Dynamic Resource Allocation and Scheduling for Cloud-Based Virtual Content Delivery Networks

Tai-Won Um, Hyunwoo Lee, Won Ryu, and Jun Kyun Choi

This paper proposes a novel framework for virtual content delivery networks (CDNs) based on cloud computing. The proposed framework aims to provide multimedia content delivery services customized for content providers by sharing virtual machines (VMs) in the Infrastructure-as-a-Service cloud, while fulfilling the service level agreement. Furthermore, it supports elastic virtual CDN services, which enables the capabilities of VMs to be scaled to encompass the dynamically changing resource demand of the aggregated virtual CDN services. For this, we provide the system architecture and relevant operations for the virtual CDNs and evaluate the performance based on a simulation.

Keywords: Virtual content delivery network, cloud-based content delivery, multimedia service distribution.

I. Introduction

Today, video streaming occupies more than half of Internet traffic [1]. Rapidly growing video traffic offers challenges to video content providers as well as internet service providers. A content delivery network (CDN) is a prevalently used overlay network built over existing physical networks, enabling video content to be distributed to the network edge, closer to end users. It can provide reliable and scalable video services, while reducing network congestion and service response time [2], [3]. With these advantages, more and more video content providers are adopting CDNs for their video service delivery [4].

However, existing CDN solutions cannot economically cope with dynamic video traffic characteristics because CDN providers commonly construct their own service systems (for example, storage and streaming servers), which are able to accommodate peak demands of the video service requests. The system deployment to cover peak demands consequently results in low efficiency and high cost of video services for all other periods [5], [6].

Many multimedia services, such as live streaming and video on demand (VoD) require not only broadband bandwidth but also reliability with a certain level of quality. In addition, new multimedia services, such as interactive multimedia, personal TV, cloud-based gaming services, and so on, are resulting in increasingly complicated networking and computing capability requirements necessary for the provision of content delivery [7]-[9]. Moreover, as service types to be provided over a CDN are quite divergent, the system management and operation of a CDN are more complicated. CDN providers are crying out for a pervasive and generalized way to accommodate the variety of multimedia services, including legacy media services and emerging new media services [10].

Manuscript received Sept. 22, 2013; revised Jan. 21, 2014; accepted Feb. 3, 2014.

This research was supported by the MSIP (Ministry of Science, ICT & Future Planning), Korea, under the R&D program supervised by the KCA (Korea Communications Agency) (KCA-2013-12-912-03-001).

Tai-Won Um (phone: + 82 42 860 5205, twum@etri.re.kr), Hyunwoo Lee (hwlee@etri.re.kr), and Won Ryu (wlyu@etri.re.kr) are with the Broadcasting & Telecommunications Media Research Laboratory, ETRI, Daejeon, Rep. of Korea.

Jun Kyun Choi (jkchoi@ee.kaist.ac.kr) is with the Department of Electrical Engineering, KAIST, Daejeon, Rep. of Korea.

A cloud is a dynamic pool of virtualized computing resources and offers an elastic scheme matching the user demand, so that allocated resources can be scaled up or down on a pay-per-use basis [11]. In an Infrastructure-as-a-Service (IaaS) cloud, users can launch their virtual machines (VMs) with required computing resources, and an IaaS cloud is able to control and maintain several VMs on the same physical server with remarkable flexibility [12]-[14]. Furthermore, by interconnecting the IaaS clouds, computing resources distributed over geographically dispersed physical nodes can cooperate with each other, which facilitates the operation of such distributed media services as a CDN [10].

Adopting and utilizing this cloud infrastructure, CDN providers can execute their services cost-effectively and efficiently by renting the cloud resources. However, there are some challenges in this cloud-based CDN [15]. Many multimedia services are provided through the same CDN system simultaneously, and it is expected that each multimedia application and content request (from users) made to the corresponding multimedia application will demand different QoS requirements when making use of the service. To encompass the needs of service users, while producing sufficient profit, a CDN provider can offer different grades of service, which are typically described in a service-level agreement (SLA), and the service cost is determined by the SLA chosen by the user [13], [14], [16].

From the aspect of a CDN provider, the SLA and service policy must be taken into account when allocating limited resources to various multimedia applications on demand [14]. Consequently, allocating and providing various types of cloud resources appropriately to multimedia applications becomes one of the key issues in a cloud-based CDN because over- or under-provisioning will directly affect the service expenditure as well as the QoS [12].

In this paper, we propose a novel cloud-based virtual CDN architecture, which aims to provide cost-efficient and elastic CDN services by multiplexing a number of video service applications with different SLAs into a VM and by allowing the VM to be scaled according to the aggregated traffic demand from the multimedia services. To minimize the service costs while satisfying the SLAs with service users, we also investigate a dynamic resource scheduling algorithm for the cloud-based CDN.

The remainder of this paper is organized as follows. We briefly review the related work in section II. Section III describes the proposed architecture for a cloud-based virtual CDN. The operational procedures of the virtual CDN are provided in section IV. Section V describes the proposed dynamic resource scheduling algorithm, and its performance is analyzed in section VI. Finally, section VII gives some concluding remarks.

II. Related Work

In this section, we review the existing works directly related to the proposed cloud-based CDN framework in terms of SLAs with cloud users, resource scheduling models, and content delivery based on cloud computing.

