Dynamically Scaling Multimedia Streaming Service on Hybrid Telco Cloud

Duong Quoc Trong, Heiko Perkuhn, and Daniel Catrein

Abstract— **Given the control over the access transport networks, telecom operators are well positioned to address many of the concerns relate to cloud computing. However, operators are at a disadvantage with respect to higher-level services or cloud computing deployments that rise above the foundational layer or involve the connection of multiple clouds. Therefore, there is a big concern: "how could operators leverage the potential strengths of their telecom networks to gain a competitive advantage instead of moving slowly and seeing revenue disappears into the cloud?".**

In this paper, we propose an applicable architecture for a multimedia streaming service on the hybrid Telco cloud that demonstrate how operators can capitalize on their potential strengths to gain competitive advantage and thereby attain new revenue streams. Operators can exploit their local presence (nearness to end users) and control of the access network to add dynamically scalable communication to the cloud service offering. A prototype is deployed and configured to both simulate for scalable streaming service in the proposed architecture and show ways of solving the associated technical issues.

***Index Terms*— Content Distribution, Dynamic Scaling, Hybrid Cloud Computing, Multimedia Streaming, Telco Cloud**

I. INTRODUCTION

Cloud computing has emerged as one of the hottest concepts in ICT today. Everybody wants a piece of the cloud. Telecom operator is not an exception and they really need to move forward. There are two main reasons why Telcos should consider becoming engaged in cloud computing.

The first is to gain the benefits of cloud computing for IT optimization (reduces costs, avoid over provisioning and increase elasticity and speed). Many of the applications and services in today's products are bundled with hardware and dimensioned according to a traffic model. These hardware/software bundles must cater for a peak load scenario that leaves the hardware underutilized most of the time. Thus, operators could build internal clouds to unbundle the software from hardware and make better use of the underlying hardware. These thereby increase utilization, flexibility and reduce the costs of their own operations [2].

The second is to exploit new business opportunities [2]. By increasing the utilization of internal resource, i.e. underlying hardware, cloud computing allows operators to use the excess for additional services. Therefore, operators start to begin themselves as service providers. For example, AT&T, Level 3 and Deutsche Telekom have provided content distribution services. Computing infrastructure services were introduced by AT&T, BT and Verizon also.

Given the control they exercise over the access transport networks, operators are well positioned to address many of the concerns relate to cloud computing (more details in [16]). Therefore, telecom operators have great opportunities to generate new revenue streams by utilizing their telecom unique assets as enablers to provide tailored cloud services for industries like media, auto, utilities and governments.

However, operators are at a disadvantage with respect to higher-level services or cloud computing deployments that rise above the foundational layer or involve the connection of multiple clouds. Most telecom operators, even those with substantial application hosting skill sets, lack market credibility when it comes to the most ambitious and potentially lucrative cloud services agendas [5]. They can't develop and deploy cloud services as well, quickly as traditional IT companies like Google, Microsoft, and IBM... Therefore, there appears a big concern: "how could operators leverage the potential strengths of their telecom networks to gain a competitive advantage instead of moving slowly and seeing revenue disappears into the cloud?"

Addressing above questions, in this paper, we will demonstrate how operators can capitalize on their potential strengths to gain competitive advantage and thereby attain new revenue streams. In order to do that, the first contribution of this paper is to characterize a class of competitive streaming cloud services that telecom operators could provide. These services are based on the cloud computing trend and especially the characteristics of the telecom infrastructure. On the one hand, they have the capability to enhance the user's experience. On the other hand, they can scale dynamically to serve appropriately workloads which always change continuously.

To verify this, we propose an applicable service architecture in which a telecom network (consists of primary sites and secondary sites) as well as third party data centers (represented by a public cloud, i.e. Amazon EC2) are taken into account in the deployment of application servers. Therefore, the service could exploit the local presence of secondary sites to enhance the end-user's experience through edge delivery methods and utilize public clouds to handle

Manuscript received July 15, 2011; revised August 03, 2011.

Duong Quoc Trong was with Ericsson GmbH, Herzogenrath, Germany and The Sirindhorn International Thai-German Graduate School of Engineering (TGGS), King Mongkut's University of Technology North Bangkok (KMUTNB), Thailand. He is now with Ericsson Services, Dusseldorf, Germany (mobile phone: +49 1578-717-5080; e-mail: trongdq@gmail.com).

