
Volume 31, issue 3, 25 February 2008 ISSN 0140-3664
ELSEVIER
computer
communications
Available online at
ScienceDirect
www.sciencedirect.com
Special issue:
Disruptive Networking with Peer-to-Peer Systems
Guest Editors: Javed I. Khan
Adam Wierzbicki

Available online at www.sciencedirect.com
ScienceDirect
Computer Communications 31 (2008) 537–550
computer communications
www.elsevier.com/locate/comcom

Minimum DVS gateway deployment in DVS-based overlay streaming

Sang-Seon Byun [a,*], Chuck Yoo [b]

[a] Signal Processing Group, Department of Electronics and Telecommunications, Norwegian University of Science and Technology, N-7491 Trondheim, Norway
[b] Department of Computer Science and Engineering, Korea University, Seong-buk gu, An-am dong 5th street, Seoul, Republic of Korea

Available online 24 August 2007

Abstract

DVS (Distributed Video Streaming) is an edge streaming architecture for providing smooth video delivery. It is to divide video among multiple streaming senders in order to effectively provide the required throughput. In P2P-based streaming, if some peers are connected with high speed physical link, it is highly probable that they receive the best quality streaming stably from an original source. On these peers, we can impose the role of DVS gateway, i.e., let them translate the original streaming to distributive form and relay to the other peers who cannot connect to the original source with enough bandwidth. In dedicated infrastructure-based streaming, DVS gateway may be recruited by a content delivery company as an infrastructure node, that is to say, this infrastructure node has enough bandwidth to receive the best quality streaming from the original source, and it thus able to relay the original streaming using the distributed streaming to usual subscribers.

In this paper, we consider a minimum DVS gateway deployment problem in DVS-based overlay streaming with satisfying bandwidth requirement of every peer or subscriber. We also consider the mandatory diversity and limitations of DVS gateways' streaming capacity. The mandatory diversity is beneficial to mitigate the degradation of quality when an available bandwidth decreases suddenly, and the streaming capacity of each DVS gateway may be limited and heterogeneous. Based on recent advances in modeling techniques in flow networks, we provide a mixed integer programming formulation of the minimum DVS gateway deployment problem, and show that this problem is NP-hard. Thus we propose Lagrangian Relaxation with reduction heuristic to obtain an approximated solution. Our theoretical simulation studies show that the results obtained by our method are close to the lower bound obtained using LP relaxation.

Keywords: Distributed Video Streaming; Path diversity; Overlay streaming; P2P streaming; Lagrangian Relaxation

1. Introduction

DVS (Distributed Video Streaming) [1,2] is an alternative to edge streaming architecture for providing smooth video delivery and to divide video among multiple streaming senders in order to effectively provide the required throughput (Fig. 1). Having multiple senders is in essence a diversification scheme in that it combats unpredictability of congestion in the Internet. Specifically, it assumes independent routes from various senders to the receiver, and argues that the chances of all routes experiencing bursty packet loss at the same time are quite small. The measurement study [18] comparing the paths between two hosts on the Internet found that "in 30–80% of the cases, there is an alternate path with significantly superior quality" to the default path. The resilient overlay network (RON) [19] also demonstrates the existence of many sufficiently disjoint paths between two nodes on the Internet.

If the route between a particular sender and a receiver experiences congestion during streaming, the receiver can redistribute rates among the existing senders, or recruit new senders so as to minimize the effective loss rate and satisfy the throughput demanded by the receiver. In addition, it is natural that a content delivery company may deploy DVS architecture for plenty of subscribers. If new subscribers enter the streaming, it may be required to recruit new senders to comply with each subscriber's demand rate, and if subscribers leave, one or more DVS

* Corresponding author. Tel.: +47 73593418; fax: +47 73593628.
E-mail addresses: ssbyun@gmail.com (S.-S. Byun), hxy@os.korea.ac.kr (C. Yoo).

0140-3664/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.comcom.2007.08.018

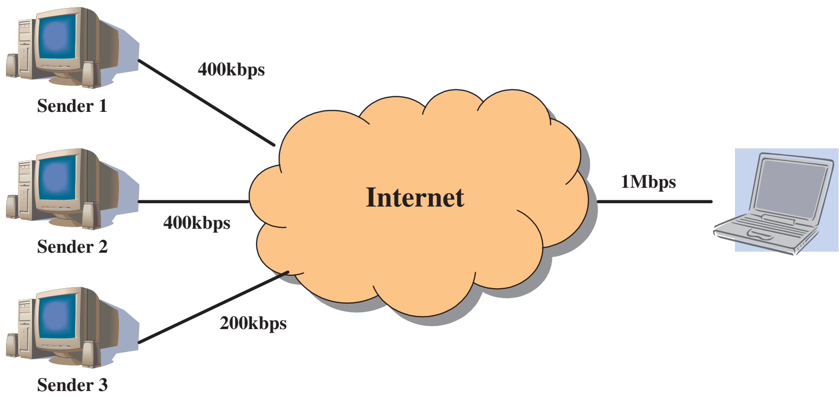


Fig. 1. Distributed Video Streaming architecture.

senders may be dropped from streaming to reduce deployment cost. Additionally, each subscriber may connect more than two DVS senders by compulsion in order to prevent from sudden quality degradation, and each sender's streaming capacity is limited.

If a subscriber (or receiver) receives streaming video from more than two senders and the route between a particular sender and the subscriber experiences congestion, the subscriber can have an opportunity to redistribute rates among the existing senders. Multiple Description Coding (MDC) approach is appropriate for the efficient rate redistributions and it is congruent with the DVS, which has been studied in detail in [3–7]. In the MDC approach, a video source is partitioned into multiple descriptions, each one assumed to be sent along a different channel. The assumption is that visual quality of the video degrades gracefully as the number of received descriptions decrease due to the channel impairments. Thus if a subscriber receives an MDC encoded streaming from more than two senders, the congestion of the path from a particular sender does not lead to the service disruption. It results in only quality degradation.

If the DVS architecture is employed in an overlay streaming (i.e., *DVS-based overlay streaming*), we increase the tolerance to packet loss and delay due to network congestion. In addition, the DVS employs an elegant rate allocation algorithm (RAA) and a packet partition algorithm (PAA). The RAA, run at the subscriber, is used in conjunction with FEC (Forward Error Correction) to minimize the probability of packet loss in bursty channel environments by splitting the sending rates appropriately across the senders. The PPA, run at the senders based on a set of parameters estimated at the receiver, ensures that every packet is sent by one and only one sender, and at the same times, minimizes the startup delay. In the DVS architecture, senders and receiver communicate with each other through control packet that contains the required information to facilitate the RAA and PPA, whose detailed operations are described in [2].

The overlay streaming architecture is classified into *P2P (Peer-to-Peer)-based* and *dedicated infrastructure-based* [15]. The DVS-based overlay streaming is applicable to both of these overlay streaming architectures. When streaming video spread through P2P-based approach, it is likely that some peers may receive the best quality streaming media stably if they are connected to the original source with enough bandwidth. On the other hand, some peers may experience quality degradation due to relatively lower bandwidth and frequent congestion. Under these circumstances, we impose the role of *DVS Gateway* (DG) on the peers who are connected to the original source or other DG with enough bandwidth to receive high quality video, i.e., let the DGs translate the original streaming to distributive from and relay to the other peers who cannot connect to the original source directly with satisfying their rate requirements. In dedicated infrastructure-based approach, the DG can also be employed by a content delivery company as a dedicated-infrastructure, that is to say, this infrastructure node has enough bandwidth to receive the best quality streaming from the original source, and it thus able to relay the original streaming using the distributed streaming to usual subscribers. In this paper, we deploy DVS approach for tree-based streaming architecture only. Thus in the rest of this paper, we do not distinguish the term 'peer' of P2P-based streaming architecture with the term 'subscriber' of dedicated infrastructure-based streaming, and we use the term 'subscriber' only.

We consider three kinds of minimum DG problem in the DVS-based overlay streaming architecture under a given environment – number of DG candidates, number of subscribers, each subscriber's requiring bandwidth and available bandwidth of each link. If too many DGs are employed for distributed streaming, it is highly probable that links – which are close to original source or some DGs – experience congestion, and also the high deployment cost may be an obstacle to the company who intends to employ DGs as the dedicated infrastructure. For all that, if DGs are connected in one or a few lines in order to reduce bandwidth overhead, subscribers which are not selected as a DG suffer long delay and the diversity deteriorates.

We first show that the problem of minimizing the number of DGs in a given network environment is NP-hard. In addition, we show that the minimum DG problems with mandatory diversity or streaming capacity limitations of DG candidates are also NP-hard. When we deploy the DVS-based overlay streaming architecture on P2P-based streaming environment, a subscriber selected as a DG may be dropped either voluntarily or involuntarily. In case DGs are recruited by a content delivery company, it is possible that a DG server is crushed. In addition, it is probable that some DGs suffer the lack of bandwidth on the path from the original source, or new subscriber shows up. In these cases, the set of DGs should be reconfigured as fast as possible in order to provide required throughput to all subscribers. Thus, based on recent advances in modeling techniques in flow networks, we provide a mixed integer programming formulation of the minimum DG problem, and provide good solutions to this problem by using *Lagrangian Relaxation* and reduction heuristic in reasonable time. Based on theoretical simulation studies, we show that our results are close to the LP relaxed lower bound by a factor of 11 or under.

