TABLE II
SIMULATION PARAMETERS OF DYNAMIC NETWORK.

Router output FIFO queue length	50 packets
Data-rate of SVC layer 1, Base Layer (BL)	1 Mbps
Data-rate of SVC layer 2, Enhancement Layer 1 (EL1)	1 Mbps
Data-rate of SVC layer 3, Enhancement Layer 2 (EL2)	1 Mbps
SVC video packet length	1400 Bytes
Average data-rate of background traffic	8-10 Mbps
Average switching time of background traffic	0.3-2 seconds
Burst probability of background traffic	0.6-0.85

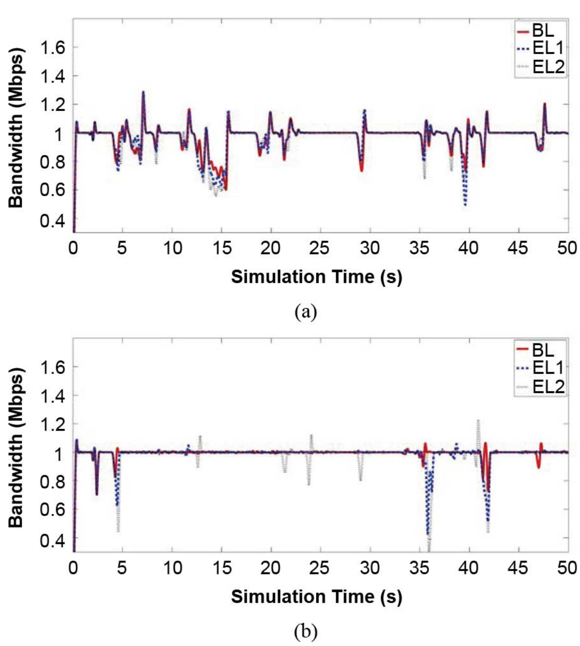


Fig. 11. Comparison of the subscriber's packet delay for SVC layers (a) Conventional Single-Path Routing Delivery Strategy (b) Multi-Path Routing Delivery Strategy.

grade their quality of experience (QoE). On the other hand, the multi-path scheme can avoid the congestions in most of the

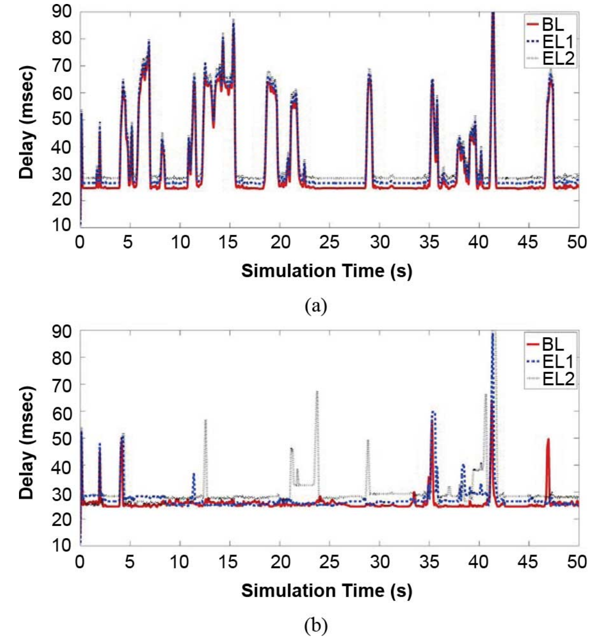


Fig. 12. Comparison on the subscriber's packet loss rate for SVC layers (a) Conventional Single-Path Routing Delivery Strategy (b) Multi-Path Routing Delivery Strategy.

