32nd IEEE Conference on Local Computer Networks

Efficient MD Coding Core Selection to Reduce the Bandwidth Consumption

† Sunoh Choi, *Sang-Seon Byun and † *Chuck Yoo

† Department of Computer Science and Engineering,
Korea University, Seongbuk-gu, Anam-dong, Seoul, Korea
{sochoi, hxy}@os.korea.ac.kr

* Graduate School of Embedded Software,
Korea University, Seongbuk-gu, Anam-dong, Seoul, Korea
ssbyun@os.korea.ac.kr

Abstract - **Multiple distribution trees and multiple description (MD) coding are highly robust since they provide redundancy both in network paths and data. However, MD coded streaming includes a redundant information, which results in additional bandwidth consumptions in entire distribution trees. In this paper, we deploy core nodes in distribution tree, and give a role of MD coding to each core node, instead of a source node then we show how amount of bandwidth consumption can be reduced. Since the problem of finding an optimal set of core nodes is proved to be NP-hard, an intuitional heuristic-based algorithm is proposed. The simulation results show that our heuristic algorithm reduces the bandwidth consumptions by about 25% in the hierarchical topology compared to the MD coding in source node only.**

I. INTRODUCTION

Recently, as the demands for multimedia streaming service have been increased, such as VOD, video conferencing, IPTV [12], and user created contents (UCC), the service with single path routing may not be able to provide enough bandwidth to serve high quality streaming.

Therefore, there have been plentiful literatures for the efficient and scalable streaming overlay construction [1, 2, 3, 4, 8, 9, 10, 14, 15, 16]. Among those approaches, multiple diverse distribution trees and Multiple Description (MD) coding were proposed to provide redundancy both in network paths and data [2]. The MD coding encodes an original stream into multiple descriptions including basic frames and additional information, and the descriptions are distributed over different trees. Therefore it allows the clients to be able to extract the original stream from the descriptions delivered through different trees. Moreover, they can assemble the original stream even if some of descriptions are lost.

Since an MD coded streaming needs redundant information [3], it requires about 20% more bandwidth compared to a bare distributed streaming technique such as DVS [13]. If the MD coding is performed at a source node only, it is certain that the total bandwidth consumptions will be higher through the entire distribution trees. On the other hand, if it is performed at some intermediate nodes, namely *MD Coding Cores*, on behalf of a source node, the total bandwidth consumption can be reduced.

In this paper, we impose the role of the MD coding on some intermediate nodes that have enough bandwidth to receive the original streaming from the original source node. Then we configure the set of cores to minimize the entire bandwidth consumptions, and in the meantime, to provide the robustness and redundancy. Since this problem is proved to be NP-hard, we propose an intuitional heuristic-based method to obtain the good solution in reasonable time. In order to evaluate our heuristic, we model this problem with mixed integer programming and compare the solution of our heuristic with LP-relaxed solution.

The rest of this paper is organized as follows. In section II, the mixed integer programming model and the proof of NP-hardness are given. In section III, the heuristic algorithm is discussed. We then present a performance evaluation of our heuristic on hierarchical topology in section IV, and concluding remarks are given in section V.

II. MIXED INTEGER PROGRAMMING & TIME COMPLEXITY OF CORE SELECTION PROBLEM

Given our assumption that all paths from a source to cores and from the cores to receivers are fixed, the core selection problem can be cast into a nonlinear programming formulation. In general, nonlinear integer programming problems are very hard to solve and, in fact, no general solution approach is known [11]. Therefore we convert this problem to a linear form.

Generally, demanding rate and available network bandwidth change discretely. Therefore a bandwidth can be expressed as a multiple of unit bandwidth. For example, if the amount of the unit bandwidth is 200 kbps, 1 Mbps is expressed as five times of the unit bandwidth. In this paper, we assume that demanding rate of each receiver and available bandwidth of each link are expressed as a multiple of unit bandwidth.

We also assume that if a receiver requires m times of unit bandwidth, the bandwidth is split into m objects and each object requires a unit bandwidth. By this assumption, we can formulate the core selection problem as a mixed integer programming model and prove that it is NP-hard.

Now we can give the mixed integer programming formulation of the core selection problem with the assumptions presented above.

Let O present the set of objects. The variables for the mixed integer programming model are given as follows:

x_i : binary, indicate a node i is selected as a core.

y_{ij} : binary, indicates that an object j is assigned to a core i.