This paper is organized as follows. In Section 2, a comparison with some related work and the DVS-based overlay streaming models are given. In Section 3, we provide the integer programming formulation for finding the minimal set of DGs in a given network topology, as well as the assignments of individual subscribers to the DG set, subject to each subscriber's bandwidth requirement, bandwidth constraint of each link, mandatory diversification and streaming capacity limitation. We show that these problems are all NP-hard. In Section 4, we adapt Lagrangian Relaxation to the mixed integer programming and propose a reduction heuristic for better solution with Lagrangian Relaxation. Section 5 provides evaluation results of our Lagrangian Relaxation with reduction heuristic algorithm on various network size and concluding remarks are given in Section 6.

2. The DVS-based overlay streaming model

In this section, we compare the DVS-based overlay streaming with previous honorable distributed streaming architecture, and describe the how the DVS architecture can be employed on overlay streaming. The important thing is that the purpose of this paper is not to argue that the DVS-based method is superior to the previous source-based distributed streaming architecture. Our main objective is to introduce a distributed streaming model that deploys DVS architecture and to minimize the number of DG.

Recently, streaming over multiple paths to provide path diversity has emerged as an approach to help overcome the problems induced by best-effort IP networks. SplitStream [16] and CoopNet [17] are renowned source-based and tree-based distributed streaming architectures. In these methods, the only original source performs distributed streaming, and other subscribers or intermediate nodes simply relay the distributed stream. The advantage of the source-based distributed streaming is that it is self-scaling; meaning as more subscribers join more bandwidth is supplied. However, original source should take responsibilities by itself rate alteration on each path. CoolStreaming/DONet [21], GridMedia [22] and AnySee [23] are gossip-based and mesh-based streaming architectures.

DVS-based approach has relatively lower scalability to the source-based distributed streaming methods since only subscribers or infrastructure servers, which are able to receive best quality streaming video, is selected as DGs. In addition, the links close to the original source may suffer congestion.[1] However, the DVS-based approach allows all DGs perform distributed streaming, thus rate redistribution is performed more easily, and parent discovery process can be done faster. For instance, if node A that performs relaying in the source-based method is dropped suddenly, a child of node A should discover other parents who relay the same frame packets with the ones which node A relayed in order to maintain its requiring throughput. On the other hand, DVS-based approach allows the child have the chance to redistribute rates among its currently connected DGs or to request new connection to other existing DGs.

We assume that the bandwidth bottleneck is not at the last hop. Our premise is that asymptotically, as broadband connections to the Internet such as DSL become prevalent, the limiting factor in streaming is packet loss and delay due to congestion along the streaming path, rather than the physical bandwidth limitation of the last hop. We also assume that if k DGs are needed to met the demand rates of all receivers, the original source has enough capacity and bandwidth to encode and transmit the best quality stream to these k DGs. We make this assumption for sake of simplicity. If this assumption is not satisfied, some subscribers are not free from quality degradation, or source-based distributed streaming method should be deployed instead. We consider that every subscriber cannot be satisfied its demand throughput without deploying distributed streaming architecture.

As described in Section 1, we apply two kinds of DVS-based overlay streaming model. One is to select DGs among existing subscribers, and the other is to select DGs among dedicated infrastructure node. The former is denoted as P2P-based model and the latter is denoted as dedicated infrastructure model (Fig. 2).

The P2P-based model has the following characteristics.

(1) It is assumed that subscribers who are dedicated as DG candidates have enough downstream bandwidth from original source to receive the best quality stream.

[1] We minimize the number of DG in order to mitigate this shortcoming also.

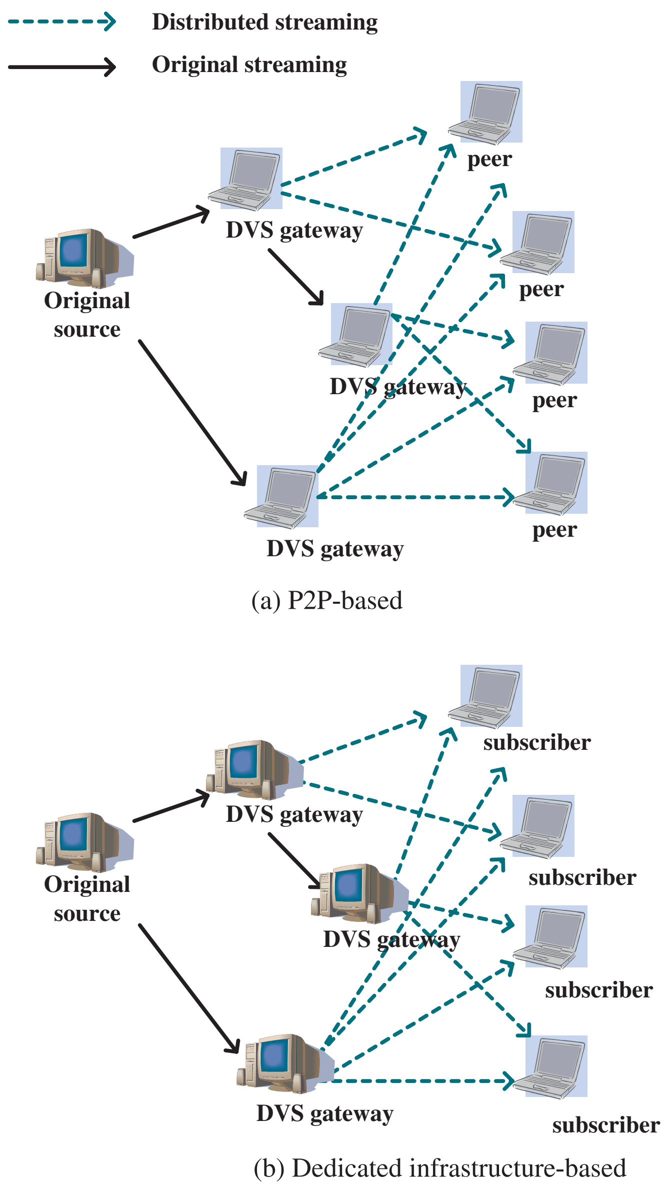


Fig. 2. DVS-based overlay streaming model.

(2) The subscribers who are not dedicated as DG candidates receive streams distributively from other subscribers who are selected as DGs.
(3) The DG candidates who are not selected as DGs receive undistributed stream from original source. This characteristic makes the distribution tree inefficient from the aspect of bandwidth consumption but shortness, interactivity and diversity are prioritized. The short and high diverse distribution tree is also employed in [17,20].

We now list the characteristics of the dedicated infrastructure-based model.

(1) A content delivery company brings in dedicated infrastructure servers and employs these servers as DG candidates.
(2) Each DG candidate has enough downstream bandwidth from original source to receive the best quality stream.
(3) Every subscriber receives streams distributively from the dedicated infrastructure servers that are selected as DGs.

3. System model and computational complexity

We solve three cases of minimum DG problem on the DVS-based overlay streaming model in Section 2. The first is to solve the minimum set of DGs subject to bandwidth constraint of each link and each subscriber's requiring rate. We denote this problem as *MinDG-Pure* problem. The second is to solve the minimum set of DGs subject to *mandatory diversification* constraint as well the constraints of *MinDG-Pure* problem. We denote this problem as *MinDG-MD* problem. The last is to achieve the minimum subset of DGs subject to heterogeneous *capacity limit* of every DG as well the constraints of *MinDG-Pure* problem. This problem is denoted as *MinDG-CL* problem. Notice that there is no difference of the problem formulation between configuring minimum DG set over P2P-based model and dedicated-infrastructure model.

The *MinDG-Pure* problem considers only the bandwidth demand of each subscriber and available bandwidth of each link. There is no diversity condition and it also assumes that the streaming capacity of each DG is unlimited. The *MinDG*-MD problem considers the mandatory diversity in order to alleviate the abrupt quality degradation that comes from a DG crash or bandwidth fluctuations. For example, if a subscriber receives streaming from two DGs distributively, and if one of the DGs experiences sudden congestion or crashes, then the subscriber has a chance to redistribute its requiring bandwidth to the other DG. The *MinDG-MD* problem also considers the constraints of the *MinDG-Pure* problem. The *MinDG-CL* problem considers the heterogeneous streaming capacity of each DG. The streaming capacity is determined by the process power of each DG. Also each subscriber can cope with the bandwidth fluctuations with requiring surplus bandwidth to its DG.

