Efficient and Scalable Cloud-Assisted SVC Video Streaming Through Mesh Networks

Suoheng Li, Zuqing Zhu*, Weiping Li, Houqiang Li
School of Information Science and Technology
University of Science and Technology of China, Hefei, China
*Email: {zqzhu}@ieee.org

***Abstract*—To improve the efficiency and scalability of video streaming in large scale mesh networks, we propose a cloud-assisted video delivering strategy that can distribute video contents to subscribers over multiple routing paths dynamically. By leveraging the computation power and numerous storage of a cloud and the flexibility of scalable video coding (SVC) scenarios, we design algorithms for multiple routing path selection and assignment, and propose to run them in a cloud for fast processing. The multi-path routing is then achieved by label-switching in the network layer. The simulation results show that the proposed algorithms achieved efficient usage of the network resources and improved user experience in terms of bandwidth, delay and jitter.**

***Index Terms*—Scalable video coding, multi-path routing, video streaming, label switching.**

I. Introduction

As an extension to the conventional video coding schemes, Scalable Video Coding (SVC) supports spacial, temporal and signal to noise ratio (SNR) scalability [1, 2] . Due to the considerations on efficient storage and delivering of videos, network applications pay more attentions on the spacial and SNR scalability of SVC. For instance, video providers may encode videos with different resolutions (i.e. spacial scalable) to accommodate to various quality requirements, hardware limitations or access speeds of the subscribers. According to the operation principle of SVC, the storage requirement of the global bit stream is approximately equal to that of the highest resolution one [1, 2]. Since bit streams with all supported resolutions can be extracted directly from the global bit stream, video storage efficiency can be greatly improved. For a given resolution, the provider may also offer more than one SNR layers, to support different playback quality. It is known that the playback quality of SVC is determined by the maximum number of consecutive layers that the subscriber can get above the base layer [1, 2].

SVC provides network providers a remarkable opportunity to achieve efficient video streaming over heterogeneous networks, especially in the cases where multi-path routing can be incorporated. Previous work in [3, 4] has demonstrated that multi-path routing could distribute media contents over different paths, and thus, achieved higher aggregated bandwidth and better adaptation with the network congestions, compared to the traditional single-path routing. The theoretical analysis in [3, 4] assumed that video traffic could be split into layers with arbitrary sizes. However, in the case of SVC, each SNR layer is atomic, and needs a specific minimum bandwidth. In [5, 6], SVC video streaming strategies with multi-path routing have been proposed by using overlay networks. However, since multi-path routing usually makes the routing and forwarding protocols more complicated, the additional protocol overheads from the overlay-based approaches can make the situation even worse. Moreover, these approaches did not fully investigated how to find multiple routing paths for SVC videos, especially in large-scale mesh networks.

In this paper, we design a cloud-assisted SVC video streaming strategy that tries to deliver the largest number of SNR layers to the subscribers through mesh networks, under bandwidth, delay and decodable constraints. We incorporate multi-path routing to improve the efficiency and scalability of SVC video streaming. Specifically, for an arbitrary source-destination pair, the video delivery can take multiple routing paths for different SNR layers. By leveraging the computation power and numerous storage of the cloud, we design algorithms for finding the multiple routing paths, and propose to run them in a cloud for fast processing. The paths are then set up by the cloud with a label- or flow-switching mechanism [7, 8]. When network status changes, the cloud will re-calculate the paths and update them. Instead of relying on a overlay-based infrastructure, the layer-path mapping is done in the network layer and significant protocol overheads can be saved.

The rest of the paper is organized as follows. Section II discusses the problem formulation of the multi-path routing problem for SVC streaming through mesh networks. Section III describes the multi-path selection algorithm. Section IV shows the performance evaluation results from simulations. Finally, Section V summarize the paper.