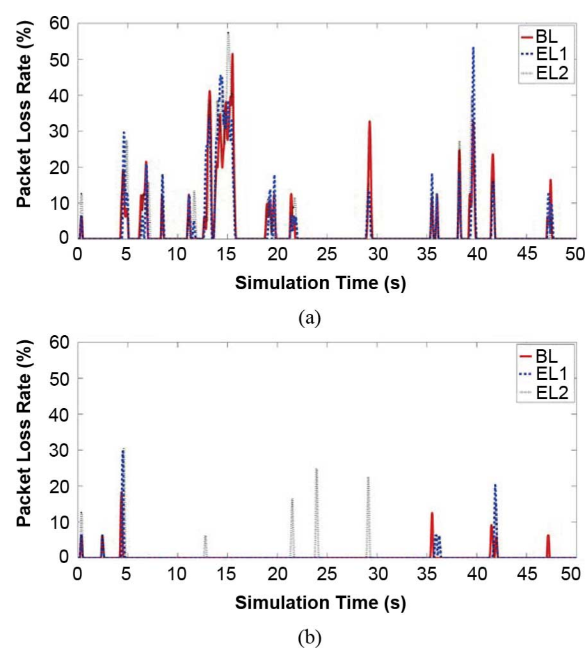


Fig. 13. Comparison of the subscriber's playback quality (a) Conventional Single-Path Routing Delivery Strategy (b) Multi-Path Routing Delivery Strategy.

cases. However, in Figs. 12–13, the delay and packet loss rate of the multipath scheme can still increase suddenly at several time points. This is mainly due to the latency in network status collection and processing. When the network status changes faster than the monitoring scheme, the algorithm may have difficulty to adjust the provisioning scheme on time and congestion can still happen. More sophisticated monitoring schemes, such as adaptive sampling, may solve this problem and we will investigate them in future works. Fig. 14 shows the playback quality of

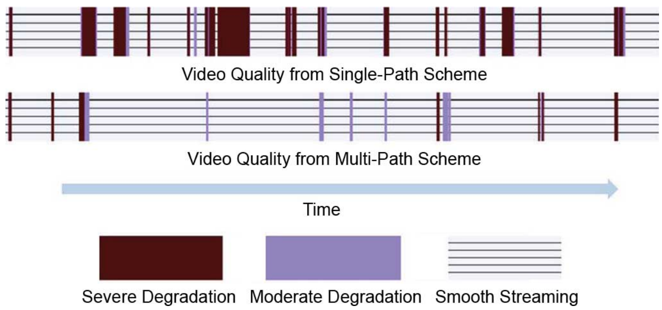


Fig. 14. Comparison of the subscriber's playback quality.

a video clip with the single- and multi-path scenarios. The period of smooth streaming, moderate quality degradation (noticeable mosaics), and severe quality degradation (video freezing) are plotted for both scenarios. It can be seen that multi-path scheme provides a much better streaming quality and user experience, when compared to the single-path one.

VI. Conclusion

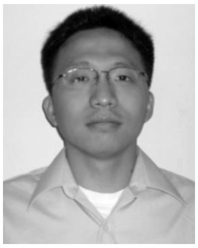
We laid out a network infrastructure that leveraged the storage and computing power of a cloud residing in the core for collecting network status and computing multi-path scalable video coding (SVC) streaming provisioning strategies. Compared to the source-routing based provisioning in the video servers, this cloud-assisted scheme could provide more cost-effective strategies with better knowledge of network environment and more powerful computation power, especially when streaming to heterogeneous clients became necessary. To realize the multi-path provisioning with high efficiency, we then proposed several algorithms for cloud-assisted SVC streaming in heterogeneous networks. To the best of our knowledge, these were the first proposals to work on the problem of adaptive multi-path routing for SVC streaming with bandwidth, delay and differential delay constraints. The simulations of the proposed algorithms in both static and dynamic network environments showed that they could achieve effective performance improvements in terms of request blocking probability, bandwidth utilization, packet delay, packet loss rate, and video play-back quality.

Acknowledgment

The authors would like to thank Mr. Zilong Bai and Miss Yanan Wu from the University of Science and Technology of China for their helps on the figures.