Prior to the formal statement of these problems, we define several denotations. Let $a(e)$ and $b(e)$ represent the amount of the actual used bandwidth for transmission of streaming media and the amount of available bandwidth at link e, respectively. And let $r(c)$ and $w(u,c)$ denote the amount of the demand bandwidth of subscriber c and the amount of subscriber c's requiring bandwidth assigned to DG u, respectively. Now the *MinDG-Pure* problem is stated formally as follows: given a network $G = (V,E)$, let $S \subseteq V$ and $C \subseteq V$ be the set of DG candidates and the set of subscribers who are not DG candidates. Determine (1) a minimum subset of nodes $U \subseteq S$ which to be selected as DGs such that the bandwidth constraint on every link $a(e) \leqslant b(e)$ is satisfied. (2) a mapping Ωwhich maps a subscriber to its DG set with satisfying the required bandwidth. That is, for each subscriber c_i, if $\Omega(c_i) = \{u_1, \ldots, u_n\} \subseteq U$, then subscriber c_i is assigned to the DG set $\{u_1, \ldots, u_n\}$, with satisfying constraint $\{w(u_1, c_i) + \cdots + w(u_n, c_i)\} \geqslant r(c_i)$.

If subscriber c_j should receive their streaming from more than div_j DGs, and we denote the DG set for subscriber c_j is U_i, for the *MinDG-MD* problem, the constraint (2) is rewritten as follows: a mapping Ω which maps a subscriber to its DG set with satisfying the required bandwidth. That is, for each subscriber c_i, if $\Omega(c_j) = U_i$, then subscriber c_j is assigned to the DG set $U_i = \{u_1, \ldots, u_n\}$ with constraints $\{w(u_1, c_j) + \cdots + w(u_n, c_j)\} \geqslant r(c_j)$ and $w(u_i, c_j) \leqslant (r(c_j)/\text{div}_j)$, for all $u_i \in U_i$.

Let p_i be the capacity of DG u_i. For the *MinDG-CL* problem, the constraint (2) is rewritten as follows: a mapping Ω which maps a subscriber to its DG set with satisfying the required bandwidth. That is, for each subscriber c_j, if $\Omega(c_j) = U_i$, then subscriber c_j is assigned to the DG set $U_i = \{u_1, \ldots, u_n\}$ with constraints. $\{w(u_1, c_j) + \cdots + w(u_n, c_j)\} \geqslant r(c_j)$ and for each DG u_i, $p_i \geqslant \sum_{cj \in C} w(u_i, c_j)$.

In this paper, we assume that the routes between the DG candidate and the subscriber are fixed and symmetric.

3.1. Mixed integer programming formulation

Given our assumption that routes are fixed between any given DG-subscriber pair, the formulations of *MinDG-Pure* problem can be cast into a nonlinear programming formulation.

Let S and C present the set of DG candidates and the set of subscribers.

The notations for the programming formulation are given as follows:

- x_i binary, indicates a node i is selected as a DG.
- w_{ij} a subscriber j's receiving bandwidth from a DG i.
- r_j the demanding rate of a subscriber j.
- λ_{ij}^e binary, indicates that an edge e lies on the path between a node i and node j.

Our objective is:

$$\text{Minimize} \sum_{i \in S} x_i \quad (1)$$

$$\text{s.t.} \sum_{i \in S} w_{ij} = r_j, \quad \text{for each } j \in C \quad (2)$$

$$\text{s.t. } w_{ij} \leqslant x_i w_{ij}, \quad \text{for each } i \in S,\ j \in C \quad (3)$$

$$\text{s.t.} \sum_{i \in S} \sum_{j \in C} w_{ij} \lambda_{ij}^e \leqslant b(e), \quad \text{for each } e \in E \quad (4)$$

$$\text{s.t. } x_i \in \{0, 1\}, \quad \text{for each } i \in S \quad (5)$$

$$\text{s.t. } w_{ij} \geqslant 0, \quad \text{for each } i \in S,\ j \in C \quad (6)$$

The first constraint makes sure that each subscriber j receives the amount of its requiring rate from the DGs. The second constraint ensures that a node i must be a DG if the node i transmits w_{ij} amount of bandwidth to subscriber j. The third constraint guarantees that the sum of the bandwidth used by all the DG subscriber pairs on each link does not exceed its available bandwidth. Since the second constraint is nonlinear, the minimum DG in DVS problem becomes a nonlinear integer programming problem. In general, nonlinear integer programming problems are very hard to solve and, in fact, no general solution approach is known [3]. Thus we convert this problem to linear form.

Generally, demanding rate and available network bandwidth change discretely. Thus a bandwidth can be expressed as a multiple of unit bandwidth (module). 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 subscriber and available bandwidth of each link are expressed as a multiple of unit bandwidth. We also assume that if a subscriber requires m times of unit bandwidth, the subscriber is split into m objects and each object requires a unit bandwidth. By this assumption, we can formulate the minimum DG problem as mixed integer programming, and prove as NP-hard. To begin with, we define two terms. We define set of subscriber objects that compose a subscriber as a *cluster*, and a representative subscriber object of each subscriber as a *cluster head*.

Now we can give the MIP (Mixed Integer Programming) formulation of the *MinDG-Pure* problem with the assumptions presented above. Let Γ present the set of subscriber objects at entire subscribers. The additional variables for the MIP are given as follows:

- y_{ij} binary variable, indicates that a subscriber object j is assigned to a DG i.

Our objective is:

$$\text{Minimize} \sum_{i \in S} x_i \quad (7)$$

$$\text{s.t.} \sum_{i \in S} y_{ij} = 1, \quad \text{for each } j \in \Gamma \quad (8)$$

$$\text{s.t. } y_{ij} \leqslant x_i, \quad \text{for each } i \in S,\ j \in \Gamma \quad (9)$$

$$\text{s.t.} \sum_{i \in S} \sum_{j \in \Gamma y_{ij} \lambda_{ij}^e} \leqslant b(e), \quad \text{for each } e \in E \quad (10)$$

$$\text{s.t. } x_i \in \{0, 1\}, \quad \text{for each } i \in S \quad (11)$$

$$\text{s.t. } y_{ij} \in \{0, 1\}, \quad \text{for each } i \in S,\ j \in \Gamma \quad (12)$$

The first constraint ensures that each subscriber object j is assigned to exactly one DG. The second constraint indicates that a node i must be a DG if a subscriber object is assigned to it. The third constraint guarantees that the sum of the bandwidth used by all the ⟨DG – subscriber object⟩ pairs on each link does not exceed its available bandwidth.

For the *MinDG-MD* problem, two variables and one constraint are added as follows:

s_{kj} binary, indicate that a object k belongs to the cluster of a subscriber j.
div_k the diversity requirement of subscriber k.

$$\text{s.t.} \sum_{j \in \Gamma} s_{kj} y_{ij} \leqslant \sum_{j \in \Gamma} s_{kj} \Big/ \text{div}k, \quad \text{for each } k \in C,\ i \in S \tag{13}$$

This constraint ensures that the amount of rate from a DG i to a subscriber k must not exceed subscriber ks requiring rate/div_k. Thus every subscriber receives their streaming data from at least div_k DGs.

For the *MinDG-CL* problem, one variable and one constraint are added as follows:

p_i the capacity of an DG i.

$$\text{s.t.} \sum_{j \in \Gamma} y_{ij} \leqslant p_i, \quad \text{for each } i \in S \tag{14}$$

This constraint guarantees that the amount of assigned rate to a DG i should not exceed its capacity.

3.2. Time complexity of the minimum DG problems

In this subsection, we prove that the *MinDG-Pure*, *MinDG-MD* and *MinDG-CL* problem are all NP-hard.

Theorem 1. *Computation of the optimal solution to the MinDG-Pure problem is NP-hard.*

Proof. The *MinDG-Pure* problem is proven to be NP-hard via a reduction from the k-median problem, which is stated as follows. Given the set of city J, the number of available facilities k, connection cost c_{ij} between a facility i and a city j and a total cost $d < \sum_{j \in J} c_{ij}$ for an arbitrary facility i, connect each city and facility that meets the total cost.

Given an instance of the k-median problem, we create an instance of the *MinDG-Pure* problem. For each city $j \in J$, create a subscriber node that demands a unit bandwidth. We denote these nodes as R. Now pair-wise connect these nodes with links that have an available bandwidth of $|J|$. Such a construction results in a clique of size of $|J|$. For each facility in the k-median problem, create a new cluster whose size is $\sum_{j \in J} c_{ij}$, for arbitrary node i, and connect each cluster head to one or more arbitrary nodes in R with new links that has $d/|J|$ amount of available bandwidth. We use S to represent these cluster heads. Every node in R should be assigned to exactly a node in S. This step creates $k \cdot \sum_{j \in J} c_{ij}$ additional nodes and $|J|$ additional links.

