Performance Evaluation for Multicasting Video over OpenFlow-based Small-scale Network

Thuyen Minh Thi*, Thong Huynh**, In-Yeup Kong***, Won-Joo Hwang****

ABSTRACT

When demand for transmitting multimedia data increases, network congestion is more likely to occur and users will suffer high loss rate as well as high delay. In order to enhance quality-of-service (QoS) of video multicasting, we need to raise transmission reliability and reduce end-to-end delay. This paper proposes a routing mechanism for a OpenFlow-based small-scale network in order to multicast video reliably with low delay. In our method, multipath routing will be applied to Multiple Description (MD) Coded video to exploit its multi-description property. Through performance evaluation, our method shows improvement on loss rate, delay and video distortion.

Key words: Video Multicast, OpenFlow, Multiple Description Coded Video, Multipath Routing, Optimization of Routing.

1. INTRODUCTION

With the development of demand for multimedia services, network load grows quickly. When congestion occurs, QoS of video multicast will drop because of high packet loss rate and delay. Packets become useless when they reach destination late. Therefore, both packet loss and delay lead to high level of video distortion. In order to deal with this problem, this paper proposes a solution that allows multicast video with low delay and low distortion. Current advanced architectures and technologies (e.g., IntServ, Diffserv, Multiple protocol label switching (MPLS)) attempt to solve such QoS problem. However, they are built on top of distributed hop-by-hop routing architecture. Without information of state and resource of the whole network, it is hard to gain efficiency of routing.

OpenFlow can define flows by various ways. This allows us to classify traffic type and define the flow that we want to support QoS enhancement. Our algorithm is run at OpenFlow controller. It keeps up-to-date information of the network state, so it is able to perform QoS routing efficiently.

In this paper, we propose a routing mechanism in order to multicast video in OpenFlow-based network with low packet loss rate and delay. However, due to the centralized architecture of OpenFlow, we aim to apply our method to a small-scale network to maintain low overhead and high stability. For a video source, there are many unicasts flows: each of them is between the server and a client. For each flow, we consider a routing problem in order to find optimal transmitting paths.

* Corresponding Author: Won-Joo Hwang, Address: (621-749) Department of Information and Communications Systems, Inje University, 197 Inje-ro, Gimhae-si, Gyeongsangnam-do, Korea, TEL: +82-55-320-3847, FAX: +82-55-322-6275, E-mail: ichwang@inje.ac.kr
† Receipt date: Mar. 10, 2014, Revision date: Jun. 24, 2014
‡ Approval date: Jul. 22, 2014
§ Department of Information and Communication System, HSV-TRC, Inje University
(E-mail: thuyenthuan@gmail.com)
** Department of Information and Communication System, HSV-TRC, Inje University
(E-mail: huythongtc@gmail.com)
*** Department of Electronic Engineering, Kumoh National Institute of Technology
(E-mail: iykong@kumoh.ac.kr)
**** Department of Information and Communication System, HSV-TRC, Inje University

These paths aim at boosting reliability and reducing delay. In the network, flows of traffic are classified at OpenFlow-enabled forwarders. Video flows are transmitted among forwarders using MD coding. In MD, video stream is encoded into several independent descriptions (i.e., substreams). The controller monitors network states and applies routing mechanism to these descriptions. Our multipath routing algorithm is used for transmitting MD coded video to destination nodes. For each of these nodes, we provides different descriptions with different paths. In this way, we exploit multi-descriptions property of MD. Our routing objective is to find optimal paths in order to minimize video distortion at each destination node while delay is kept below play-back deadline.

Our model can be used in various applications. For instance, it can be applied to a small-scale network, e.g., a regional part of a network operator's backbone. It will enable video service providers to multicast data with low loss rate and delay.

The summary of our contributions in this paper is expressed as follows:

- We propose a method to multicast video in OpenFlow-based network with low loss rate and delay. Our method performs multipath routing for each connections (Section III).
- We formulate an optimization problem for routing MD coded video (Section IV). Additionally, we use optimization methodologies to solve the modelled multipath routing problem (Section V).

2. RELATED WORKS

Video transmission in general and video multicast in particular have found a number of applications. We describe here some previous works relating to them.