1. SLA and Resource Scheduling Based on Cloud Computing

Service providers should efficiently utilize their cloud resources to minimize cost for the service provisioning and also satisfy QoS requirements, which means that resource allocation and scheduling to fulfill SLAs with users are quite critical [15], [17]. In this paper, "scheduling" refers to the way applications are allocated to a VM as well as the way VMs are allocated to the available computing resources.

In cloud computing, there has been much research to provide resource allocation and scheduling based on the SLA between the cloud provider and the user [13]. However, resource scheduling that meets the SLA is still a big issue in cloud computing because there exist many different service environments and requirements.

There have been a number of resource allocation and scheduling methods to control VM resources in physical servers to guarantee that SLAs are kept; see [11], [13], [15], [17], and [18] for examples. Recent research mainly dealt with resource scheduling based on the cloud environment with budget or resource constraints [12], [19]-[22]. Specifically, a bandwidth allocation scheme for data center networks was proposed by Li and others [18] to guarantee QoS and achieve high bandwidth utilization. This scheme uses a centralized management unit to monitor data centers and to differentially allocate bandwidth according to the SLA of the application. Villegas and others [12] investigated various provisioning and allocation policies in IaaS cloud and analyzed the performance.

In our proposed scheduling scheme, a number of virtual CDN services, which may have different SLA contracts, are collocated in a VM and content requests from users are treated with different priorities according to their virtual CDN services. Moreover, among the various types of resources in IaaS cloud, we mainly focus on the resource scheduling for cloud bandwidth, because bandwidth is recognized as a significant factor affecting the performance of such bandwidth-intensive services as a CDN [18].

2. Cloud CDN

The basic framework to dynamically construct a virtual CDN based on cloud computing was proposed in [2], [23]. Moreira and others [2] proposed a programmatic adaptation modification in the CDN infrastructure, wherein replica servers

ETRI Journal, Volume 36, Number 2, April 2014
http://dx.doi.org/10.4218/etrij.14.2113.0085

used to store copies of original content are virtually created and modified on the cloud infrastructure. Closest to our work is Jin and others' [23] proposed content-delivery-as-a-service (CoDaaS), which enables on-demand virtual content delivery service overlays for user-generated content (UGC). In CoDaaS, when a CDN provider receives a content delivery request from a UGC provider, it configures a virtual content delivery overlay, based on the cloud after calculating its content distribution tree.

Recently, some progress [5], [6], [24] has been made in an aspect of resource allocation for cloud-based video services. Niu and others [24] proposed a predictive resource auto-scaling system that reserves the required bandwidth resources for the VoD provider to match its short-term resource demand. Aggarwal and others [6] proposed a virtualized cloud infrastructure with a time-shifting function to better utilize deployed resources and to lower a provider's costs for real-time IPTV services. He and others [5] described a theoretic model to investigate the trade-off between the cloud cost and the achieved user quality of experience for cloud-based video streaming and demand dynamics.

In general, when a VM is created and maintained in an IaaS cloud, it places considerable burden upon the hypervisor and management works supplied by the cloud provider. So, it is non-realistic if a VM is occupied by only one CDN service in consideration of the capability of a VM that is commercially used by Amazon, Google, and so on. Our work differs from previous research on cloud CDNs in that we consider a number of CDN services, which run on a VM and compete for VM resources, whereas previous works mainly dealt with a single CDN scheduling in a physical machine.

III. Proposed Cloud-Based Virtual CDN Architecture

Figure 1 shows a simplified architecture of an on cloud-based virtual CDN. The control plane shown in Fig. 1 features a plurality of service management systems (SMSs) and transport control systems (TCSs), as well as a cloud broker (CB). The control plane configures a virtual CDN in response to a content provider's request and can be operable for the content provider to transmit and distribute video content to service users through the configured virtual CDN.

The data plane consists of a number of distributed cloud data centers, which have at least one VM for use in the provisioning of a media service. The VM, which includes resources for media processing, content storage, and networking, is used to configure a virtual CDN.

1. SMS

SMSs receive a virtual CDN configuration request, including service requirements (for example, service level, content property, geographic coverage, virtual CDN beginning/end time, and so on) from a video content provider, and request a TCS to create a new virtual CDN that meets its service requirement. Once a virtual CDN is successfully created and assigned by the TCS for the content provider, the SMS provides a management environment for the virtual CDN to enable the content provider to manage and monitor its virtual CDN. Furthermore, the SMS is responsible for informing the closest virtual CDN node where the user can access and download video content upon receiving a content request from the virtual CDN user.

2. TCS

A TCS dynamically configures a virtual CDN using the cloud resource registered in the CB upon receiving the virtual CDN configuration request from the SMS. By analyzing the service requirements, the TCS constructs an optimized topology, which may be a hierarchical tree structure connecting data center, to efficiently and effectively distribute video content from a content source to a lot of user devices. In the event of a deficiency or surplus of virtual CDN resources, the TCS can adjust the allocated capacity of VMs or migrate VMs to other data centers which are capable of accommodating user requests. The example illustrated in Fig. 1 shows two TCSs, each of which can be connected with one or more SMSs and the CB through a control interface.