II. Problem Formulation

A. Network Model

In this work, we consider a mesh network infrastructure as illustrated in Fig. 1. We assume that the cloud resides in the core network, and has the capability of collecting both video contents and network status proactively. Therefore, by leveraging its computation resources and numerous storage, the cloud processes the video contents, encodes them with SVC, and stores the results in one or more appropriate server(s) in the core network. Meanwhile, the cloud collects and processes the network status to figure out the optimized

978-1-4673-0009-4/12/$26.00 ©2012 IEEE

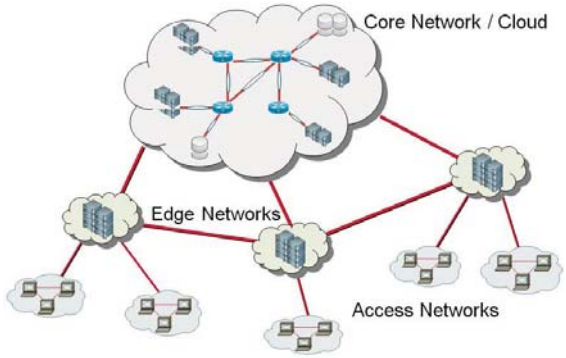

Fig. 1. Mesh network infrastructure for cloud-assisted SVC video streaming.

SVC video streaming strategy (i.e. for the best Quality of Experience (QoE)) for each request from the subscribers, and set up the routing paths by using label-switching. Specifically, the cloud instructs the routers to set up label-switching paths, and assign specific labels to the content server for encoding the SVC packets.

B. Multi-path Routing Problem

We consider the network topology as $G(V, E)$, where V is the node set, E is the network link set. $BW_{u,v}^{(Link)}$ represents the available bandwidth for the link $L_{u,v}$ between two nodes u and v ($u, v \in V$) that are directly connected. For a video streaming from node s to node d ($s, d \in V$), the content is in a scalable encoding that consists of N_i SNR layers, where i is the unique ID of the video content. Each of these layers requests a minimal bandwidth of $BW_{i,j}^{(Content)}, j = 1, ..., N_i$. We define the $R_{s,d}$ as the set of paths from s to d in the network. Then, the available bandwidth of each path $r_{s,d}^{(k)}$ is:

$$BW_{k,s,d}^{(Path)} = min(BW_{u,v}^{(Link)}), \forall k, [u, v : L_{u,v} \in r_{s,d}^{(k)}] \quad (1)$$

The delay of each path $r_{s,d}^{(k)}$ is $D_{s,d}^{(k)}$. $f_{k,s,d}^{(i,j)}$ denotes the path assignment flag. If Layer j of content i will be delivered over $r_{s,d}^{(k)}$, $f_{k,s,d}^{(i,j)}$ is 1, otherwise, it is 0. Then, the cloud-assisted multi-path routing becomes an optimization problem: Given $G(V, E)$ and $BW_{u,v}^{(Link)}$, for each video streaming from s to d for content i, the cloud needs to find a delivery strategy that can maximize deliverable SNR layer M with a $R_{s,d}^{(Selected)}$ from $R_{s,d}$, under the constraints of:

Bandwidth Constraint:

$$\sum_{j=1}^{N_i} BW_{i,j}^{(Content)} f_{k,s,d}^{(i,j)} \leq BW_{k,s,d}^{(Path)}, \forall k \quad (2)$$

Delay Constraint:

$$max(D_{s,d}^{(k)}) - min(D_{s,d}^{(k)}) \leq J_{max}, \quad \{k : r_{s,d}^{(k)} \in R_{s,d}^{(Selected)}\} \quad (3)$$

Decodable Constraint:

$$\sum_{k=1}^{|R_{s,d}|} f_{k,s,d}^{(i,j)} = 1, \{j : j \leq M\} \quad (4)$$

Here, J_{max} denotes the maximum jitter the system can tolerate between layers. It can be seen that the optimization is a NP-hard problem [9]. Therefore, the mixed integer linear programming (MILP) model listed above can only lead to optimal solutions for small networks, when the computation time is limited. We thus propose a heuristic algorithm for solving it.

III. Multi-path Selection Algorithm

The objective of the multi-path selection algorithm is to deliver more SNR layers under bandwidth, delay and inter-layer constraints. Since $BW_{u,v}^{(Link)}$ of a practical network is variable, we need to reserve a margin when assigning one layer to a selected path $r_{s,d}^{(k)}$. We define a scaling factor α for this purpose. Therefore, after assigning layer j to a routing path $r_{s,d}^{(k)}$, we update the bandwidth of the links with:

$$BW_{u,v}^{(Link)} = BW_{u,v}^{(Link)} - (1 + \alpha) BW_{i,j}^{(Content)}, \quad \{u, v : L_{u,v} \in r_{s,d}^{(k)}\} \quad (5)$$

Note that, this margin is only the calculations of multiple routing paths in the cloud, and it does not influence the overall network utilization. In addition to selecting the routing paths with sufficient available bandwidth, our algorithm also tries to avoid frame drops due to high latency or jitter in the paths. We use link delay (i.e. The one-way delay between adjacent routers with empty input and output queues) to estimate the path delay. According to the H.264 SVC standard, a subscriber has to get all the lower SNR layers when decoding a specific SNR layer [2]. Therefore, we assign layers to the routing paths based on their order j, (i.e. Layer 1 is first assigned, then Layer 2, and so on). For each subscriber, we stop the path assignment when its QoS requirement is reached or we cannot find a path under the constraints.