In [1], the authors proposed a routing protocol for multicast. They used single path to transmit data from sender to each receiver. The single path was based on hop count. In this way, if the capacity of the shortest path did not meet the QoS requirement, the method could not provide a path with high QoS. In [2], the authors dealt with building shared multicast tree. They used multipath routing and proposed spanning-joins protocol. When a new node wanted to join multicast tree, it broadcasted request messages to its neighborhood. Therefore, this process may face the problem of overhead.

In [3], the authors dealt with video transmission problem. They used metric distortion but the objective function was Lagrangian cost function. It was not in close form as ours. Moreover, objective function did not consider packet loss due to missing play-back deadline. Similarly, Mao et al. [4] used distortion-rate function to estimate distortion based on data bit rate. Besides, they did not take play-back deadline into consideration. In [5], the authors streamed MD video over overlay networks. Distortion was estimated by a function of source rate. They had this approximate function by experiment. In [6], differential multicast routing was considered, however, it was for DWDM networks. Therefore, it is not suitable for our target networks.

In our method, centralized control mechanism of OpenFlow is employed so that we do not have problem of overhead. We derive distortion based on delay, traffic load, packet loss and play-back deadline. With distortion, we have a metric that is closer to user's perception than metrics in [7]. We use this metric to estimate the performance of multicast paths. Besides propose a mechanism, we build a mathematical model as well as provide its formulation. Furthermore, we propose a multipath routing algorithm in order to benefit from multi-description property of MD coding.

3. SYSTEM MODEL

3.1 Network description

In our network (Fig. 1), data will travel from the server, through OpenFlow-based network, then
arrive at the clients. For each forwarder, we consider two types of traffic: traffic of video flows and traffic of non-video flows. Each video flow is routed on two paths from the server to its client. Non-video flows are routed by default routing methods. This OpenFlow-based network is denoted as a directed graph.

3.2 Problem

For each server-client pair of multicast transmission, we form a routing problem to find its two optimal paths. Path \( r \) transmits data with distortion \( D(r) \). \( r^* \) denotes one of two optimal paths. The problem can be stated as finding

\[
    r^* = \arg \min_{r \in R} (D(r))
\]

subject to

\[
    t_e(r) \leq \Delta
\]

\[
\sum_{(v,w) \in E} l_{(u,v)} - \sum_{(v,w) \in E} l_{(v,w)} = \begin{cases} 1, & u = s \\ -1, & u = d \\ 0, & \text{otherwise} \end{cases}
\]

where \( E \) is link set; \( R \) is path set; \( v \) and \( u \) are nodes; \( l_{(u,v)} = 1 \) if the link \( (u,v) \) is on path \( r \); similarly, \( l_{(v,u)} = 1 \) if the link \( (v,u) \) is on path \( r \). First constraint guarantees that play-back delay requirement is met, i.e., expected end-to-end delay \( t_e(r) \) must be less than or equal to play-back deadline \( \Delta \). The value of \( \Delta \) is estimated as in [8]. Second constraint guarantees that every path \( r \) is formed by connected links and each path connects source node \( s \) and destination node \( d \).

This paper proposes a model for the encoding mode that generates two descriptions. Routing algorithm will find two paths for two descriptions. As in [4], distortion \( D \) of each pair of paths \( (r_1, r_2) \) is

\[
    D = \sum_{i=0}^{1} (p_{i,0}D_{i,0} + p_{i,1}D_{i,1})
\]

\( D_{i,0} \) is distortion of a pixel when its data from two descriptions are lost. \( p_{i,0} \) is central distortion when none of descriptions is lost; \( D_{i,0}, D_{i,1} \) are side distortions obtained when first description, second description is lost, respectively. Similarly, \( p_{i,0}, p_{i,1}, p_{0,0}, p_{0,1} \) are probabilities with which none of descriptions is lost, first description is lost, second description is lost, both description are lost, respectively.

In order to estimate distortion, we use ROPE as in [9] and make formulation as in [4].
4. PROBLEM FORMULATION

4.1 Derivation of packet loss rate $p_{i,j}$

Two causes responsible for packet loss are bad link state and play-back deadline missing.