3. CB

For virtual CDN services, the CB rents a set of VM instances from one or more cloud providers. Upon receiving a request from the CB, the cloud provider allocates a VM that is capable of storing and distributing video content via the virtual CDN to a large number of user devices. Each virtual CDN configuration request can be accommodated through the provisioned VM instances in the distributed data centers. To manage and control the VMs, the CB includes a control/management function and protocols for supporting a variety of cloud data centers. It also monitors the normal operation and utilization of the VMs. After allocating the registered VM stores in response to the request of the TCS, the CB records in an internal database, how the resources of each VM are scheduled or allocated to the virtual CDN.

4. Cloud

Cloud computing maintains virtualized computing resources to aggregate and share a large number of distributed physical resources and offers an elastic scheme that matches user demand, so that the allocated resources including processor,

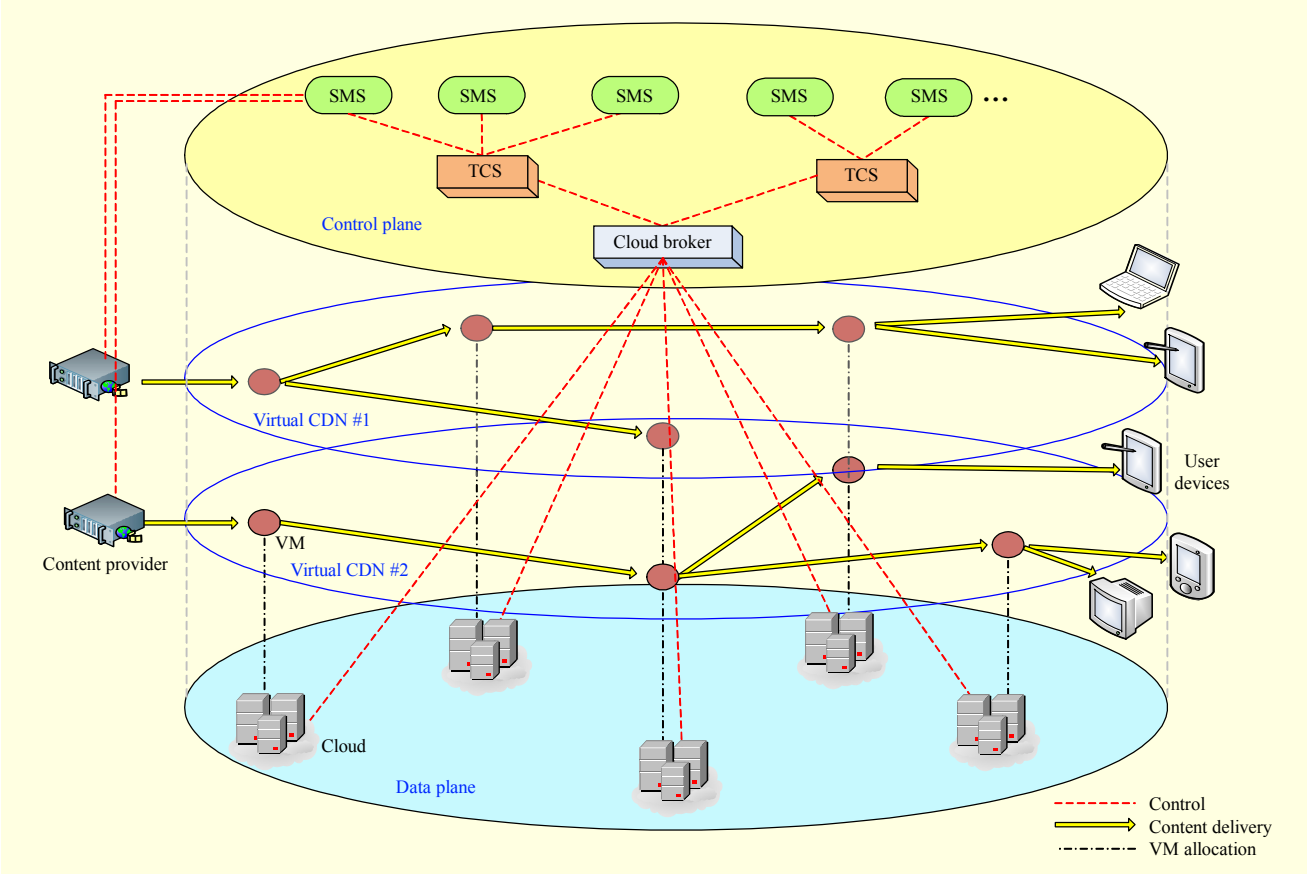

Fig. 1. Cloud-based virtual CDN architecture.