We assume that the cloud is capable of collecting the delay and available bandwidth of all links in the network, and store them in two matrices, $D_{u,v}^{(Link)}$ and $BW_{u,v}^{(Link)}$, respectively. *Algorithm 1* illustrates the routing path selection procedures for a SNR layer, from server node s_0 to the subscriber node d_0. In *Phase I*, we first remove links whose available bandwidth is insufficient from the topology by setting their delays to infinity in $D_{u,v}^{(Link)}$. The shortest paths from all the other nodes in V to d_0 are then calculated by applying the Dijkstra Algorithm. We push s_0 in the $HopList$, calculate the lower bound DB of the additional delay, and enqueue a structure that contains the lower bound DB of additional delay from s_0 to d_0, the total delay from s_0, and the current $HopList$ in a priority queue Q. *Phase II* searches for k shortest loopless paths and stores them in list P as candidates of the routing path assignment in *Phase III*. For each intermediate node v, we add the minimum delay from v to d_0 to path's current delay (s_0 to v), to get the lower bound DB' of the whole

path delay. The $D_{i,j}^{(Max)}$ is the upper bound of the total delay for delivering the SNR layer, obtained by Eqn. (3). In *Phase III*, when the candidate paths are obtained, we select the one with the largest available bandwidth for delivering the SNR layer. After assigning the routing path, we update the available bandwidth matrix $BW_{u,v}^{(Link)}$ accordingly.

Algorithm 1 Routing Path Selection For a SNR Layer

Input: $G(V,E)$, $D_{u,v}^{(Link)}$, $BW_{u,v}^{(Link)}$, $D_{i,j}^{(Max)}$, s_0, d_0, k, α, $BW_{i,j}^{(Content)}$
Output: $Hoplist$, $D_{i,j}^{(Max)}$ $BW_{u,v}^{(Link)}$
{Phase I}
for each $L_{u,v} \in E$ **do**
  if $BW_{u,v}^{(Link)} < (1+\alpha)BW_{i,j}^{(Content)}$ **then**
    $D_{u,v}^{(Link)} = \infty$
  end if
end for
$\{r_{u,d_0}^{Shortest}\} = Dijkstra(D^{(Link)}, d_0), \forall u \in V, u \neq d_0$
$DB = Delay(r_{s_0,d_0}^{Shortest})$
$Push(HopList, s_0)$
$Q.Enqueue([DB, 0, HopList])$
{Phase II}
$P \leftarrow \emptyset$
while $Q \neq \emptyset$ **do**
  $[DB, D, HopList] = Q.Dequeue()$ with smallest DB
  $u = Top(HopList)$
  for each neighbor v of u that does not in $HopList$ **do**
    $HopList' = HopList$
    $Push(HopList', v)$
    $D' = D + D_{u,v}^{Link}$
    $DB' = D' + Delay(r_{v,d_0}^{Shortest})$
    if $v = d_0$ **then**
      $P.Enqueue([D', HopList'])$
      if Size(P)=k **then**
        goto Phase III
      end if
    else if $DB' < D_{i,j}^{(Max)}$ **then**
      $Q.Enqueue([DB', D', HopList'])$
    end if
  end for
end while
{Phase III}
if list P is empty **then**
  return $\emptyset$
else
  select $HopList$ in P with maximum bandwidth
  update available bandwidth matrix $BW_{u,v}^{(Link)}$
  restore $D_{u,v}^{(Link)}$ to the initial values
  update delay constraint $D_{i,j}^{(Max)}$ if necessary
  return $HopList$, $BW_{u,v}^{(Link)}$
end if