Without loss of generality, we denote two optimal paths with joint part and disjoint parts. We use Gilbert-Elliott model to formulate loss rate on joint part. Loss rate on disjoint parts is formulated by multiplication law of probability.

$p_{i,j}$ in (2) is calculated as

$$
\begin{align*}
  p_{0,0} &= p_{\text{up}} p_{\text{down}} p_{\text{int}} p_{\text{on-time}} \\
  p_{0,1} &= p_{A1} + p_{B1} + p_{C1} \\
  p_{1,0} &= 1 - p_{0,0} - p_{0,1} - p_{1,1} \\
  p_{1,1} &= p_{A2} + p_{B2} + p_{C2} \\
  p_{1,1} &= 1 - p_{0,0} - p_{0,1} - p_{1,0}
\end{align*}
$$

(3)

where

$$
\begin{align*}
  p_{A1} &= p_{\text{down}} p_{\text{int}} p_{\text{on-time}} \\
  p_{B1} &= p_{\text{up}} p_{\text{int}} (1 - p_{\text{on-time}}) p_{\text{on-time}} \\
  p_{C1} &= p_{\text{up}} p_{\text{int}} p_{\text{down}} p_{\text{on-time}} (1 - p_{\text{on-time}}) \\
  p_{A2} &= p_{\text{down}} p_{\text{int}} p_{\text{on-time}} \\
  p_{B2} &= p_{\text{up}} p_{\text{int}} (1 - p_{\text{on-time}}) p_{\text{on-time}} \\
  p_{C2} &= p_{\text{up}} p_{\text{int}} p_{\text{down}} p_{\text{on-time}} (1 - p_{\text{on-time}})
\end{align*}
$$

(4)

Equations (3) and (4) are described as follows.

First, in order for both packets to arrive at destination without loss ($p_{0,0}$), we need all following conditions:

- joint part is in good condition (with probability $p_{\text{up}}$),
- $i$-th packet passes the $i$-th disjoint part intact ($p_{\text{int}}$) and also arrives at its destination on-time ($p_{\text{on-time}}$).

Second, when there is only second packet arrives at client ($p_{0,1}$), we expect that loss occurs at joint part, first disjoint part or first packet arrives late with probabilities $p_{A1}, p_{B1}, p_{C1}$, respectively. In this case, there are three possibilities as follows:

- joint part is in bad condition ($p_{\text{down}}$),
- second packet passes second disjoint part intact ($p_{\text{int}}$) and also arrives at its destination on-time ($p_{\text{on-time}}$),
- first packet is lost at first disjoint part ($1 - p_{\text{int}}$),
- second packet passes second disjoint part intact ($p_{\text{int}}$) and also arrives at its destination on-time ($p_{\text{on-time}}$),
- or
- joint part is in good condition ($p_{\text{up}}$),
- first packet passes first disjoint part intact ($p_{\text{int}}$) but not on-time ($1 - p_{\text{on-time}}$),
- second packet passes second disjoint part intact ($p_{\text{int}}$) and also arrives at its destination on-time ($p_{\text{on-time}}$).

Probability $p_{0,1}$ of event when first packet arrived at destination successfully and second packet gets lost is obtained from $p_{A2}, p_{B2}, p_{C2}$ by the same way with obtaining $p_{1,0}$. The $p_{0,1}$ is probability when both packets are lost.

1. Probabilities relating to link state ($p_{\text{up}}, p_{\text{down}}, p_{\text{int}}$)

Packet loss in wired connection mainly comes from drop due to congestion. In this two-state Gilbert-Elliott model, we use up-state and down-state to denote the good and bad condition of the joint part. Loss rate on disjoint parts is formulated by multiplication law of probability.

After collecting burst length and packet loss rate from forwarders, controller is able to estimate $p_{\text{up}}, p_{\text{down}}, p_{\text{int}}$ of joint part and $p_{\text{int}}, p_{\text{on-time}}$ of disjoint parts as in [5].