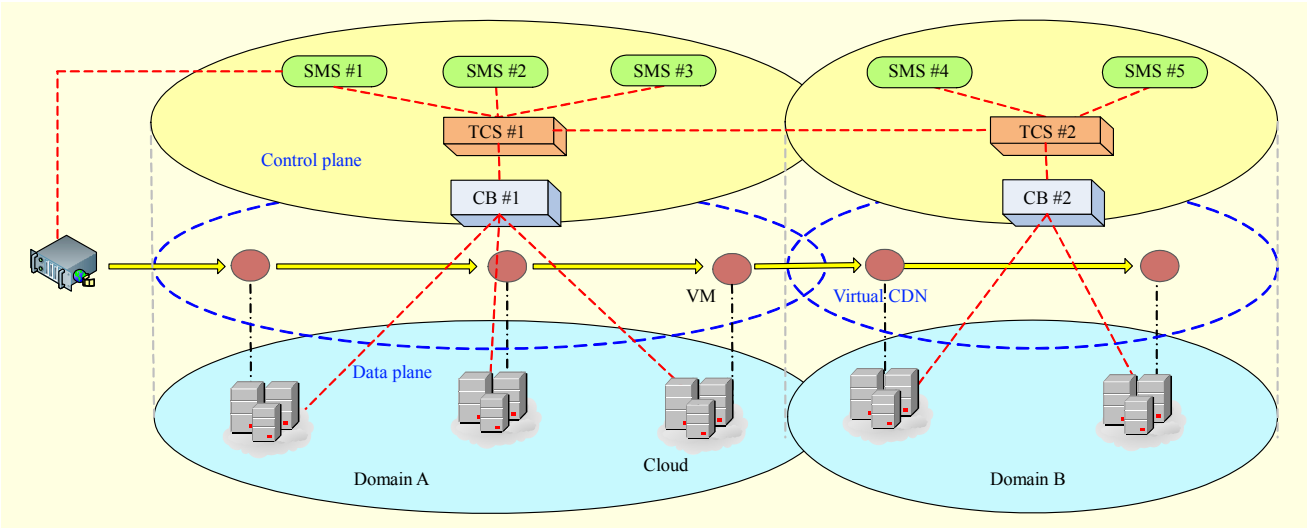

Fig. 2. Example of configuring virtual CDNs based on distributed control architecture.

memory and bandwidth can be scaled up or down with a pay-per-use base. CB can launch VM with required computing and networking resources, and IaaS cloud is able to control and maintain several VMs on the same physical server with remarkable flexibility.

In the proposed virtual CDN architecture, replica services (caches storing copies of original server content), are established in VMs instantiated in cloud data centers. New replica servers can be created and added to the virtual CDN infrastructure according to user demands. Furthermore, the replica server is able to be repositioned, adapting the infrastructure to user demands and resource usage states.

Figure 2 shows a diagram illustrating an example of an architecture in which the TCSs coordinate with a CB in multiple domains and in a distributed manner. Each TCS may be a distributed one that operates in connection with a CB. Distributed TCSs obtain cloud resource information from the CBs. After obtaining resource information from the CB, a distributed TCS performs a simplification and abstraction of the information and the topology, in the domain and thereafter floods the distributed TCS with its own domain information resulting from the simplification and abstraction.

Performing the simplification and abstraction of the resource information and topology of the domain according to the policy, may vary based on the purpose. For example, let us assume that domains A and B are provided by different cloud and network providers, and virtual CDN is dynamically created between distributed TCS #1 and TCS #2. To exchange information for use, TCS #1 may provide TCS #2 with some specifications, including the processor, memory, and bandwidth of the cloud data centers that belong to domain #A and that are connected to an external network, such as domain #B, along with other network connection information. Accordingly, the TCS #1 may prevent detailed information of the cloud data centers and network topology of domain A from being disclosed to all service providers in domain B.

The distributed TCSs may regularly update the resource and topology information of each domain, and it will flood the domain information to other TCSs. Each of the TCSs collects the information, thereby enabling the overall virtual CDN configuration to be identified.

In response to a dynamic virtual CDN configuration request from SMS #1, TCS #1 analyzes the received request and then attempts to dynamically configure a virtual CDN by requesting CB #1 in the same domain. In addition, when necessary, TCS #1 transmits a virtual CDN configuration request to TCS #2 so as to configure a virtual CDN that traverses multiple domains.

IV. Virtual CDN Operational Procedures

Figure 3 shows a diagram illustrating an example of the procedures used to create a virtual CDN and transmit content.

Firstly, the CB and cloud data centers perform resource registration procedures between them. A content provider requests an SMS for a scheduled virtual CDN registration to render a virtual CDN service (for example, VoD, live streaming, and so on) at a pre-set time. An SMS then transmits a virtual CDN configuration request to a TCS, to trigger a new virtual CDN creation. As previously mentioned in clause 3.1, the virtual CDN configuration request includes the service requirements needed to create a virtual CDN customized to the content provider.

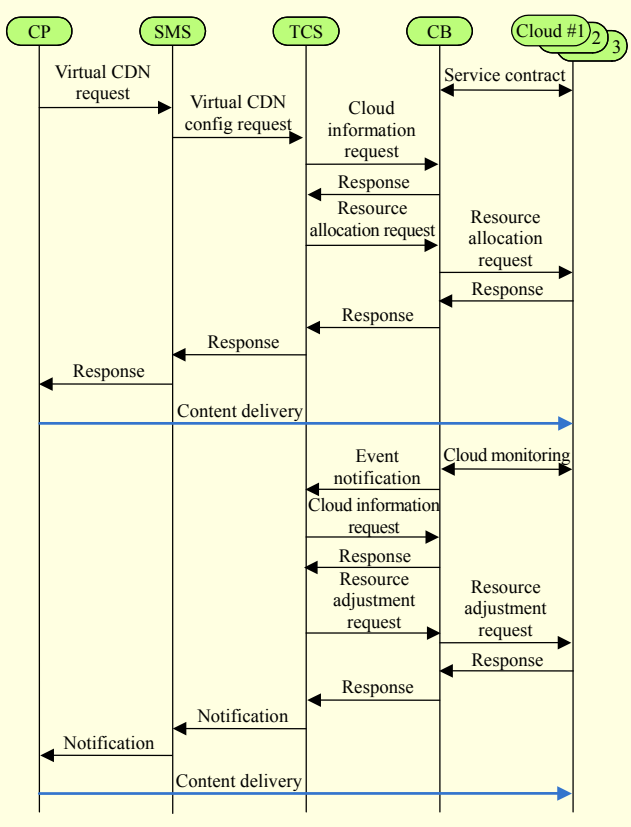

Fig. 3. Virtual CDN creation and content delivery procedure.