Algorithm 2 illustrates the overall routing path selection procedure for multiple SNR layers and multiple subscribers. All subscriber requests $u_{i,s,d}$ (from node d to s for video content i) and their QoS requirements $u_{i,s,d}^{(QoS)}$ are first stored in a request queue REQ. Available bandwidth of each link is calculated and stored as initial value of $BW_{u,v}^{(Link)}$. We then perform the inter-layer routing path selection in a Round-Robin way, starting from the base layer $j = 1$. Basically, for a given SNR layer order j, we apply *Algorithm 1* to each request in REQ for selecting the routing path. k is set according to the computing power of the cloud, α can be either constant to all requests or variable based on their priorities, and the delay constraint $D_{i,j}^{(Max)}$ is obtained based on Eqn. (3). For example, when a path with a delay of 10 units has been selected for a SNR layer of content i and J_{max} is 40 units, the delay constraint will be 50 units for the path selection of consequent SNR layers. When there is no path can be found for a particular request $u_{i,s,d}$ under the constraints, or its QoS requirement $u_{i,s,d}^{(QoS)}$ is reached, we stop the path selection by deleting the request from REQ.

Algorithm 2 Overall Routing Path Selection

for all pending requests $u_{i,s,d}$ **do**
  $REQ.Enqueue([u_{i,s,d}, u_{i,s,d}^{(QoS)}])$
end for
get latest $BW_{u,v}^{(Link)}$
for each SNR layer order j in the ascending order **do**
  for all pending requests $u_{i,s,d}$ in REQ **do**
    $REQ.Get([u_{i,s,d}, u_{i,s,d}^{(QoS)}])$
    apply *Algorithm 1* to the j-th SNR layer of $u_{i,s,d}$
    if *Algorithm 1* returns $\emptyset$ **then**
      $REQ.Dequeue([u_{i,s,d}, u_{i,s,d}^{(QoS)}])$
    end if
    if $u_{i,s,d}^{(QoS)}$ is reached **then**
      $REQ.Dequeue([u_{i,s,d}, u_{i,s,d}^{(QoS)}])$
    end if
  end for
end for

IV. Performance Evaluation

A. Simulation Setup

The SVC in simulations is restricted to the SVC extension of H.264/AVC. We evaluated the algorithms in a 4×4 grid network topology, as shown in Fig. 2. In addition to the 16 routers, the network also includes a video provider, a video subscriber, several background servers and clients, to emulate the practical scenario. For simplicity, Fig. 2 does not include the details of the cloud infrastructure, but it can be considered as a network entity that is connected to the grid network. The servers and the router in the top left corner of Fig. 2 can be consider as a part of the cloud. The cloud collects network status proactively, run routing path selection algorithms for the requests, and provide delivery strategies for the video provider. The video packets are labeled according to the routing path selection, and the routers decide the output port for each packet based on its label. For every output port, each router equips an

TABLE I
SIMULATION PARAMETERS

Router output FIFO queue length	50 packets
Data-rate of SNR layer 1, Base Layer (BL)	1 Mbps
Data-rate of SNR layer 2, Enhancement Layer 1 (EL1)	2 Mbps
Data-rate of SNR layer 3, Enhancement Layer 2 (EL2)	2.5 Mbps
SVC video packet length	1400 Bytes
Average data-rate of background traffic	4-7 Mbps
Average switching time of background traffic	0.3-3 seconds
Burst probability of background traffic	0.5-0.98

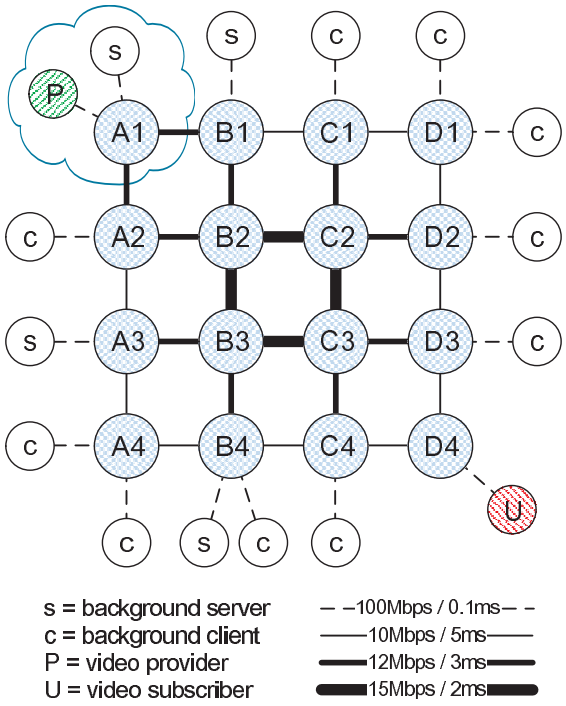

Fig. 2. Network topology for simulations.