2. Probabilities relating to missing play-back deadline ($p_{\text{on-time}}$)

Queuing is the main cause of delay in our network. In order to measure delay, Discrete Time Markov Chain (DTMC) (Fig. 2) is adopted and applied to queuing process at all nodes [10]. At each node's queue, we examine the number of packets of the description that we want to estimate delay. This number changes over time and forms the
states of a Markov chain. This chain is (X) and its state set is \( \{Z, \ldots, Z, \ldots, Z\} \), where \( n \) is queue size. We estimate the waiting time of a packet in node’s queue in order to model queuing delay. (Y) is an absorbing Markov chain derived from (X) by ignoring all arrival after the considered packet arrives.

Our formulation is based on [10]. However, for video multicast in wired links, we do not consider retransmission mechanism. Therefore, each time unit, a node always transmits a packet successfully. \( P \) is probability that (X) stays at the same state between two time units. In our model, \( P \) is probability with which the node is serving another packet rather than a packet of examined description.

From modelled DTMC, we have probability that \( i \)-th description is transmitted on-time by path \( r \). This probability is \( P \) and it is calculated as

\[
P_{\text{on}} = P(k \leq \frac{A}{T}) = \sum_{k=0}^{\frac{A}{T}} f_{s}[r](k)
\]

where \( f_{s}[r](k) \) is probability that end-to-end delay on path \( r \) equals to \( k \) time unit(s) \( T \).

Now, \( P_{\text{on}} \) in (5) is put into (3) and (4) to fully derive (2).

4.2 Derivation of end-to-end delay

In above DTMC, the amount of time \( t_{u,v} \) that \( (X) \) transits from state \( Z \) to state 1 is calculated as shown below

\[
t_{u,v} = \sum_{j=0}^{n} P_{X_{v},\overline{j}} t_{u,v} (j) + 1
\]

where \( P_{X_{v},\overline{j}} \) is element \((i,j)\) in transition probability matrix of chain \((X)\).

Based on \( t_{u,v} \), we can estimate delay on link \((u,v)\). This delay is queuing delay \( t_{u,v} \) of node \( v \). It is given as

\[
x = \sum_{i=0}^{n} P_{X_{v},\overline{i}} H_{X_{v},\overline{i}}(1 - \epsilon) - \epsilon
\]

where \( P_{X_{v},\overline{i}} = \sum_{i=0}^{n} P_{X_{v},\overline{i}}(1 - \epsilon) \) is amount of time for \( v \)-th node to transit from state \( Z \) to state \( Z \), \( H_{X_{v},\overline{i}} \) is \( i \)-th element in steady-state probability matrix \( H_{X} \) of chain \((X)\) of \( v \)-th node.

Finally, we can estimate end-to-end delay \( t_{v}(r) \) based on (6). \( t_{v}(r) \) is the sum of queuing delay \( t_{u,v} \) at all nodes in path \( r \). It is given as

\[
x = \sum_{u=1}^{n} t_{u,v}, (u,v) \in r
\]

where \( n \) is number of nodes in the network.

5. SOLUTION PROCEDURE

If we use binary indicator vectors of links to form routes, our problem is combinatorial optimization that belongs to discrete optimization. For each server-client pair, we apply subgradient method to the search for optimal solutions. This is an efficient method in discrete optimization. In each iteration of subgradient algorithm, we solve the problem’s relaxed form. Our subgradient algorithm stops when value of subgradient equals to zero. This time, the solution of relaxed problem is globally optimal solution. This solution provides two optimal paths for transmitting video to a client.
After finding out solution of a client, the algorithm continues with other clients in multicast group.

Lagrangian relaxation of original problem (1) has following form

\[ r^* = \arg \min_{r \in R} (D(r) + \lambda (t_r(r) - \Delta)), \lambda \geq 0. (7) \]

Then Lagrangian relaxation dual has following form

\[ L^*(\lambda) = \max_{\lambda \geq 0} [\arg \min_{r \in R} (D(r) + \lambda (t_r(r) - \Delta))]. \]

The relaxed form (7) is a routing problem. We can use objective function to estimate routing cost, we can solve this problem by using a basic routing algorithm (e.g., Dijkstra's algorithm). An idea to deal with discrete optimization problems is exploiting its structure in order to speed up the search for the solution. We exploit Dijkstra's Shortest Path Algorithm to find routing solutions for the Lagrangian relaxation dual. The idea of Dijkstra's algorithm is spreading out from the source node until destination node is met. When the graph of network is represented by a tree rooted at source node, our algorithm runs along the branches on tree from the root.