References

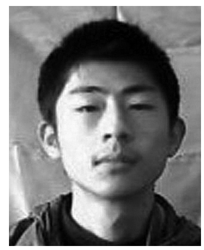
[1] A. M. Odlyzko, "Internet traffic growth: Sources and implications," *Proc. SPIE*, vol. 5247, pp. 1–15, 2003.
[2] P. Frossard *et al.*, "Media streaming with network diversity," *Proc. IEEE*, vol. 96, no. 1, pp. 39–53, Jan. 2008.
[3] J. Chen *et al.*, "Multiplath routing for video delivery over bandwidth-limited networks," *IEEE J. Sel. Areas Commun.*, vol. 22, Dec. 2004.
[4] A. Chow *et al.*, "Toward picture-perfect streaming on the interent," in *Proc. QEST*, Sep. 2005, pp. 63–72.
[5] W. Zhang *et al.*, "Reliable adaptive multipath provisioning with bandwidth and differential delay constraints," in *Proc. INFOCOM*, Apr. 2010, pp. 1–9.
[6] T. Nguyen *et al.*, "Multiple sender distributed video streaming," *IEEE Trans. Multimedia*, vol. 6, no. 2, pp. 315–326, Apr. 2004.
[7] P. Xia *et al.*, "Optimal bandwidth assignment for multiple-description-coded video," *IEEE Trans. Multimedia*, vol. 13, no. 2, pp. 366–375, Apr. 2011.
[8] J. Apostolopoulos *et al.*, "Path diversity for enhanced media streaming," *IEEE Commun. Mag.*, vol. 42, pp. 80–87, Aug. 2004.
[9] Y. Lee *et al.*, "A constrained multipath traffic engineering scheme for MPLS networks," in *Proc. ICC*, Jun. 2002, pp. 2431–2436.
[10] J. Apostolopoulos *et al.*, "On multiple description streaming with content delivery networks," in *Proc. INFOCOM*, Apr. 2002, pp. 1736–1745.
[11] W. Li, "Overview of fine granularity scalability in MPEG-4 video standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 11, no. 2, pp. 301–317, Mar. 2001.
[12] Y. Liang *et al.*, "Analysis of packet loss for compressed video: Does burst-length matter?," in *Proc. ICASSP*, Apr. 2003, pp. 684–687.
[13] D. Marpe *et al.*, "Overview of the scalable video coding extension of the H.264/AVC standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 9, pp. 1103–1120, Sep. 2007.
[14] T. Schierl *et al.*, "Using H.264/AVC-based scalable video coding (SVC) for real time streaming in wireless IP networks," in *Proc. ISCAS*, 2007, pp. 3455–3458.
[15] S. Misra *et al.*, "Polynomial time approximations for multi-path routing with bandwidth and delay constraints," *Proc. INFOCOM*, pp. 558–566, Apr. 2009.
[16] A. Begen *et al.*, "Multi-path selection for multiple description encoded video streaming," in *Proc. ICC*, Jun. 2003, pp. 1583–1589.
[17] D. Jurca *et al.*, "Video packet selection and scheduling for multipath streaming," *IEEE Trans. Multimedia*, vol. 9, no. 2, pp. 629–641, Apr. 2007.
[18] D. Jurca *et al.*, "Media flow rate allocation in multipath networks," *IEEE Trans. Multimedia*, vol. 9, no. 5, pp. 1227–1240, Oct. 2007.
[19] H. Guo *et al.*, "Cooperative media data streaming with scalable video coding," *IEEE Trans. Knowl. Data Eng.*, vol. 20, no. 9, pp. 1273–1281, Sep. 2008.
[20] S. Tao *et al.*, "Improving VoIP quality through path switching," in *Proc. INFOCOM*, Apr. 2005, pp. 2268–2278.
[21] N. Franchi *et al.*, "Multiple description video coding for scalable and robust transmission over IP," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 2, pp. 321–334, Mar. 2005.
[22] G. Wang *et al.*, "Multiple-description coding for overlay network streaming," *IEEE Multimedia*, vol. 14, no. 1, pp. 74–82, Feb. 2007.
[23] C. Hsu *et al.*, "Partitioning of multiple fine-grained scalable video sequences concurrently streamed to heterogeneous clients," *IEEE Trans. Multimedia*, vol. 10, pp. 457–469, Apr. 2008.
[24] Z. Ma *et al.*, "A new multi-path selection scheme for video streaming on overlay networks," in *Proc. ICC*, Jun. 2004, pp. 1330–1334.
[25] S. Huang *et al.*, "Survivable multipath provisioning with differential delay constraint in telecom mesh networks," in *Proc. INFOCOM*, Apr. 2008, pp. 191–195.
[26] M. Ghareeb *et al.*, "Scalable video coding (SVC) for multipath video streaming over video distribution networks (VDN)," in *Proc. ICOIN*, Jan. 2011, pp. 206–211.
[27] M. Ghareeb *et al.*, "An adaptive QoE-based multipath video streaming algorithm for scalable video coding (SVC)," in *Proc. ISCC*, Jun. 2011, pp. 824–829.
[28] E. Rosen *et al.*, "Multiprotocol Label Switching Architecture: RFC3031." 2001.
[29] Z. Chen *et al.*, "Perception-aware multiple scalable video streaming over WLANs," *IEEE Signal Process. Lett.*, vol. 17, no. 7, pp. 675–678, Jul. 2010.
[30] I. Politis *et al.*, "Optimizing video transmission over wireless multimedia sensor networks," in *Proc. GLOBECOM*, Dec. 2008, pp. 1–6.
[31] S. Mao *et al.*, "Multipath routing for multiple description video in wireless ad hoc networks," in *Proc. INFOCOM*, Apr. 2005, pp. 740–750.
[32] J. Wang *et al.*, "A multipath approach for reliable high quality video transport in indoor 60 GHz radio networks," in *Proc. LCN*, 2010, pp. 504–511.
[33] S. Li *et al.*, "Efficient and scalable cloud-assisted SVC video streaming through mesh networks," in *Proc. ICNC*, Jan. 2012, pp. 944–948.
[34] Z. Huang *et al.*, "CloudStream: Delivering high-quality streaming videos through a cloud-based SVC proxy," in *Proc. INFOCOM*, Apr. 2011, pp. 201–205.