output FIFO queue. When an output FIFO queue is full, the router will drop all the new packets. The subscriber can request a video content from the provider. When the routing path selection has been done, the provider sends SVC packets to the subscriber with a fixed packet length. The background servers and clients are set up for emulating network congestions. These servers can change their target clients randomly and sent background traffic to them in a burst-mode with an average data-rate and a predefined burst probability. The generation of the background traffic follows the Poisson process with a uniform distributed packet size from 64 to 1500 Bytes. These background packets are routed over the shortest routing path calculated from the link delay matrix $D_{u,v}^{(Link)}$. Each link in the topology has a predefined throughput and transmission delay, as illustrated in Fig. 2. We assume all the SVC videos are encoded into three SNR layers. Table I shows the simulation parameters.

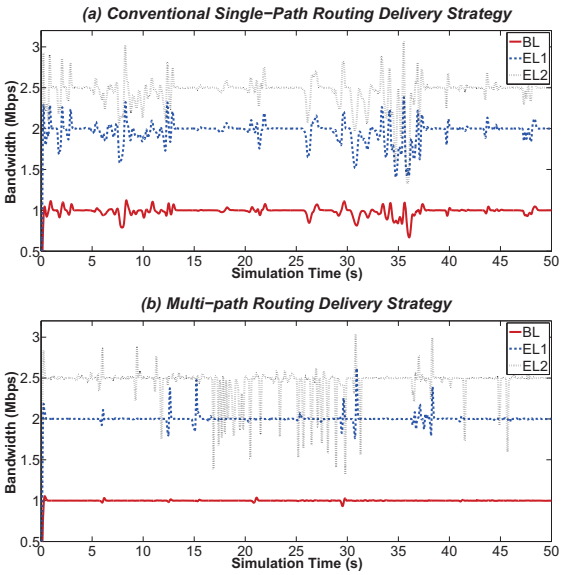

Fig. 3. Comparison of the subscriber's receiving bandwidth for SNR layers.

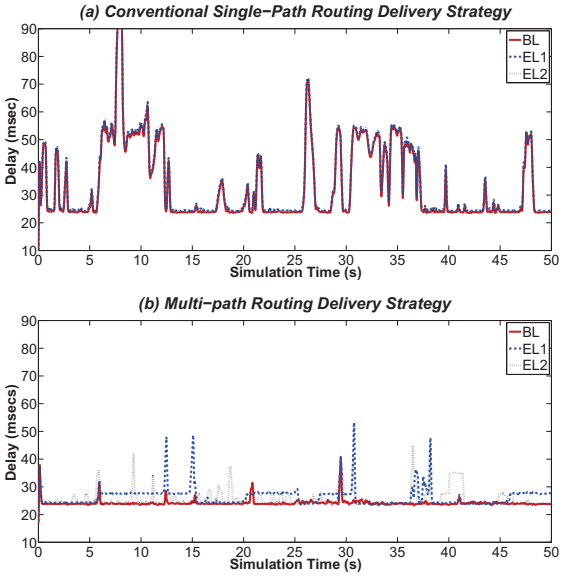

Fig. 4. Comparison of the subscriber's packet delay for SNR layers.

B. Simulation results

Fig. 3-5 show the comparisons for the subscriber's receiving bandwidth, packet delay and packet loss rate of different SNR layers. Here, the packet delay includes both the link delay from physical transmission, and the queuing delay from the FIFOs in the routers. A video packet is considered to be lost when it has not arrived at the subscriber within a preset decodable time frame. We assume there are three SNR layers, base layer (BL), enhancement layer 1 (EL1), and enhancement layer 2 (EL2), with the data-rates in Table I.

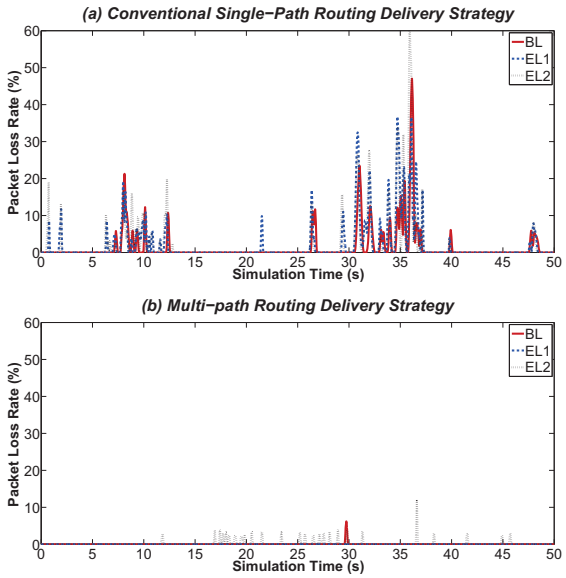

Fig. 5. Comparison on the subscriber's packet loss rate for SNR layers.