$\lambda^{e}_{\Omega i}$: binary, indicates that a link e lies on the path between the source Ω and core i.

δ^{e}_{ij} : binary, indicates that a link e lies on the path between a core i and a object j.

B : bandwidth of the original stream.

Our objective is:

0742-1303/07 $25.00 © 2007 IEEE
DOI 10.1109/LCN.2007.64

$$Minimize\sum_{e\in E}\left(\sum_{i\in V-\Omega}\lambda_{\Omega i}^{e}x_{i}B+\sum_{i\in V-\Omega}\sum_{j\in O}\delta_{ij}^{e}y_{ij}\right) \quad (7)$$

$$s.t\sum_{i\in V-\Omega}y_{ij}=1\text{, for each } j\in O \quad (8)$$

$$s.t\ x_{i}\geq y_{ij}\text{, for each } i\in V-\Omega, j\in O \quad (9)$$

$$s.t.\sum_{i\in V-\Omega}\lambda_{\Omega i}^{e}x_{i}B+\sum_{i\in V-\Omega}\sum_{j\in O}\delta_{ij}^{e}y_{ij}\leq b(e)\text{,} \quad (10)$$

$$\text{for each } e\in E$$

$$s.t.\ x_{i}\in\{0,1\}\text{, for each } i\in V-\Omega \quad (11)$$

$$s.t.\ y_{ij}\in\{0,1\}\text{, for each } i\in V-\Omega, j\in O \quad (12)$$

The first constraint ensures that each object j is assigned to exactly one core. The second constraint indicates that a node i must be a core if and only if an object is assigned to it. The third constraint guarantees that the sum of the bandwidth used by all core-object pairs and all source-core pairs on each link does not exceed its available bandwidth.

Now we prove the core selection problem with minimizing bandwidth consumption is NP-hard.

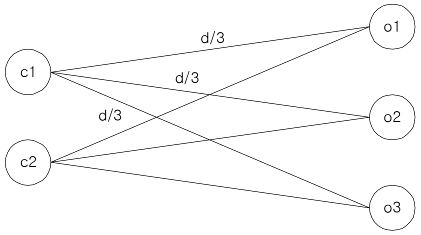

Figure 1. Reduction from set-covering problem

***Theorem* 1:** Given a graph $G = (V,E)$ and a distinct core set $C\subseteq V$. Each edge $e\in E$ has a capacity $b(e)$. For each object $o\in O$ and each node $c\in C$, a unique path P_{co} specifies the route from c to o. Each edge along P_{co} consumes bandwidth if object o is assigned to core c. It is NP-hard to determine the best λ such that for each object $o\in O$, $\lambda(o)$ gives the core it is assigned to, and the total bandwidth consumption is minimal subject to the capacity constraint on each link.

Proof: We reduce the set-covering problem to our Core Selection problem. In the set-covering problem, an instance (X,F) consists of a finite set X and F a set of subsets of X, such that every element of X belongs to at least one subset in F. Each subset $f_i\in F$ has cost c_i. The problem is to find a minimum cost subset $C\subseteq F$ whose members cover all of X. The total cost is $d(=\sum_{fi\in C} c_i)<\sum_{fi\in F} c_i$.

Given an instance of set-covering problem, we create an instance of our Core Selection problem: for each subset $f_i\in F$, create a potential core $p_i\in P$, and for each element $x_j\in X$, create an object $o_j\in O$. For each potential core p_i, lets c_i be the total bandwidth consumed for streaming to its connected objects.

Clearly, if Core Selection problem has solution with d total bandwidth consumption, each element in X is covered by set C exactly once with total cost d. On the other hand, if set-covering problem has solution with total cost d, a corresponding optimal Core Selection solution can be constructed by assigning an object to exactly one core. Therefore, our Core Selection problem is NP-hard.

III. HEURISTIC ALGORITHM FOR CORE SELECTION

As proved in the previous section, it is impossible to solve the core selection problem in reasonable time. Therefore in this section, we propose an intuitional heuristic-based algorithm. Since we are unable to obtain an integer optimal solution for even small sized network in reasonable time, we use the Linear Programming (LP) relaxation as a crude lower bound to evaluate our greedy heuristic.

Our greedy algorithm consists of three steps.

Step 1: find the set of potential cores which can receive original stream from source directly.