Clearly, any node in S can only be serviced by itself or the node in its own cluster since its only available total bandwidth from other nodes cannot exceed d which is less than its own requirement $\sum_{j \in J} c_{ij}$. Therefore, the solution to this *MinDG-Pure* instance has at least k DGs. If the *MinDG-Pure* problem has exactly k DGs, every cluster must have one DG with the total bandwidth consumption d. Consequently, if we assume that only cluster heads can be DG, all nodes in S and only nodes in S are DGs. Since every node in R is serviced by a node in S, the DG-subscriber relation between nodes in S and R gives a k-median solution that satisfies the total cost d. On the contrary, suppose that the k-median problem has a solution, then a k DG solution can be constructed as follows: for a node $r \in R$, find the facility on which the city represented by r is connected, and let r be serviced by the node $s \in S$ that represents that facility. This completes the proof (Fig. 3). □

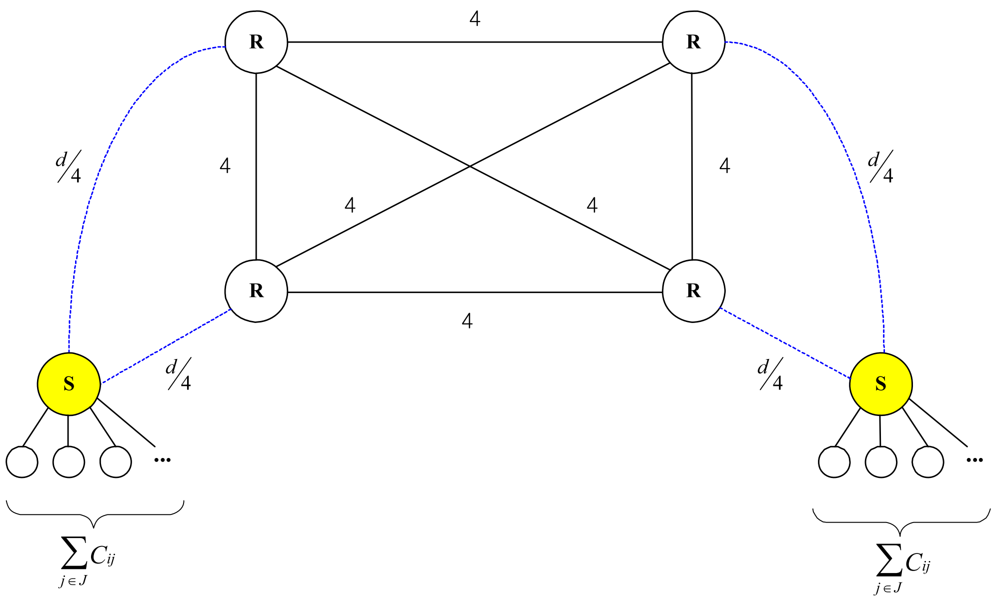


Fig. 3. Reduction from k-median problem in case $k = 2$.

The *MinDG-MD* problem is reducible from SMC (Set Multi Cover) problem [8].

Theorem 2. *Computation of the optimal solution to the MinDG-MD problem is NP-hard.*

Proof. The SMC problem is stated formally as follows: Given an universal set U, a set S that is a set of subsets $S = \{S_1, \ldots, S_k\}$ of U and c_i the cost of subset S_i, find a set $C \subset S$ that meets cost $d < \sum_{S_i \in S} c_i$ and each element of U is covered at least $r_i > 0$ times by C.

Given an instance of the SMC problem, we create an instance of *MinDG-MD* problem. For each subset $S_i \in S$, create a DG candidate, and for each element $u_j \in U$, create a subscriber node. Let each subscriber u_j's demanding rate be d_j. Then choose r_j DGs among S randomly and connect each subscriber u_j to r_j chosen DGs with links that have bandwidth of d_j/r_j. In addition, for each DGs S_i, let c_i be the total bandwidth used for streaming to its connected subscribers. Thus

$$c_i = \sum_{u_j \in S_i} (d_j / r_j) \tag{15}$$

$$d = \sum_{u_j \in U} d_j \tag{16}$$

$$d = \sum_{S_i \in C} \sum_{u_j \in S_i} (d_j / r_j) = \sum_{S_i \in C} c_i \tag{17}$$

We use Q_j to represent the chosen DGs for u_j.

Clearly, all DGs in Q_j should stream to subscriber u_j with the sending rate of d_j/r_j in order to satisfy u_j's demanding rate d_j. Thus if every subscriber u_j is assigned to exactly r_j DGs with the total bandwidth consumption d, we can find the set C which is the solution of the SMC such that each u_j is covered r_j times by subsets in C with the total cost d using Eqs. (15)–(17). On the other hand, if every element in u_j is covered r_j times by subsets in C with the total cost d, we can find the set of DGs C such that every subscriber u_j is assigned to r_j DGs with the total cost d. Thus *MinDG-MD* problem is NP-hard.

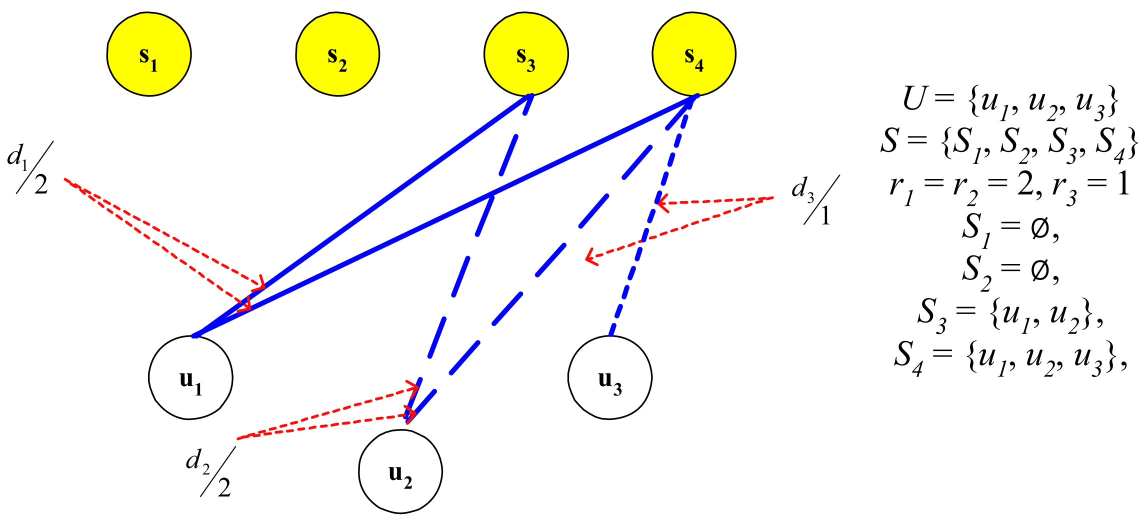
The *MinDG-CL* problem is proven to be NP-hard via a reduction from the capacitated k-median problem [9]. The capacitated k-median problem has an additional constraint to the k-median problem that each facility has non-uniform capacity (Fig. 4). □

Theorem 3. *Computation of the optimal solution to the MinDG-CL problem is NP-hard.*

Proof. The capacitated k-median problem is stated formally as follows: Given the set of city J, the number of available facilities k, connection cost c_{ij} between a facility i and a city j, capacity limit of facility i u_i and a total cost $d < \sum_{j \in J} c_{ij}$ for an arbitrary facility i, connect each city and facility that meets the total cost and capacity limit of each facility.

Given an instance of the capacitated k-median problem, we create an instance of *MinDG-CL* problem. For each city $j \in J$, create a subscriber object that requires unit bandwidth and pair-wise connect these subscriber objects with links that have an arbitrary bandwidth. We use R to denote these subscriber objects. Such a construction results in a clique of size $|J|$. For each facility in the capacitated k-median problem, create a subscriber whose bandwidth demand is unit bandwidth, and connect this node to an arbitrary subscriber object in R with a new link that has $d/|J|$ amount of bandwidth. We denote these subscribers as S. Every subscriber object in R should be assigned to exactly a subscriber in S. We assume that for each $r_j \in R$ and $s_i \in S$, $u_j <$ (unit bandwidth) and $u_i = |J|/k$.

Clearly, any subscriber in S can only be served by itself since its connected subscriber objects have capacity limit



(a) *MinIDS-MD* problem (b) SMC problem

Fig. 4. Reduction from SMC problem.

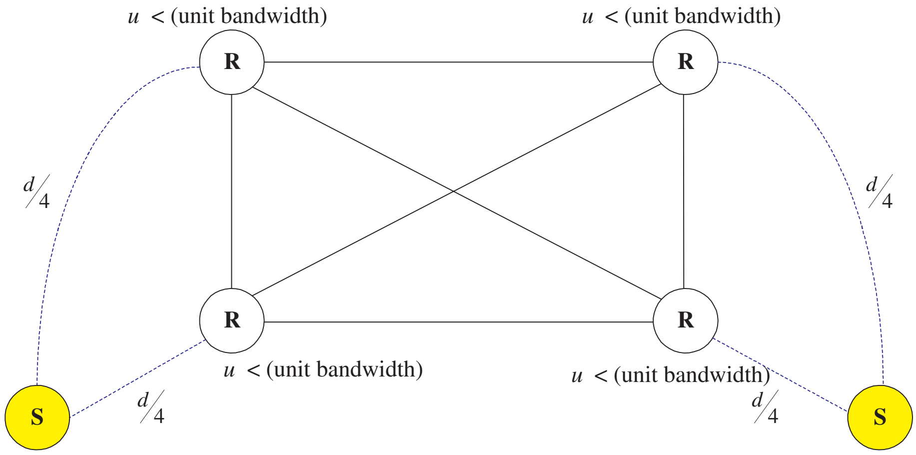


Fig. 5. Reduction from capacitated k-median problem in case $k = 2$.