Upon receiving the virtual CDN configuration request, the TCS issues a resource information request to a CB and in turn, receives a resource information response, thereby identifying the cloud resources that are necessary for configuring the virtual CDN. After checking the available resources, the TCS sends a resource allocation request to the CB for cloud resources that are required for the virtual CDN configuration.

The CB may accept the resource allocation request once the presence of available resources is confirmed. In addition, around the time when the resources start to be used, the CB requests corresponding VMs in data centers to allocate resources.

In response to a resource allocation request being successful through the CB, the TCS transmits a virtual CDN configuration response message to the SMS to notify a successful allocation of the cloud resources that constitute a virtual CDN. After receiving the virtual CDN configuration response message, the SMS notifies the content provider that the scheduled virtual CDN registration has been successful. The content provider transmits and distributes content, data, or the like through the virtual CDN generated at a pre-set time, that is, a virtual overlay network established among data centers that provides the cloud resources.

By periodically sending status and statistic information

request message, the CB collects status and statistic information from VMs. Through these operations, the CB monitors the usage state of each VM regarding the processor, memory, and bandwidth and detects the presence of any errors or malfunctions within the cloud systems.

If it is determined that a VM is turned off or has errors, the CB may identify the virtual CDNs transmitted through the cloud node and send an event notification message to the TCS that manages the virtual CDN. In response to the event notification message, the TCS requests the CB to modify allocated cloud resources or to migrate VMs to another data center. In response to the request, the CB requests the cloud, to which the resources have been previously allocated, to change the resources and receives responses to the request from the cloud. Alternatively, the CB may request a new VM, which has never been used, to allocate virtual CDNs.

When the change in resource adjustment is successful, the CB transmits a response to the TCS to notify the success, and the TCS transmits a virtual CDN adjustment notification message to the SMS. Through these operations, content can be transmitted and distributed from the content source to user devices over the reconfigured virtual CDN.

At a virtual CDN service termination time, the TCS transmits a resource release request to the CB, and in turn, the CB requests each cloud provider to release the allocated VMs and receives a resource release response, which notifies the release of the allocated VMs.

V. Dynamic Resource Scheduling for Virtual CDN

For a variety of reasons, such as QoS, content type, scalability, security, and so on, content providers are forced to use various types of virtual CDNs, and subsequently a virtual CDN provider is required to build a customized virtual CDN to suit the needs of a content provider. However, there are some practical restrictions to building virtual CDNs that satisfy service requirements based on a cloud infrastructure.

First, commercial cloud providers normally offer restricted service contracts, in terms of contract duration and change of allocated resources. Such a restricted contract model makes it challenging for CDN providers to determine required VM capability while satisfying the SLAs of virtual CDN users.

Second, in providing the virtual CDN services, there is a trade-off between the service objectives of the virtual CDN provider and its content provider; the former needs to minimize the rental cost of VM instances, while the latter should maximize the QoS to increase the service satisfaction of end users. The simplest way to maximize QoS is the over-provisioning of VM, which consequently increases the rental cost of the VM.

```
Algorithm Dynamic resource allocation and scheduling algorithm
  Input: Available host list and running VM list
  Input: Running virtual CDNs with Gold, Silver, or Bronze class
  Input: User content requests
  globalvariable requestRateVariation, avResponseDelay;
1:    // Dynamic resource allocation algorithm
2:    if requestRateVariation > upperThreshold then
3:        if avResponseDelay > delayThreshold then
4:            begin vmIncreaseProcess(avResponseDelay);
5:        else
6:            begin vmIncreaseProcess;
7:        end if
8:    else if requestRateVariation < lowerThreshold then
9:    begin vmDecreaseProcess;
10:   end if
11:   // Non-preemptive priority scheduling algorithm
12:   for all Queue, High to Low do
13:       if queueSize != 0 then
14:           Dequeue a content request;
15:           lookup corresponding content;
16:           schedule the content;
17:           update statistics and repeat;
18:       end if
19:   end for
```

The proposed scheduling algorithm for virtual CDN makes use of three service classes (Gold, Silver, and Bronze) in consideration of the fact that a large number of service classes would produce significant management overhead.

Periodically, the virtual CDN node examines the variation of arrival rate. If the current arrival rate exceeds the previous arrival rate by the pre-defined upper threshold value, then it also verifies whether the service response delay of content requests, that belong to Gold class, is over the delay threshold or not. If so, the virtual CDN node informs the CB of this situation by adding the current statistics information of the virtual CDN. Otherwise a CB can execute the adjustment of the capacity of the VM by itself and request the cloud to increase the VM as much as a resource unit. Similarly, if the current arrival rate is decreased under the lower threshold, then the VM adjustment is also triggered to save the VM rental expense (See above algorithm, line 2 to 10).