Step 2: pick greedily other core if there are any objects that are not assigned yet to any core. Our heuristic is to choose a core such that a source-core pair and a core-object pair consume the smallest bandwidth without violating bandwidth constraints.

Step 3: try to reassign the object of the new core to the prior cores, if the overall bandwidth usage can be reduced when compared to the previous assignment. Care is taken to ensure that bandwidth constraints are not violated.

In the second step, for example on figure 2, if $s - c_1$ pair and $c_1 - o_1$ pair have the smallest bandwidth usage among all the source-core and all the core-object pairs, the object o_1 is assigned to the core c_1. If $s - c_1$ pair and $c_2 - o_2$ pair have the next smallest bandwidth usage, object o_2 have to be assigned to the core c_2. However, in the third step, if the overall bandwidth usage when o_2 is assigned to the prior core c_1 is smaller than when o_2 is assigned to c_2, o_2 is assigned to c_1. If object o_3 is not assigned to the core c_1 due to the bandwidth constraints, object o_3 is assigned to the core c_2.

We analyze the complexity of the heuristic as follows: At first, in order to find the set of potential cores, we have to find nodes to which the original stream can be transmitted. Since we assume that all paths between a source and all other nodes are pre-calculated, we should test all the paths between the source and all other nodes whether they have enough bandwidth for the transmission of the original streaming. If we let the number of nodes be n, this step takes $O(n)$ [5]. Then, in order to find the smallest bandwidth usage among all the potential source-core and potential core-object pairs, we should iterate $|C||O|$ times, where $|C|$ is the number of potential cores, and $|O|$ is the number of objects. Therefore this step takes $O(|C||O|) = O(n^2)$. Taking into account all three steps, the total running time for our proposed algorithm is $O(n^3)$ if there are kn objects.

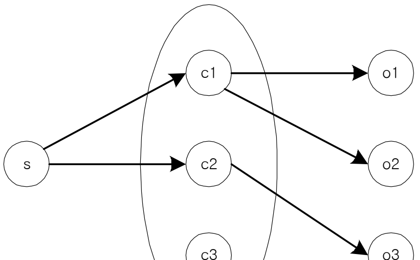

Figure 2. Object o_1 and o_2 are assigned to core c_1 and object o_3 is assigned to core c_2.

IV. SIMULATION RESULTS

In this section, we evaluate the performance of our heuristic algorithm on a hierarchical topology. For simplicity, we make the reasonable assumption of shortest path routing.

We generate hierarchical network topology using the GT-ITM topology generator [6] with network size of 200 nodes and evaluate our algorithm in cases; there are 5, 10, 15 and 20 receivers respectively. Receivers are selected randomly among nodes which cannot receive the original streaming due to the shortage of available bandwidth. Also we assume that the residual bandwidth of each link is distributed between 1 and 10 units and the bandwidth requirement of each receiver is 2 units.

Since the hierarchical network topology is composed of metro and core networks, there are three types of links: (1) intra-core link, (2) intra-metro link, and (3) metro-to-core link. We assume that the bandwidths of intra-core link and metro-to-core link are large enough. It is reasonable assumption of today's provider networks [1].

The goal of our simulation is to see the effectiveness of our heuristic. Thus the performance metric is the total amount of bandwidth consumption in the topology.

Ideally we would like to see how far our solution is from the integer optimal solution. We are unable to obtain optimal solution for any reasonable sized network since our integer programming solver[7] could not compute the optimal solution even for a small sized network. Therefore we use the LP relaxation of our problem to compute a crude lower bound on the optimal solution. Then we compare our heuristic with the LP lower bound.

From Figure 3, we see that the heuristic algorithm performs better than the MD coding in source node. Bandwidth usage of the heuristic is 25% less than the MD coding in source node.

The bandwidth required for our heuristic is close to the lower bound. Bandwidth usage of the heuristic is 13% more than the lower bound.

These results show that we are able to reduce the total amount of bandwidth consumption using our heuristic algorithm in reasonable time.