$u_j <$ (unit bandwidth), which is less than its own demand, unit bandwidth. Therefore, the solution to this *MinDG-CL* instance has at least k DGs. Consequently, if the *MinDG-CL* problem has exactly k DGs, all subscribers in S and only subscribers in S are DGs, with the total bandwidth consumption $d$and satisfying the capacity condition. On the other hand, suppose that the capacitated k-median problem has a k-facility that satisfies the total cost d, and capacity condition, then a $k-$ DG solution can be made as follows: for a $r \in R$, find the facility on which the subscriber object represented by r is assigned, let r be served by the node $s \in S$ that represents that facility. This completes the proof (Fig. 5). □

4. Lagrangian Relaxation and reduction heuristic

4.1. Lagrangian Relaxation

Lagrangian Relaxation [10] has proved to be a useful tool in various combinatorial optimization problems. In this approach, sets of constraints are relaxed and dualized by adding them to the objective function with penalty coefficients, the Lagrangian multipliers. The objective of the relaxation is to dualize, possibly after a certain amount of remodeling, the constraints linking the component together in such a way that the original problem is transformed into disconnected and easier to solve subproblems. One is able in this way to obtain bounds on the actual integer optimal value, and separate solutions to the individual subproblems which, while not necessarily consistent because they may violate some of the linking constraints, might however suggest ways of constructing good globally feasible solutions.

We relax the second constraint (Eq. (9)) of the original problem. Then the Lagrangian function is written using the *dual multipliers* π associated only with this constraint and the original objective function as follows:

$$L(\mathbf{x},\mathbf{y};\boldsymbol{\pi}) = \sum_{i\in S} x_i + \sum_{i\in S}\sum_{j\in\Gamma} \pi_{ij}(y_{ij} - x_i) \quad (18)$$

$$L(\mathbf{x},\mathbf{y};\boldsymbol{\pi}) = \sum_{i\in S}(1 - \sum_{j\in\Gamma}\pi_{ij})x_i + \sum_{i\in S}\sum_{j\in\Gamma}\pi_{ij}y_{ij} \quad (19)$$

Now, if we try to minimize the Lagrangian with respect to primal variables **x**, **y** in regard to the remaining constraints (Eqs. (8), (10)–(12)), we arrive at a following new problem in variables **x** and **y**.

$$\text{Minimize} L(\mathbf{x},\mathbf{y};\boldsymbol{\pi}) = \sum_{i\in S}(1 - \sum_{j\in\Gamma}\pi_{ij})x_i + \sum_{i\in S}\sum_{j\in\Gamma}\pi_{ij}y_{ij} \quad (20)$$

$$\text{s.t.} \sum_{i\in S} y_{ij} = 1, \quad \text{for each } j\in\Gamma \quad (21)$$

$$\text{s.t.} \sum_{i\in S}\sum_{j\in\Gamma} y_{ij}\lambda^e_{ij} \leqslant b(e), \quad \text{for each } e\in E \quad (22)$$

$$\text{s.t. } x_i \in \{0,1\}, \quad \text{for each } i \in S \quad (23)$$

$$\text{s.t. } y_{ij} \in \{0,1\}, \quad \text{for each } i\in S, j\in\Gamma \quad (24)$$

It is now easier to see the two decomposed problems, one over the variable **x** and the other over the variable **y**, which are certainly dependent on the assumed values of $\boldsymbol{\pi}$. The advantage of such decoupling is the ability to solve these subproblems efficiently. The original problem can be decomposed with the parameter $\boldsymbol{\pi}$ as follows:

$$W(\boldsymbol{\pi}) = \min_{\mathbf{x},\mathbf{y}} \begin{bmatrix} L(\mathbf{x},\mathbf{y};\boldsymbol{\pi}): \\ \text{s.t.} \sum_{i\in S} y_{ij} = 1, \ \text{for each } j\in\Gamma, \\ \text{s.t.} \sum_{i\in S}\sum_{j\in\Gamma} y_{ij}\lambda^e_{ij} \leqslant b(e), \ \text{for each } e\in E, \\ \text{s.t.} x_i \in \{0,1\}, \ \text{for each } i\in S, \\ \text{s.t.} y_{ij}\in\{0,1\}, \ \text{for each } i\in S, j\in\Gamma \end{bmatrix}$$
$$= \min_{\mathbf{x}} \begin{bmatrix} \sum_{i\in S}(1-\sum_{j\in\Gamma}\pi_{ij})x_i: \\ \text{s.t.} x_i\in\{0,1\}, \ \text{for each } i\in S \end{bmatrix}$$
$$+ \min_{\mathbf{y}} \begin{bmatrix} \sum_{i\in S}\sum_{j\in\Gamma}\pi_{ij}y_{ij}: \\ \text{s.t.} \sum_{i\in S} y_{ij} = 1, \ \text{for each } j\in\Gamma, \\ \text{s.t.} \sum_{i\in S}\sum_{j\in\Gamma} y_{ij}\lambda^e_{ij} \leqslant b(e), \ \text{for each } e\in E, \\ \text{s.t.} y_{ij}\in\{0,1\}, \ \text{for each } i\in S, j\in\Gamma \end{bmatrix} \quad (25)$$

These decoupled subproblems can be solved independently. The subproblem for variable $\mathbf{x}$ is solved easily since optimal x_i takes the value 1 if $(1-\sum_{j\in\Gamma}\pi_{ij})<0$, otherwise 0.

Then the generalized dual problem of the *MinDG-Pure* problem is:

$$\max_{\pi}[W(\boldsymbol{\pi}):\boldsymbol{\pi}\geqslant 0] \tag{26}$$

Therefore the result of Eq. (26) is the lower bound of the original problem. In addition, we can obtain the approximate feasible primal solution through the algorithm, which is generally called *Lagrangian Relaxation based dual algorithm*, and is shown in Table 1.

In the algorithm, the parameter *max_bw* and *total_link* mean that the largest bandwidth demand among all subscribers and the number of edges in a topology. We initiate the variables k_{max}, ρ_{max_iter}, ρ, ρ_{iter} and ρ_{min} in accordance with the ones that are defined in [10], and π^0 is filled with random value. As the iteration progresses, the feasible primal (F^{best}) and lower bound (F^{dual}) approach gradually to the integer optimal by adjusting $\boldsymbol{\pi}$ using Eq. (29). After the overall iteration ends, we regard the final value of F^{best} as the approximated optimal solution.

If the problem is well-behaved, then the optimal solution to Eq. (26) will be an optimal solution to the primal problem. However, this problem has the *integrality property* [11]. The integrality property describes a class of problems where the solution to the Lagrangian relaxed problem always has integer solutions even if the integer constraint is removed. The subproblem for $\mathbf{x}$ (Eq. (25)) has an integer solution whether the integer constraint – $x_i\in\{0,1\}$ – is dropped or not, since optimal x_i takes always the value 1 if $(1-\sum_{j\in\Gamma}\pi_{ij})<0$, otherwise 0. The consequence of the integrality property is that the solution to the dual problem can be no better than the solution to the LP relaxation.

Table 2
The heuristic algorithm for DG reduction

Step 0:	let the set of nodes who satisfy $\mathbf{x}_i^k=1$ be C
Step 1:	sort all nodes in set C in decreasing assigned bandwidth
	for each $i\in C$ {
Step 2:	Try to reassign all the bandwidth assigned to i to other nodes in C
Step 3:	**if** (all reassigned) {
	$C=C-i$;
	$\mathbf{x}_i^k=0$;
	}
	}

Table 3
Simulation parameters

Parameter	Value
Network size	40 nodes, 80 nodes, 120 nodes, 250 nodes
Link bandwidth	In case network size of 40: 800 kbps ~2 Mbps In case network size of 80: 2 Mbps ~5 Mbps In case network size of 120 & 250: 3 Mbps ~8 Mbps
Demand rate of each subscriber	200–800 kbps
Number of mandatory diverse DG	2
Capacity limit of each DG	1–6 Mbps
Number of DG candidates	In case network size of 40: 8 In case network size of 80: 16 In case network size of 120: 20 In case network size of 250: 35
Size of unit bandwidth	100 kbps

Despite having the integrality property there are several reasons for persevering with a solution using Lagrangian Relaxation. The main reason is that, while the dual problem may not converge to the optimal