Heiko Perkuhn is with Research Multimedia Technologies, Ericsson GmbH, Ericsson Allee 1, 52134 Herzogenrath, Germany (e-mail: heiko.perkuhn@ericsson.com).

Daniel Catrein is with Research Multimedia Technologies, Ericsson GmbH, Ericsson Allee 1, 52134 Herzogenrath, Germany (e-mail: daniel.catrein@ericsson.com).

unpredictable peak loads. Particularly, we choose a multimedia streaming service to represent the targeted service group for two reasons. First, multimedia streaming has distinguishing characteristics which make it challenging. Second, multimedia content has grown substantially over the past years and gained the largest share of the data traffic in mobile networks [8].

The second contribution of this paper is a proof-of-concept prototype. This is a complete system that provides a live streaming service to the end-user. Signals from a satellite are encoded into a stream which will be sent to streaming servers to serve client requests. The prototype shows that operators completely could deploy streaming servers dynamically into distributed telecom sites (primary and secondary) located worldwide as well as public clouds to meet the needs of clients. This is the most important requirement for the architecture in the first contribution. Thus, the prototype shows an approach to solve related technical issues to make the streaming service work and thereby provide a practical evidence for the previous contributions.

The rest of this paper is organized as follows: Section II provides a background on Telco cloud including typical mobile network topology analysis and basic Telco Cloud developments. Then Section III extends it to the architecture for the multimedia streaming service on the Hybrid Telco Cloud. After that, Section IV shows a prototype to demonstrate for the introduced scalable streaming service and explains the important ideas behind the deployment. In the end, Section V concludes the paper and determines further steps and directions for future researches.

II. BACKGROUND ON TELCO CLOUD

A. Telecom network topology and site model

The following is based on the work in [9], where the authors introduce a network topology and site model of typical mobile networks of today, comprising geographically separate primary and secondary sites. A typical mobile network consists of about 3 to 5 primary sites and 5 to 15 secondary sites.

Fig. 1 depicts schematically the structure of such a network. Primary sites host the main functional elements of a mobile network. This includes service nodes, servers and external network connectivity. They can be compared with data centers and are interconnected via a backbone.

Secondary sites serve as concentration and distribution sites. They are typically much smaller than primary sites and host network equipment like media gateways (MGw), radio network controllers (RNC) or base station controllers (BSC). Secondary sites are also suited to host server type nodes, such as caches or media streaming servers. However if these exist in today's architectures, they are statically deployed accruing to long term capacity and demand predications.

Typically, secondary sites are connected to two primary sites (dual-homed) for redundancy reasons. Note that Fig. 1 only shows a single connection to ease readability. Secondary sites connect to external networks and other sites via the primary sites they are connected to. However, with the increase in mobile data traffic, e.g., due to the evolution of HSPA and the rise of LTE, secondary sites also tend to have a direct connection to external networks, i.e., the Internet.

Radio network equipment can be connected to both, primary sites and secondary sites. This can be experienced in almost all deployed mobile radio networks.

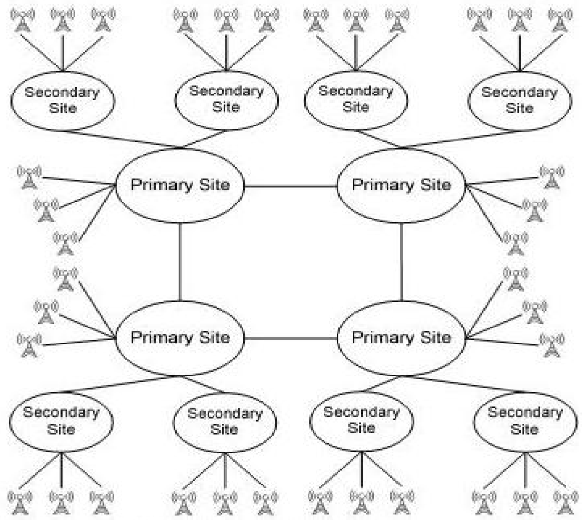

Figure 1. Schematic structure of a mobile network

B. Extending to Telco Cloud

Operators have already in hand a set of data centers facilities and high-bandwidth pipes to the end user. As so many enterprises already have virtualized their data centers, it shouldn't be too difficult to transform this infrastructure to Cloud based architecture. Telco Cloud will be established once operators fully transform their infrastructure into one or several clouds and then connect them together as a distributed resource pool.