Zuqing Zhu received the PhD degree from the Department of Electrical and Computer Engineering, University of California, Davis, in 2007. From Jul. 2007 to Jan. 2011, he worked in the Service Provider Technology Group of Cisco Systems, San Jose, as a senior R&D engineer. In Jan. 2011, Dr. Zhu joined the University of Science and Technology of China (USTC), as an Associate Professor. His research interests are energy-efficient network designs, multimedia networks, access network technologies, and optical communications and networks. He has published more than 70 papers on the journals and conferences of IEEE, IEE and OSA. Dr. Zhu has been in the technical program committees (TPC) of INFOCOM, ICC, GLOBECOM, ICCCN and etc. Dr. Zhu is also an editorial board member of Elsevier Journal of Optical Switching and Networking (OSN), Springer Telecommunication Systems Journal (TSMJ), Wiley European Transactions on Telecommunications (ETT), and etc. He is a Senior Member of IEEE and a Member of OSA.

Suoheng Li (S'13) received the B.S. degree from the University of Science and Technology of China, Hefei, China, in 2011, where he is currently working toward the Ph.D. degree.

His research focuses on next-generation Internet, multimedia networks, and video streaming.



Xiaoliang Chen received the B.S. degree from the North University of china, Taiyuan, China, in 2011. He is currently working toward the Ph.D. degree at the University of Science and Technology of China, Hefei, China.

His research focuses on next-generation Internet, multimedia networks, and green networks.