Table 1
Lagrangian Relaxation based dual algorithm

Step 0:	Define $\boldsymbol{\pi}^0$, $k_{max}=100$, $\rho_{max\_iter}=7$, $\rho=2.0$, $\rho_{iter}=0$, $\rho_{min}=0.001$, $F^{best}=\infty$, $W^{upper}=max\_bw*\ total\_link$, $F^{dual}=-\infty$, $k=0$
Step 1:	$k=k+1$, $\rho_{iter}=\rho_{iter}+1$.
	Given $\boldsymbol{\pi}^k$, solve $W(\boldsymbol{\pi})$ as decoupled subproblems in $\mathbf{x}$ and $\mathbf{y}$ to obtain solutions $\bar{\mathbf{x}}^k$ and $\mathbf{y}^k$.
Step 2:	Use $\mathbf{y}^k$ to compute feasible $\mathbf{x}^k$ that satisfies $\mathbf{y}^k\leqslant\mathbf{x}^k$.
	Use $\mathbf{x}^k$ and $\mathbf{y}^k$ to compute primal objective F.
	if $(F<F^{best})F^{best}=F$, $\mathbf{x}^{best}=\mathbf{x}^k$, $\mathbf{y}^{best}=\mathbf{y}^k$, $W^{upper}=F^{best}$.
Step 3:	Use decoupled solutions $\bar{\mathbf{x}}^k$ and $\mathbf{y}^k$ to compute
	$\nabla W(\boldsymbol{\pi}^k)=(y_{00}^k-\bar{x}_0^k,y_{01}^k-\bar{x}_0^k,\ldots,y_{0j}^k-\bar{x}_0^k,\ldots,y_{ij}^k-\bar{x}_i^k)$ (27)
	dual objective: $W(\boldsymbol{\pi}^k)$, obtain using Eq. (25)
	step size: $tk=\rho(W^{upper}-W(\boldsymbol{\pi}^k))/\Vert\nabla W(\boldsymbol{\pi}^k)\Vert^2$
	dual variable: $\pi_{ij}^{k+1}=max\{\pi_{ij}^k+tk\nabla W(\pi_{ij}^k),0\}$
Step 4:	**if** $(W(\boldsymbol{\pi}^k)>F^{dual})$ $F^{dual}=W(\boldsymbol{\pi}^k)$
	else if $(\rho_{iter}>\rho_{max\_iter})$ $\rho=\max\{\rho/2,\rho_{min}\}$, ρ_{iter}=0
Step 5:	**if** $(t_k<0.0001)$ stop
	else go to Step 1

Table 4
Simulation results

	MinDG-Pure	*MinDG-MD*	*MinDG-CL*	*MinDG-MD & CL*
LP relaxation (40 nodes)	1.094	1.094	1.1043	1.1514
LR primal (40 nodes)	5	6	6	6
LR with reduction (40 nodes)	**3**	**4**	**3**	**5**
Integer optimal (40 nodes)	3	–	–	–
LP relaxation (80 nodes)	1	1	1	1
LR primal (80 nodes)	2	11	11	13
LR with reduction (80 nodes)	**2**	**4**	**2**	**10**
LP relaxation (120 nodes)	1	1	1	1
LR primal (120 nodes)	5	15	15	15
LR with reduction (120 nodes)	**2**	**4**	**2**	**11**
LP relaxation (250 nodes)	1	1	1	1
LR primal (250 nodes)	7	18	20	25
LR with reduction (250 nodes)	**3**	**8**	**3**	**20**

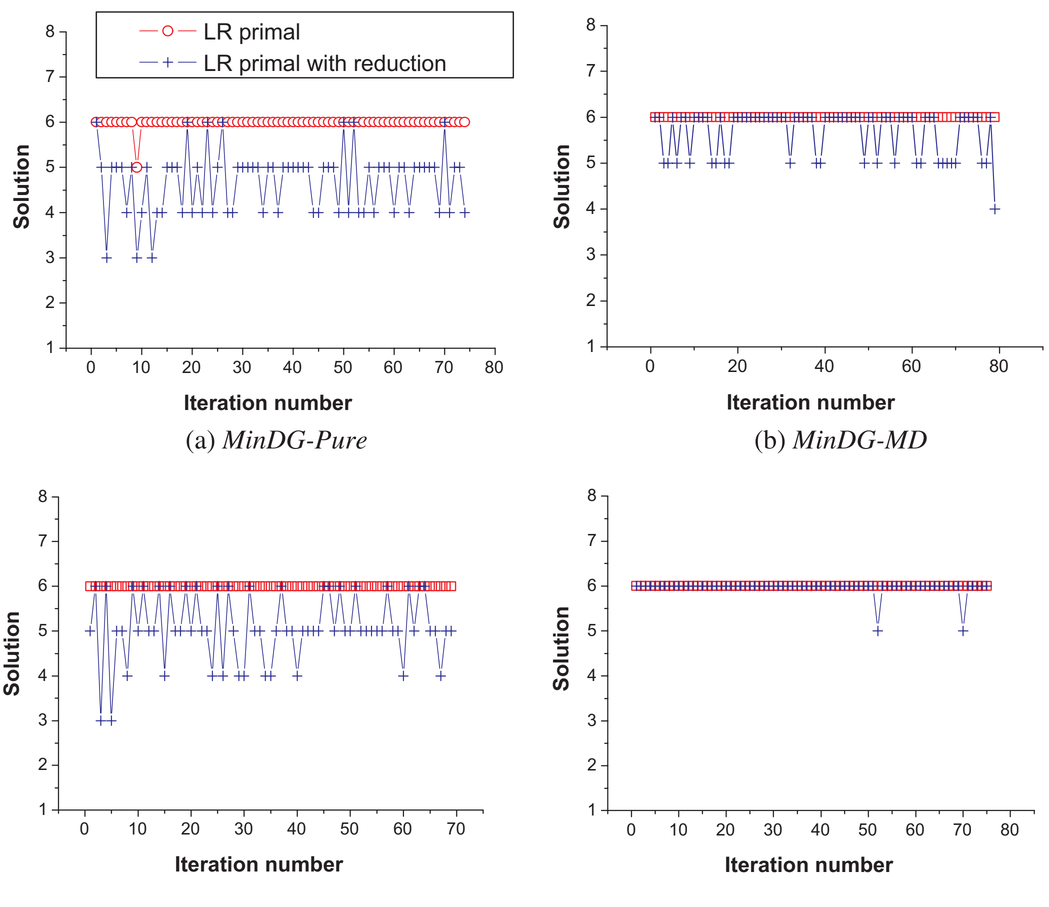
solution, reconstruction of a primal solution from a Lagrangian dual is much more straightforward than reconstructing from a LP relaxed solution. Secondly, iterations of Lagrangian Relaxation algorithm will, in general, be quicker than attempting to solve the LP relaxation [12].

4.2. Heuristic for DG reduction

At each iteration of the Lagrangian dual algorithm above presented, the solution of subproblem $\mathbf{y}^k$ is used to compute feasible $\mathbf{x}^k$ that satisfies $\mathbf{y}^k \leqslant \mathbf{x}^k$. Thus it is highly probable that each element of $\mathbf{x}^k$ is set to 1, and this brings too many DG candidates to be selected as DG. To mitigate this problem, we propose a heuristic for DG reduction and adapt the heuristic to the Lagrangian dual algorithm. The heuristic algorithm (Table 2) is embedded in Step 2 right after the feasible $\mathbf{x}^k$ is computed.

5. Evaluation results

In this section, we show the simulation results of the Lagrangian dual algorithm and our reduction heuistic. Our simulation system has Xeon 3 GHz dual CPU and 2 GB main memory. We simulate our proposal on networks of 40, 80, 120 and 250 nodes that are generated by ToGend [14]. We use GLPK [13] library to implement



(a) *MinDG-Pure* (b) *MinDG-MD* (c) *MinDG-CL* (d) *MinDG-MD& CL*

Fig. 6. In case network size of 40, the obtained solution at each iteration.

the Lagrangian Relaxation based dual algorithm. Unfortunately, in our system, our GLPK library cannot obtain the result of Lagrangian Relaxation on network size of more than 250 nodes due to memory allocation failure. Notable simulation parameters are shown in Table 3.