In this way, telecom operators do provide more than simply a set of data center facilities and a pipe to the end user. Operators give customers access to a highly efficient virtual environment through a portal, supplying resources that run the gamut from firewalls and load balancers to the servers themselves. Rather than controlling network and computation facilities as separate entities, operators can optimize resource allocations by considering network and computing resources as a unified whole [2].

III. ARCHITECTURE OF MULTIMEDIA STREAMING SERVICE ON HYBRID TELCO CLOUD

A. Design Rationale

As shown in section I and II, telecom operators really need to move forward to cloud computing and actually some of them have started. In near future, they could engage in cloud computing in different directions. In this section, based on the cloud computing trend and especially the characteristics of telecom infrastructure (section II_A), we will characterize a competitive class of cloud services that operators could provide.

To characterize these services, first of all we address a class of applications and services that requires low latency and high bandwidth which are difficult to perform well and deliver the expected quality of experience (QoE). Particularly, we choose a multimedia streaming service to represent the targeted service group because of two reasons. First, multimedia streaming, with its distinguishing characteristics, requires support on delay guarantees, bandwidth reservation, and flexible error control, which makes it challenging [10][11]. Second, multimedia content has grown substantially over the past years and gained the largest share of the data traffic in mobile networks [8].

Standing on this point, we will propose an applicable architecture for a multimedia streaming service on the Hybrid Telco Cloud. In this architecture, the telecom network (consists of primary sites and secondary sites) as well as third party data centers (represented by public cloud, i.e. Amazon EC2) are taken into account for deployment of application servers (which are the streaming servers in the rest of the paper). Therefore, the service on one hand would exploit strengths of the operators' mobile networks (local presence of secondary sites) to enhance end user experience through edge delivery methods. On the other hand, it can be scaled dynamically, even to public clouds, to meet the changing demand of clients without worrying about unpredictable peak loads. While the first requirement is resolved quite well (but not yet mature), especially with the emergence of CDN and edge delivery methods [10], the second one is somehow more open, particularly in handling peak load ([6], section II_A).

B. Architecture

Fig. 2 shows the applied multimedia streaming system on the Hybrid Telco Cloud. The architecture consists of 3 parts: the private Telco cloud, the public cloud and the cloud manager.

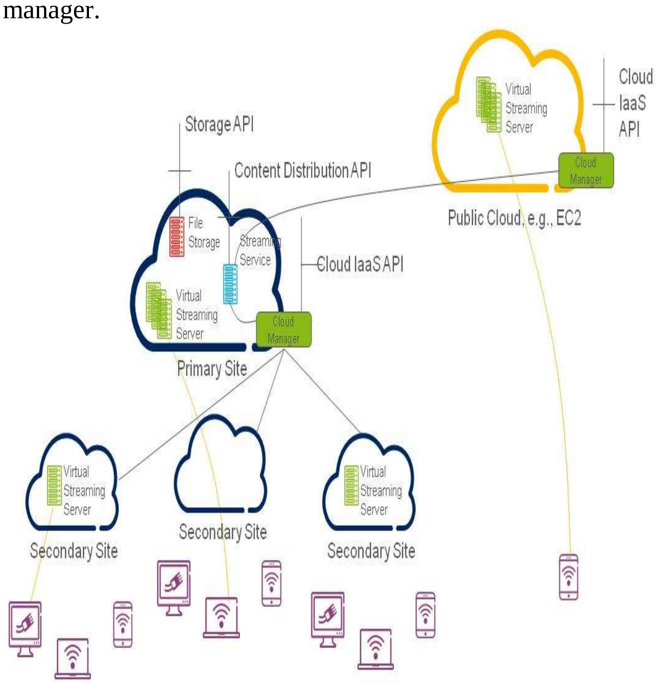

Figure 2. Architecture of streaming service on Hybrid Telco Cloud

The private Telco Clouds: This component is the Telco Cloud described in Section II_B. Currently, involved telecom companies and researchers are working on extending current telecom network infrastructure to Telco Cloud. There are more and more promising results obtained [2][3][4][5]. Its realization is more or less just a matter of time. In this paper, we will look at one step further by assuming that the Telco Cloud has been established and all the sites (primary and secondary) are extended into cloud as well as connected successfully. Then the whole telecom network will become a unified distributed resource pool.