According to the SLA of each virtual CDN, the content request which belongs to a virtual CDN is dealt with differentiated resource scheduling methods. When watching VoD content via the virtual CDN, service response delay, the time from the arrival of a content request until the streaming starts, is quite critical for the service quality of the experience. In this paper, we adapt priority-based scheduling, which is a relatively simple but effective method of supporting preferential service to reduce the service response delay (See above algorithm, line 12 to 19).

ETRI Journal, Volume 36, Number 2, April 2014
http://dx.doi.org/10.4218/etrij.14.2113.0085

We consider the virtual CDN that is running with threads pool and different queues. When a content request arrives at a virtual CDN node located at a VM, it is analyzed to find out to which virtual CDN and class it belongs.

The content requests are classified as Gold, Silver, or Bronze and then placed in descending order into different queues based on this classification. Content requests with a high priority are given precedence over those with lower priorities. After that, the content requests are scheduled from the head of a given queue only if all queues of higher priority are empty, and are processed according to the non-preemptive priority scheduling algorithm, where the video content transmission will not be interrupted until it is finished.

We assume that the required processor load and bandwidth of the content requests received by a VM, would be deterministic because streaming bandwidth and duration, and processing load for the content can be recognized once content is selected.

VI. Performance Analysis

We analyze the performance of the proposed algorithm using discrete event simulation. The simulation topology is shown in Fig. 4. virtual CDNs are created on cloud by using the SMS, TCS, and CB as described in sections 3 and 4. After establishing the CDNs, the content requests arrive at the VM where the virtual CDN node is located. Under this scenario, we are mainly interested in the resource allocation scheduling algorithm working in the VM.

There are three priority classes, each having a Poisson arrival pattern with mean arrival rate λ_i. Let λ be the total arrival rate, $\lambda = \lambda_{Gold} + \lambda_{Silver} + \lambda_{Bronze}$, where $\lambda_{Gold} = \lambda_{Silver} = \lambda_{Bronze}$, and λ increases from 0.01 to 0.30.

We also assume that the average streaming duration is 1,000 s which is exponentially distributed, and each of the three classes of content requests has the same streaming duration. Initial bandwidth given to a VM is set to 200 Mbps, and the bandwidth of each streaming is 10 Mbps, which is considered to be of high-definition resolution. In addition, we assume that bandwidth allocated to a VM can be scaled up or down at the request of a CB to the cloud service provider.

We analyze the average service response delay of each class excluding steaming time, and propagation delay and transmission time, which then gives us the utilization and normalized cost for content requests.

In this simulation, we define the adaptation function for resource adjustment as an exponential function with base 2. While the mean service response delay of the content requests belonging to Gold class exceeds the predefined threshold, the increase amount will be doubled at every checkpoint.

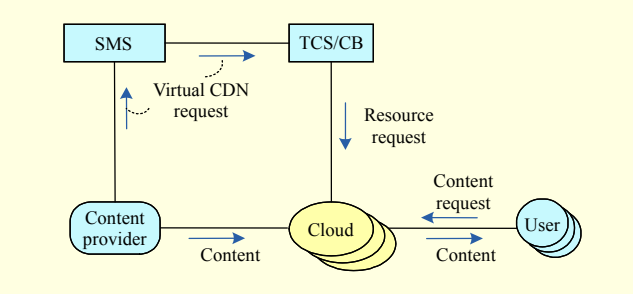

Fig. 4. Simulation topology with ten ingress nodes.

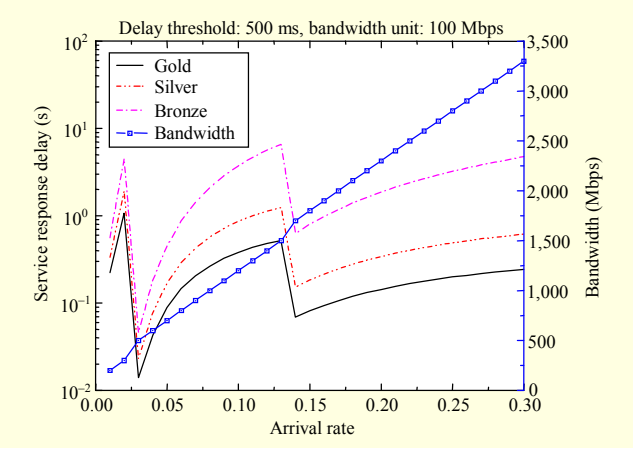

Fig. 5. Service response delay vs. arrival rate (case 1).

With the first case, we set the delay threshold of the Gold class at 500 ms and the bandwidth unit of the cloud at 100 Mbps. Figure 5 shows the effect of the proposed resource allocation and scheduling algorithm in terms of the service response delay. As the arrival rate of content requests increases, the service response delay of each class is increased. However, rapid increases in the arrival rate and the service response delay results in the allocation of much resources including bandwidth. As shown in Fig. 5, the increase in bandwidth resource is doubled at 0.03 and 0.14 on the horizontal axis, so that the service response to delay dramatically decrease in all cases. Overall, it restricts the service response delay of the content request that belongs to Gold class within the delay threshold. The plot also indicates that the variation width of the service response delay tends to decrease as the allocated bandwidth resource and the arrival rate increase. Intuitively, enough bandwidth and aggregated content requests may help to peak demand fluctuation, in an effect known as statistical multiplexing.