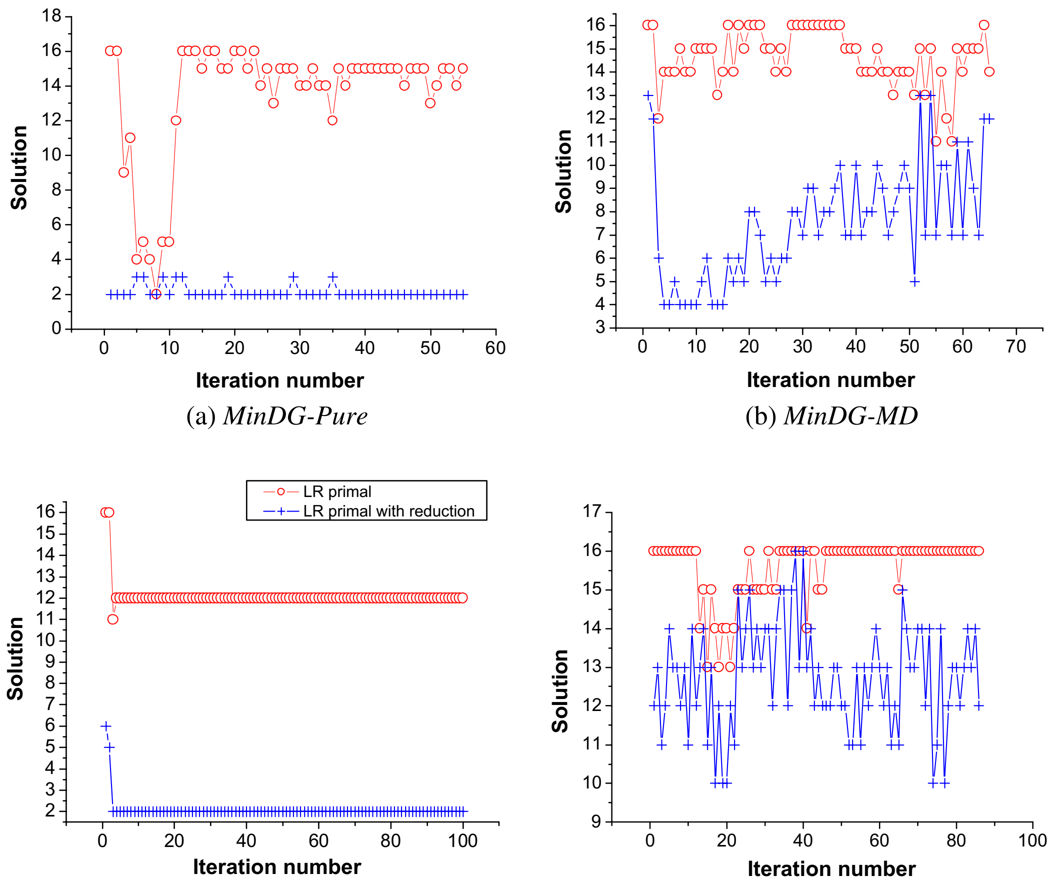
Ideally we would like to observe how different our algorithm is from the integer optimal solution. However it took about 667 s our GLPK to compute the optimal solution of the *MinDG-Pure* problem on just 40 nodes due to exponential time complexity of the problem. For the *MinDG-MD* and *MinDG-CL* problem, we could not obtain the optimal solution of 40 nodes until 24 h have elapsed. Thus our algorithm is compared with the lower bound by LP relaxation.

We measure and compare the minimum number of DG achieved by the Lagrangian Relaxation alone and the one achieved by the Lagrangian Relaxation with reduction heuristic.

Table 4 presents the measured result of each problem. The rows that are named as 'LR primal' show the results achieved by the Lagrangian Relaxation alone. The column that is named as '*MinDG-MD & CL*' shows the results subject to both the mandatory diversity and the capacity limit condition. Since the reduction heuristic is inserted into the Lagrangian Relaxation algorithm, we can achieve both results – the result achieved by the Lagrangian Relaxation alone and the result achieved by the Lagrangian Relaxation with reduction heuristic – in one execution.

As shown in Table 4, we observe that the Lagrangian Relaxation with reduction heuristic outperforms the use of the Lagrangian Relaxation alone. The maximum integrality gap is 20 in case of the *MinDG-MD & CL* problem on network size of 250 nodes. In case of network size of 40 nodes, the best solution of *MinDG-Pure* problem is integer optimal.

It spends 2833 s to finish the entire iterations of the Lagrangian Relaxation with reduction heuristic procedure in case of the *MinDG-MD & CL* problem on network size of 250 nodes. However, in case of network size of 40 nodes, the best solutions of the *MinDG-Pure* and *MinDG-CL* problem are obtained at iteration 3 and it spends just about 1 s to finish iteration 3. In case of network size of 80 nodes, the best solutions of all problems are obtained earlier than iteration 3 and it spends about 2 s. In case of network size of 120, the best solutions of the *MinDG-Pure* problem and *MinDG-MD* problem are achieved earlier than iteration 3 as well and it spends about 4 s. In case of network size 250, the best solutions of the *MinDG-Pure, MinDG-CL*,



(a) *MinDG-Pure* (b) *MinDG-MD*

(c) *MinDG-CL* (d) *MinDg-MD& CL*

Fig. 7. In case network size of 80, the obtained solution at each iteration.

and *MinDG-MD* & *CL* are obtained at iteration 1, 8 and 17, respectively.

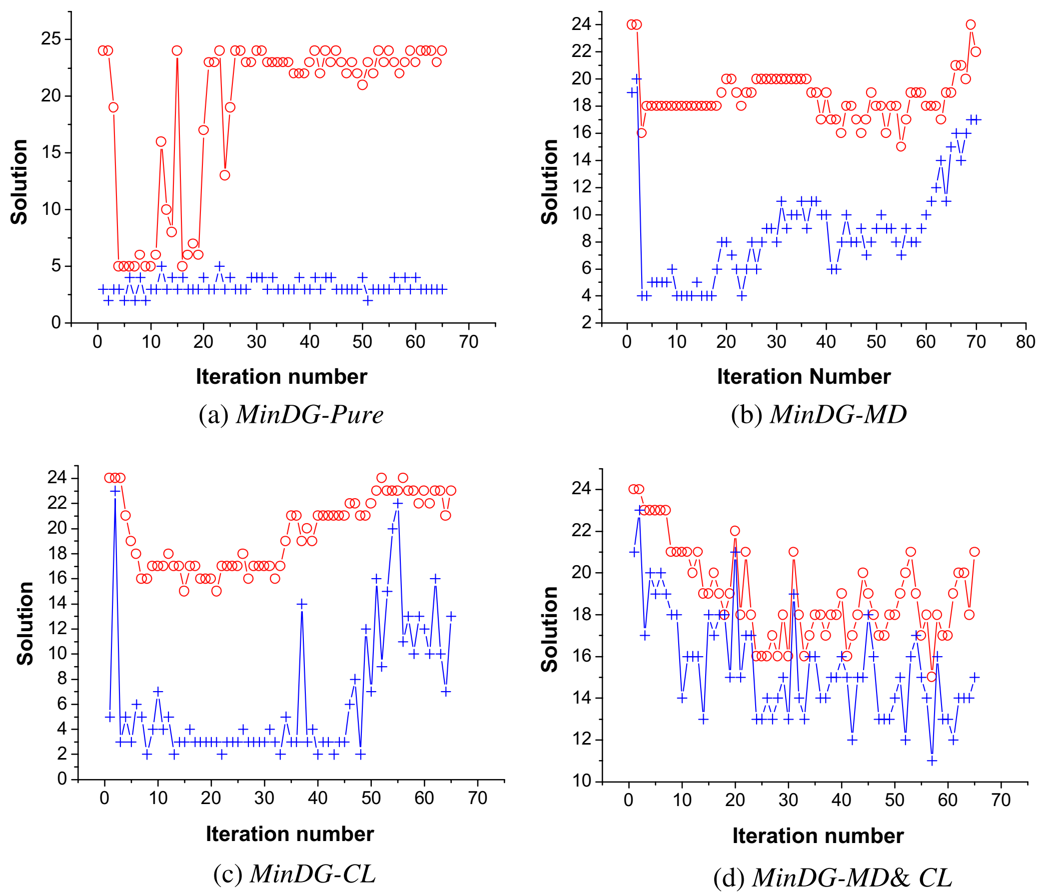
In case network size of 40 nodes, the best solutions of the *MinDG-MD* problem and *MinDG-MD* & *CL* problem are obtained at iteration 79 and iteration 52, respectively. In case network size of 120 nodes, the best solution of *MinDG-MD* & *CL* problem is obtained at iteration 57. In case network size of 250 nodes, the best solution of the *MinDG-MD* problem is obtained at iteration 38.

Figs. 6–9 plot the solution obtained at each iteration. As shown in Fig. 6(b and d), the solutions those are one bigger than the best solutions are obtained earlier than iteration 5. Thus, the solution approximated to the best solution can be obtained in a few seconds. Above all, our Lagrangian Relaxation with reduction heuristic needs much shorter period than solving the integer programming, and our methodology may still be useful for comparison and assessment purpose.

6. Conclusions

In this paper, we employed the DVS architecture in overlay streaming in order to achieve higher throughput, and to increase tolerance to packet loss due to network congestion, and proposed the Lagrangian Relaxation based dual algorithm and reduction heuristic in order to minimize the number of intermediate DVS sender in reasonable time. We also consider the feasible constraints of mandatory diversity and capacity limitation. The three types of minimum DG problem are individually proven to be NP-hard, reducing from the *k*-median problem, set multi cover problem and capacitated *k*-median problem. Our simulation results demonstrate the effectiveness of our Lagrangian Relaxation with reduction heuristic comparing with pure Lagrangian Relaxation.