Conventional Dijkstra's algorithm will run from source node and gradually add more nodes to a set of selected nodes. For each iteration, it examines all unselected nodes that directly link to at least one of its selected nodes. The set of those examined nodes is called candidate set. From that candidate set, the algorithm selects the best node with least cost from the source node. Our modified algorithm goes through the same processes with Dijkstra's algorithm but it will select a pair of nodes when it starts at the source node. After that, at each iteration, the algorithm selects only one best node based on cost (i.e., distortion). In order to estimate cost, (7) requires two paths. That why the algorithm picks up two nodes at first. After that, with each candidate node, the algorithm adds up previous selected node to form a pair of paths.

6. SIMULATION

We build a simulated OpenFlow-based network in Opnet modeler. In the network, a server multicasts video data to clients. Our algorithm is implemented at a logically centralized node. This plays the role of controller in OpenFlow. Input data is CIF version of MD coded video. Information of video sequences is shown in Table 1.

Our method is compared with a method which uses Dijkstra's Shortest Path algorithm (DSP). That method mixes all kinds of traffic and performs single path routing. Our method combines routing algorithm (RA) and optimization methodologies (OM), so that it is named RAOM. The evaluating metrics are received data, end-to-end delay and PSNR. The results show below are collected at one of the clients.

In order to evaluate loss rate, we set up a scenario in which RAOM and DSP transmit Container sequence. For each method, we stream all 12 seconds of the sequence one time. Cumulative received data is collected at destination node. Our method sets constraint on end-to-end delay. However, in objective of distortion, packet loss is taken into consideration. This explains that our method yield less loss than DSP. In Fig. 3, we add the solid line to show play-back rate. According to Fig. 3, our method achieves higher data rate than DSP. DSP takes 3 seconds to reach play-back rate while RAOM does not. This means DSP will encounter initial delay longer than RAOM. On average, RAOM achieves 4.5 kbyte/s higher than DSP.

<table>
<thead>
<tr>
<th>Table 1, Information of video sequences</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of frames</td>
</tr>
<tr>
<td>Size (bit)</td>
</tr>
<tr>
<td>Time (s)</td>
</tr>
<tr>
<td>Mean frame size (bit)</td>
</tr>
<tr>
<td>Mean bit rate (bit/s)</td>
</tr>
</tbody>
</table>

Performance Evaluation for Multicasting Video over OpenFlow-based Small-scale Network
In order to measure end-to-end delay, we run a scenario for Foreman sequence. Results is shown in Fig. 4. When the load of forwarders raises, the queuing delay will suffer. At first, RAOM produces relatively high delay. This can be explained that computing complexity requires processing time. In this way, after controller successfully solves the problem, delay gradually drops. Overall, our method gains lower delay than DSP.

In order to compare PSNR, we run scenarios of Foreman sequence. We calculate PSNR based on YUV files. However, we only show PSNR of Y component because it is the most widely-used component for PSNR. Additionally, this is per-frame PSNR, i.e., it is calculated for each frame. Fig. 5 plots PSNR of 63 of total 400 frames in Foreman sequence. The graph shows that our method achieves higher PSNR than DSP. Moreover, RAOM appears to have highly dynamic routing mechanism when its PSNR fluctuates greatly in comparison with DSP.

7. CONCLUSION

In this paper, we propose a routing method with QoS enhancement for video multicast. In order to have determining metric for routing, distortion of MD video is predicted. That estimation is done with regard to delay, traffic load, packet loss. Packet loss is estimated by link state model. Delay is estimated by queuing model. We solve routing problem of each server-client pair by combining routing algorithm (i.e., Dijkstra’s algorithm) and optimization methodologies (i.e., relaxation, subgradient). Multipath routing is applied to MD to gain loss-resilient transmission. The routing problem is formulated as an optimization problem. It first is relaxed by Lagrangian method; then it is solved by subgradient method. Each subgradient iteration calls a modified Dijkstra’s algorithm to find locally optimal pair of paths. In simulation, we compare our method with a conventional method. Our approach gains better performance in terms of received data, delay and PSNR.

REFERENCE