Secondly, we set the delay threshold of the Gold class at 200 ms and the bandwidth unit of the cloud at 50 Mbps, as shown in Fig. 6. The fine-grained resource control and tight management of the service response delay, cause rather significant variations in the service response delay. It may also

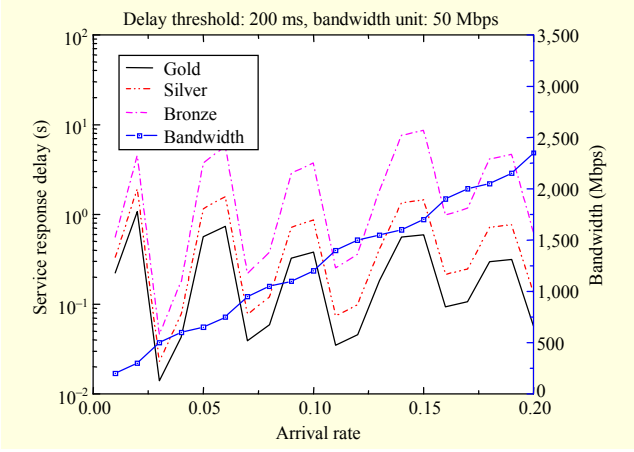

Fig. 6. Service response delay vs. arrival rate (case 2).

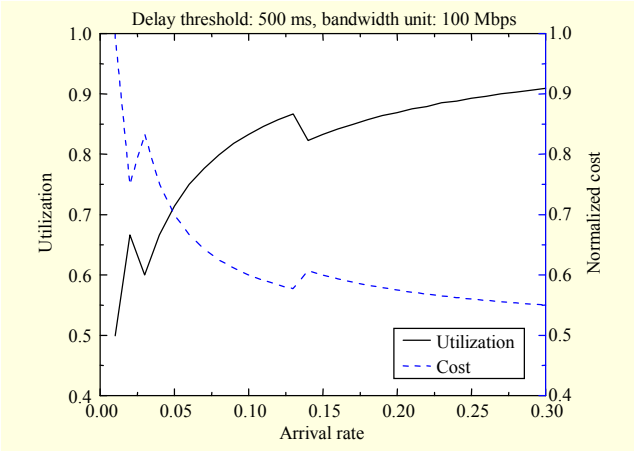

Fig. 7. Utilization and normalized cost (case 1).

be a burden to control systems such as CB, TCS and SMS as well as the cloud data center. Therefore, the determination of threshold values considering service conditions and restrictions would be quite critical to the provision of cloud-based CDN services.

Finally, we analyze the resource allocation and scheduling algorithm for the first case, with regards to cost. Figure 7 shows the utilization and normalized cost that is the ratio of content requests to bandwidth. As we can guess, increasing the number of content requests lowers the cost per content request because of the efficient use of the cloud resources.

VII. Conclusion

By taking into consideration the evolution and diversity of multimedia services, customized service environments provided by the virtual content delivery network (virtual CDN) will give many opportunities to content providers and users to access the virtual CDN from the point of view of service cost as well as differentiated service provision. To achieve it, the dynamic resource allocation, and scheduling, for the provisioning of virtual CDNs through cloud infrastructure are quite critical. However, there is a trade-off between QoS for virtual CDN services and the rental cost of cloud resources.

The proposed virtual CDN architecture realizes the dynamic and automatic setup of virtual CDNs based on distributed cloud infrastructure. Furthermore, it realizes cost-effective elastic virtual CDN services by using the dynamic resource scheduling algorithm while fulfilling the service level agreement. Without maintaining any CDN equipment, content providers can build a virtual CDN customizable to their multimedia service environment.

In future work, we plan to implement the proposed virtual CDN capabilities in commercial cloud computing and improve performance by applying more practical scheduling methods.

References

[1] "Cisco Visual Networking Index: Forecast and Methodology, 2012-2017," White paper, Cisco, May 29, 2013.

[2] A. Moreira et al., "A Case for Virtualization of Content Delivery Networks," *Proc. Winter Simulation Conf.*, Phoenix, AZ, USA, Dec. 11-14, 2011, pp. 3178-3189.

[3] S.P. Ponnusamy and E. Karthikeyan, "Cache Optimization on Hot-Point Proxy Caching Using Weighted-Rank Cache Replacement Policy," *ETRI J.*, vol. 35, no. 4, Aug. 2013, pp. 687-696.

[4] Z. Zhuang and C. Guo, "Building Cloud-Ready Video Transcoding System for Content Delivery Networks (CDNs)," *Proc. IEEE, GLOBECOM*, Anaheim, CA, USA, Dec. 3-7, 2012, pp. 2048-2053.

[5] J. He et al., "On the Cost-QoE Trade-off for Cloud-Based Video Streaming under Amazon EC2's Pricing Models," *IEEE Trans. Circuits Syst. Video Technol.*, vol. PP, no. 99, Sept. 2013.

[6] V. Aggarwal et al., "Optimizing Cloud Resources for Delivering IPTV Services through Virtualization," *IEEE Trans. Multimedia*, vol. 15, no. 4, June 2013, pp. 789-801.

[7] H.-C. Kim et al., "An MPEG-4 Compliant Interactive Multimedia Streaming Platform Using Overlay Networks," *ETRI J.*, vol. 28, no. 4, Aug. 2006, pp. 411-424.

[8] S.-K. Kim et al., "A Personal Videocasting System with Intelligent TV Browsing for a Practical Video Application Environment," *ETRI J.*, vol. 31, no. 1, Feb. 2009, pp. 10-20.