This paper does not aim to an algorithm for the detection of dynamic join/leave and network fluctuations (i.e., bandwidth and connectivity). This paper proposes a fast algorithm for configuring an approximated optimal DVS gateway set and overlay topology under the given state of membership and network. The approximated optimal solution can be used to assess and evaluate the quality of on-line solutions. Of course, the fast and accurate detection and reflection of membership and network fluctuation is indispensable. Thus the *Distributed SNMP* [24,25] can be co-deployed with our approach, or entire network may be partitioned into smaller zones and our approach can be applied to the smaller zones. The zones can be configured dynamically and distributively. This co-



(a) *MinDG-Pure* (b) *MinDG-MD* (c) *MinDG-CL* (d) *MinDG-MD& CL*

Fig. 8. In case network size of 120, the obtained solution at each iteration.

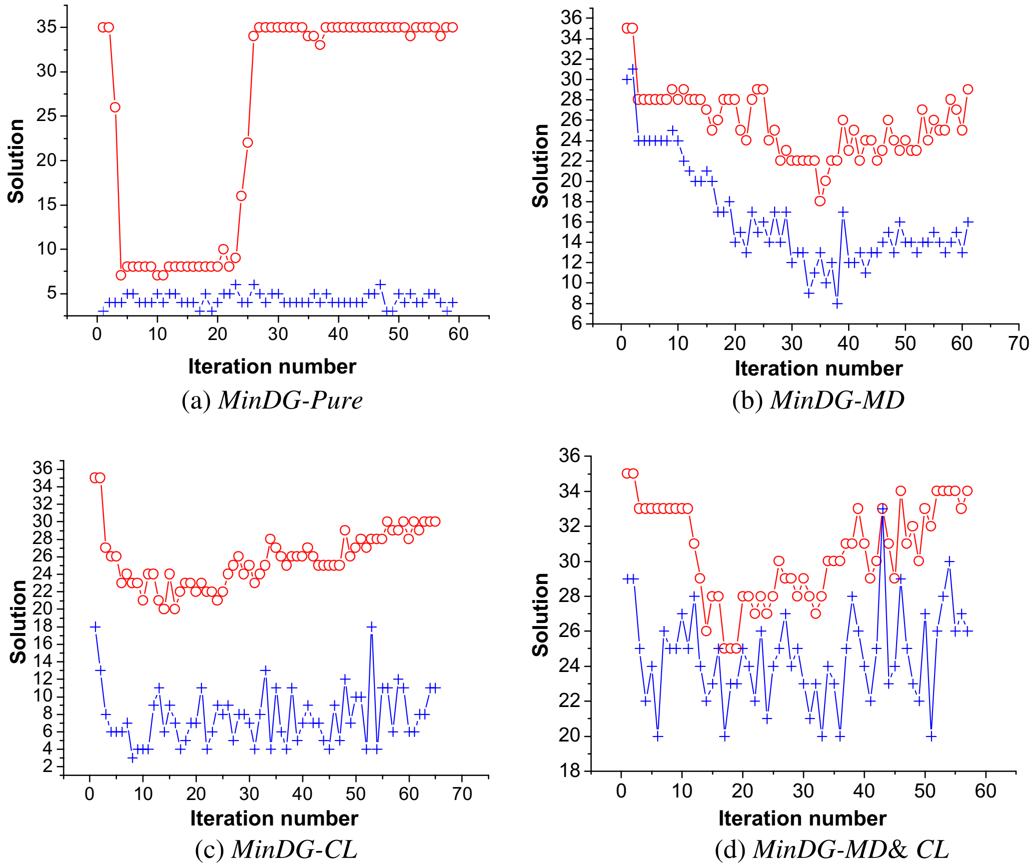


Fig. 9. In case network size of 250, the obtained solution at each iteration.

deployment is the one of main challenges in our future work.

Future work also involves optimizing the source-based distributed streaming methods and investigating the *fair networks* [10]. The concept of fair networks is more practical to real network situation since it considers all subscribers cannot be met their requirements. We also plan to adapt an intuitional heuristics directly to the minimum DG problems in order to overcome the memory limitations of GLPK and shorten the computation time.

Acknowledgements

This work was carried out during the tenure of an ERCIM ‘‘Alain Bensoussan’’ Fellowship Programme.

References

[1] T. Nguyen, A. Zakhor, Multiple sender distributed video streaming, IEEE Trans. Multimedia 6 (2004) 315–326.
[2] T. Nguyen, A. Zakhor, Distributed video streaming, in: Proc. SPIE, vol. 4673, 2002, pp. 186–195.
[3] A. Reibman, H. Jafarkhani, Y. Wang, M. Orchard, R. Puri, Multiple description coding for video using motion compensated prediction, in: Proc. ICIP, vol. 3, 1999, pp. 837–841.
[4] J. Apostolopoulos, Reliable video communication over lossy packet networks using multiple state encoding and path diversity, in: Proc. SPIE, vol. 4310, 2001, pp. 392–409.
[5] R. Puri, K. Ramchandran, K. Lee, V. Bharghavan, Forward error correction (fec) codes based multiple description coding for internet video streaming and multicast, Signal Process.: Image Commun. 6 (2001) 745–762.
[6] K. Goyal, J. Kovacevic, Generalized multiple description coding with correlating transforms, IEEE Trans. Image Process. 10 (2001) 351–366.
[7] V. Padmanbhan, H. Wang, P. Chou, Resilient peer-to-peer streaming, Technical report MSR-TR-2003-11 (2003).
[8] V. Vazirani, Approximated Algorithms, Springer, New York, 2001.
[9] J. Chuzhoy, Y. Rabani, Approximating k-median with non-uniform capacities, in: Proceedings of the ACM-SIAM Symposium on Discrete Algorithms, 2005, pp. 952–958.
[10] M. Pioro, D. Medhi, Routing, Flow, and Capacity Design in Communication and Computer Networks, Morgan Kaufmann, Los Altos, CA, 2004.
[11] A. Geoffrion, Lagrangian relaxation for integer programming, Math. Program. Study 2 (1974) 82–114.
[12] M.G. Rutten, S. Maskell, M. Briers, N.J. Gordon, Multipath track association for over-the-horizon radar using Lagrangian relaxation, in: Proc. SPIE, vol. 5428, 2004 452–463.
[13] GNU linear programming kit. Available from: www.gnu.org/software/glpk/glpk.html.
[14] A Notre-Dame based topology generator. Available from: http://www.eng.tau.ac.il/~osnaty/togend.html.
[15] A. Ganja, H. Zhang, Internet multicast video delivery, in: Proc. IEEE, vol. 93, 2005, 159–170.

[16] M. Castro, P. Drushel, A. Kermarrec, A. Nandi, A. Rowstron, A. Singh, SplitStream: high-bandwidth content distribution in a cooperative environment, in: Proc. IPTPS, 2003.
[17] V. Padmanabhan, H. Wang, P. Chou, K. Sripanidkulchai, Distributing streaming media content using cooperative networking, Technical report MSR-TR-2002-37, 2002.
[18] S. Savage, A. Collins, E. Hoffmanm J. Snell, T. Anderson, The end-to-end effects of internet path selection, in: Proc. ACM SIGCOMM, 1999, 200–289.
[19] D.G. Andersen, Resilient overlay networks, S.M. Thesis, Massachusetts Inst. Technol., Cambridge, 2001.
[20] A. Riabov, Z. Liu, L. Zhang, Overlay multicast trees of minimal delay, in: Proc. ICDCS, 2004.
[21] X. Zhang, J. Liu, B. Li, T. Yum, CoolStreaming/Donet: a data-driven overlay network for peer-to-peer live media streaming, in: Proc. INFOCOM, 2005.
[22] M. Zhang, L. Zhao, Y. Tang, J. Luo, S. Yang, Large-scale live media streaming over peer-to-peer networks through global Internet, in: Proc. P2PMMS, 2005.
[23] X. Liao, H. Jin, Y. Liu, L.M. Ni, D. Deng, AnySee: peer-to-peer live streaming, in: Proc. INFOCOM, 2006.
[24] L. Li, M. Thottan, B. Yao, S. Paul, Distributed network monitoring with bounded link utilization in IP networks, in: Proc. INFOCOM, 2003.
[25] U. Hoffmann, I. Milouchewa, Distributed measurement and monitoring in IP networks, in: Proc. ISAS, 2001.

Sang-Seon Byun received the B.S and M.S. degree in Computer Science from Korea University, Seoul, Korea, in 1996 and 2002, and Ph.D. in computer science in Korea University in 2007. He was a research professor in Graduate School of Embedded Software, Korea University. He is currently a research fellow in Signal processing group in NTNU, Norway. His current interests are in optimization of P2P network and on demand streaming in P2P.

Chuck Yoo received the B.S. and M.S. degree in Electronic Engineering from Seoul National University, Seoul, Korea and the M.S. and Ph.D. in computer science in University of Michigan. He worked as a researcher in Sun Microsystems Laboratory from 1990 to 1995. He is now an professor in Department of Computer Science and Engineering, Korea University, Seoul, Korea. His research interests include high performance network, multimedia streaming, and operating systems. He served as a member of the organizing committee for NOSSDAV 2001.