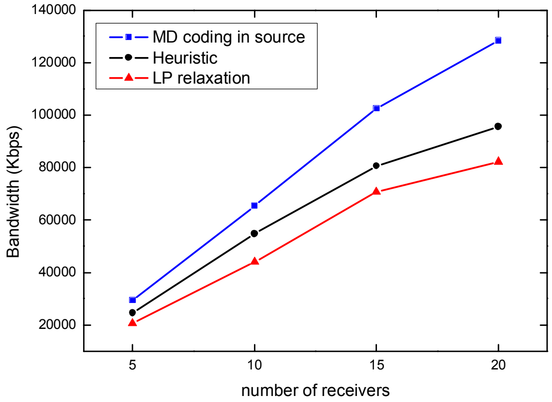

Figure 3. Bandwidth usage as number of receivers in the 200 node hierarchical network

V. CONCLUSION

Since MD coded streaming needs not only basic frames but also redundant informations, it consumes more bandwidth of about 20%. Therefore we delegate the role of the MD coding cores to some intermediate nodes, namely core nodes, in order to minimize the entire bandwidth consumption.

In this paper, we address the problem of the optimal core selection with respect to the overall bandwidth consumption. We prove that the problem is NP-hard, and propose a heuristic algorithm. In order to evaluate it, we compare the solution with LP-relaxed solution.

The heuristic algorithm shows good result in terms of the overall bandwidth consumption. Bandwidth usage of the heuristic is 25% less in the 200 node hierarchical network, and moreover our solution is close to the LP lower bound.

Our future work is to extend the scope of our problem formulation with mention of delay constraints. We will strive to reduce the start-up delay and transmission delay through simulations and implementations.

ACKNOWLEDGEMENT

This research was supported by the MIC (Ministry of Information and Communication), Korea, under the ITRC (Information Technology Research Center) support program supervised by the IITA (Institute of Information Technology Assessment). We are grateful to Dr. Venkateswaran, (Georgia Tech) for his kind and cooperative advice and feedback.

REFERENCES

[1] Jiancong Chen, Gary Chan, Victor O. K. Li, "Multipath Routing for Video Delivery Over Bandwidth-Limited Neworks", IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS, VOL. 22, NO. 10, DECEMBER 2004.

[2] V. Padmanabhan, H. Wang, P. Chou, "Resilient Peer-to-Peer Streaming", IEEE International Conference on Network Protocols, 2003.

[3] John G. Apostolopoulos, "Error-Resilient Video Compression via Multiple State Streams", International Workshop on Very Low Bitrate Video Coding, Kyoto, Japan, October 1999

[4] John G. Apostolopoulos, Tina Wong, Wai-t ian Tan, Susie Wee, "On Multiple Description Streaming with Content Delivery Networks", IEEE INFOCOM, 2002.

[5] T. H. Cormen, C. E. Leiserson, and R. L. Rivest, Introduction to Algorithms. Cambridge, MA: MIT Press, 1998.

[6] E. W. Zegura, K. Calvert, and S, Bhattacharjee, GT-ITM : Georgia Tech Internetwork topology models.
Available : http://www.cc.gatech.edu/projects/gtitm/

[7] GNU Linear Programming Kit (GLPK),
http://www.gnu.org/software/glpk

[8] J. G. Apostolopoulos, "Reliable video communication over lossy packet networks using multiple state encoding and path diversity", Visual Communications and Image Processing (VCIP), January 2001.

[9] M. Castro, P. Drushel, A. Kermarrec, A. Nandi, A. Rowstron, A. Singh, "SplitStream: high bandwidth content distribution in a cooperative environment", Proc. IPTPS(2003)

[10] V. Padmanabhan, H. Wang, P. Chou, K. Sripanidkulchai, "Distributed streaming media content using cooperative networking", Technical report MSR-TR-2002-37, 2002.

[11] A. Reibman, H. Jafarkhani, Y. Wang, M. Orchard, R. Puri, "Multiple description coding for video using motion compensated prediction", Proc. ICIP 3 (1999)

[12] HEAVY READING, "Assuring Quality of Experience for IPTV", White Paper, http://www.heavyreading.com.

[13] T. Nguyen, A. Zakhor, "Multiple sender distributed video streaming", IEEE Trans. Multimedia 6 (2004)

[14] X. Zhang, J. Liu, B. Li, T. Yum, "CoolStreaming/Donet: A data-driven overay network for peer-to-peer live media streaming", Proc. INFOCOM, 2005

[15] M. Zhang, L. Zhao, Y. Tang, J. Luo, S. Yang, "Large-scale live media streaming over peer-to-peer networks through global Internet", Proc P2PMMS, 2005

[16] X. Liao, H. Jin, Y. Liu, L. M. Ni, D. Deng, "AnySee: peer-to-peer live streaming", Proc. INFOCOM, 2006