[9] K.I. Kim et al., "Cloud-Based Gaming Service Platform Supporting Multiple Devices," *ETRI J.*, vol. 35, no. 6, Dec. 2013, pp. 960-968.

[10] H. Wen et al., "Effective Load Balancing for Cloud-Based Multimedia System," *Int. Conf. EMEIT*, Harbin, Heilongjiang, China, vol. 1, Aug. 12-14, 2011, pp. 165-168.

[11] E. Pacini, C. Mateos and G.C. Garino, "SI-Based Scheduling of

Scientific Experiments on Clouds," *IEEE Int. Conf. IDAACS*, Berlin, Germany, vol. 02, Sept. 12-14, 2013, pp. 699-704.

[12] D. Villegas et al., "An Analysis of Provisioning and Allocation Policies for Infrastructure-as-a-Service Clouds," *IEEE/ACM Int. Symp. CCGrid*, Ottawa, ON, Canada, May 13-16, 2012, pp. 612-619.

[13] U. Schwiegelshohn and A. Tchernykh, "Online Scheduling for Cloud Computing and Different Service Levels," *IEEE IPDPSW*, Shanghai, China, May 21-25, 2012, pp. 1067-1074.

[14] X. Wang et al., "Design and Implementation of Adaptive Resource Co-allocation Approaches for Cloud Service Environments," *Proc. ICACTE*, Chengdu, China, vol. 2, Aug. 20-22, 2010, pp. 484-488.

[15] A.K. Das et al., "An Intelligent Approach for Virtual Machine and QoS Provisioning in Cloud Computing," *Proc. ICOIN*, Bangkok, Thailand, Jan. 28-30, 2013, pp. 462-467.

[16] T.N. Pham et al., "DILoS: A Dynamic Integrated Load Manager and Scheduler for Continuous Queries," *Proc. IEEE ICDEW*, Hannover, Germany, Apr. 11-16, 2011, pp. 10-15.

[17] V.C. Emeakaroha et al., "SLA-Aware Application Deployment and Resource Allocation in Clouds," *IEEE COMPSACW*, Munich, Germany, July 18-22, 2011, pp. 298-303.

[18] Y. Li et al., "Application Utility-Based Bandwidth Allocation Scheme for Data Center Networks," *Int. Conf. PDCAT*, Beijing, China, Dec. 14-16, 2012, pp. 268-273.

[19] Q. Zhu, and G. Agrawal, "Resource Provisioning with Budget Constraints for Adaptive Applications in Cloud Environments," *IEEE Trans. Services Comput.*, vol. 5, no. 4, Oct. 2012, pp. 497-511.

[20] H. Song, J. Li, and X. Liu, "IdleCached: An Idle Resource Cached Dynamic Scheduling Algorithm in Cloud Computing," *Int. Conf. UIC/ATC*, Fukuoka, Japan, Sept. 4-7, 2012, pp. 912-917.

[21] J. Rao et al., "QoS Guarantees and Service Differentiation for Dynamic Cloud Applications," *IEEE Trans. Netw. Service Manag.*, vol. 10, no. 1, Mar. 2013, pp. 43-55.

[22] Z.-W. Yuan and X.-G. Sang, "A Study on Resource Scheduling Strategy in the Enterprise Service Cloud," *Proc. ICSAI*, Yantai, China, May 19-20, 2012, pp. 854-857.

[23] Y. Jin et al., "CoDaaS:An Experimental Cloud-Centric Content Delivery Platform for User-Generated Contents," *IEEE ICNC*, Maui, Hawaii, USA, Jan. 2012, pp. 934-938.

[24] D. Niu et al., "Quality-Assured Cloud Bandwidth Auto-Scaling for Video-on-Demand Applications," *Proc. IEEE INFOCOM*, Mar. 25-30, 2012, pp. 460-468.

Tai-Won Um received his BS degree in electronics and electrical engineering from Hongik University, Seoul, Rep. of Korea, in 1999 and his MS and PhD degrees from the School of Engineering, Information and Communications University, Daejeon, Rep. of Korea, in 2000 and 2006, respectively. He is currently a senior researcher with ETRI, Daejeon, Rep. of Korea.

Hyunwoo Lee received his BS degree in electronics engineering from Hankuk Aviation University, Kyonggi, Rep. of Korea, in 1993 and his MS and PhD degrees in communication and information engineering from Hankuk Aviation University, Kyonggi, Rep. of Korea, in 1995 and 2005. Since 1995, he has been working as a senior researcher in the Broadband Convergence Network Interworking Laboratory, ETRI, Daejeon, Rep. of Korea. He is currently working as the director of the Media Networking Research Section, ETRI.

Won Ryu received his BS degree in computer science and statistics from Pusan National University, Busan, Rep. of Korea, in 1983 and his MS degree in computer science and statistics from Seoul National University, Seoul, Rep. of Korea, in 1988. He received his PhD degree in information engineering from Sungkyunkwan University, Kyonggi, Rep. of Korea, in 2000. He is currently working as the managing director of the Intelligent Convergence Media Research Department, ETRI.

Jun Kyun Choi received his BS degree in electronics from Seoul National University, Seoul, Rep. of Korea, in 1982, and his MS and PhD degrees from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, Rep. of Korea, in 1985 and 1988, respectively. He worked for ETRI from 1986 to 1997 and is currently working as a professor with KAIST.