For the conventional scenario with single shortest-path routing, the three SNR layers are delivered through the same fixed path. Due to the high correlation between these layers, their bandwidth occupations are similar. Therefore, as shown in Fig. 3(a), when the network situation is not good, there will be significant decrease in bandwidth for all of the three layers, and the subscriber will have difficulty to recover a smooth video streaming. While in our multi-path routing scenario, the routing paths are calculated dynamically based on the network status. Since the BL takes the best routing path in terms of available bandwidth, its bandwidth is guaranteed even in the case of network congestions (as shown in Fig. 3(b)). The bandwidth variations on EL1 and EL2 are also reduced in Fig. 3(b).

In Fig. 4(a) and 5(a), high delay and packet loss can occur when there is a network congestion, for the traditional scenario. For the results of our multi-path routing scenario in Fig. 4(b) and 5(b), the packet loss rate is minimized, due to the fact that we will change the affect path to a new one in network congestions, before the routers' FIFO queues become overflowed. Since we make available bandwidth as the highest priority in the routing path selection, there can be situations where the lower SNR layer takes a path with relatively longer delay but larger bandwidth. As illustrated in Fig. 4(b), from 5 to 15 seconds of the simulation time, EL1 takes a path with longer delay, compared to the one of EL2. But since the available bandwidth of EL1's path is larger, it does not experience packet loss at about 12 seconds as EL2 due to the network congestion, as shown in Fig. 5(b). In Fig. 5(b), we can also see that there is no packet loss for EL1, but there are packet losses at 30 seconds for BL. This is because that to emulate a practical case, we introduce a delay for the cloud's data collection of link characteristics. Thus, the path-switching of BL may be a little late due to this delay.

V. Conclusion

In this paper, we proposed a cloud-assisted video delivering strategy that distributed video contents to subscribers over multiple routing paths dynamically. The multi-path routing was achieved by label-switching in the network layer, and two algorithms were proposed for multiple routing path selection and assignment. By leveraging the computing power of a cloud and the flexibility of scalable video coding (SVC) scenarios, we achieved efficient usage of the network resources and improved user experience in terms of bandwidth, delay and jitter, with the algorithms. Simulation results showed that our algorithm effectively improved the video delivery capability of the network and the video streaming's tolerance to network congestions.

References

[1] W. Li, "Overview of fine granularity scalability in MPEG-4 video standard", *IEEE Trans. Circ. Syst. Vid.*, vol. 11, pp. 301-17, Mar. 2001.

[2] H. Schwarz, and et. al., "Overview of scalable video coding extension of the H.264/AVC standard", *IEEE Trans. Circ. Syst. Vid.*, vol. 17, pp. 1103-1120, Sept. 2007.

[3] A. Chow, and et. al., "Toward picture-perfect streaming on the Internet", *in Proc. of QEST 2005*, pp. 63-72, Sept. 2005.

[4] J. Chen, and et. al.,"Multipath Routing for Video Delivery Over Bandwidth-Limited Networks", *IEEE J. Select. Areas Commun.*, Vol. 22, pp. 1920-32, Dec. 2004.

[5] M. Jain, and et. al., "Path selection using available bandwidth estimation in overlay-based video streaming", Comp. Netw. J., vol. 52, pp. 1-12, Aug. 2008.

[6] M. Ghareeb, and et. al., "Scalable video coding (SVC) for multipath video streaming over video distribution networks (VDN)", *in Proc. of ICOIN 2011*, pp. 206-211, Jan. 2011.

[7] Y. Lee, and et. al., "A constrained multipath traffic engineering scheme for MPLS networks", *In Proc. of ICC 2002*, pp. 1-5, Apr. 2002.

[8] E. Rosen, and et. al., "Multiprotocol label switching architecture", *RFC 3031*, Jan. 2001.

[9] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, 1979