The public clouds: Besides internal resources, in this system, we also use external resources, i.e. from public clouds, to handle/offload the peak load if the installed capacity within the network is exceeded. In special events that attract a lot of audiences watching at the same time, i.e. World Cup, Olympic...the number of clients/requests can be rapidly increased to an unexpected number which is out of the operators' capacity. Thus, in order to handle the peak load, operators could rent the exceeded resources from a public cloud and just pay for what they use.

The Cloud Manager: Practically the public and private clouds used in the architecture will rely on different technologies. This imply different virtualization technologies, different application program interfaces (APIs) to request and control resources, as well as different network and connectivity environments.

Therefore, we introduce in the heart of the architecture a common management framework, so-called cloud manager. It provides a single interface to applications, such as the streaming server, in order to request new resources and release those that are idle. The cloud manager allocates the resources by creating and configuring Virtual Machine (VM) images in different clouds.

More specifically, the cloud manager contains a flexible deployment engine that matches on one hand the properties of clouds, available VM images, and on the other hand the requirements of applications (required OS, i.e. Ubuntu, or environments, i.e. Java environments). This engine enables deployments in hybrid cloud environments without having to generate specific VM images for each application and cloud. Instead, all VM instances will be created and configured automatically at boot time by applying scripts onto a few template virtual machines [2].

The Cloud APIs: Besides the three components above, we can see that there are also three Cloud APIs presented in the service architecture: The Cloud IaaS APIs, which allow cloud infrastructure to be added, reconfigured, or removed in real time; The storage APIs allows content providers worldwide to store their media content in their own way; The content distribution APIs allow content providers to implement their strategy in delivering content through system's network.

Fig. 3 shows these APIs together with its Telco Cloud's component. While storage and content distribution APIs come from functional component, IaaS API [1] comes from management component. There is only content distribution API provided from SaaS layer [1]; others are in PaaS layer [1]. These 3 types of extensive API above are very important as it allows third parties content provider to participate and deliver their multimedia content. The variety of the content will determine the success level of streaming service.

Working mechanism: Based on the demand, the cloud manager will decide where and when to deploy the virtual machine (VM) instances. The deployment of VMs includes two steps. Firstly, "empty" VMs (without a specific application server) would be deployed into secondary sites,

ISBN: 978-988-18210-9-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

primary sites or public clouds. After that, applications, i.e. streaming servers are installed into those "empty" VMs by scripts. These configurations are performed automatically at boot time.

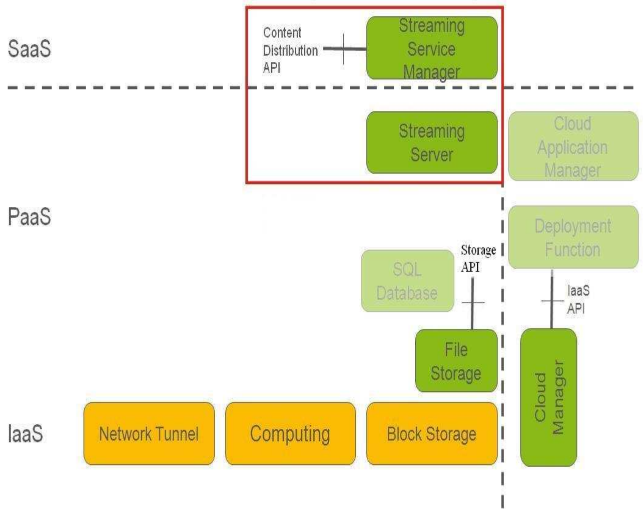

Figure 3. Teco Cloud's Components and APIs

When a client streaming request is sent to the cloud manager through a website or portal, the cloud manager will determine the location of the request and then redirect that request to a streaming server in the respective network that is close to the client who sent the request. This helps to reduce latency, avoid bottle necks and thereby improve quality and user experience.

To control the number of instances dynamically based on request and demand, the cloud manager will determine one "warning threshold". At the beginning when the first request is sent to the cloud manager, it will create a new VM instance to serve this request. As the number of requests increases until it reaches to the threshold, all the requests are still served from that VM instance. If the number of requests continues to increase and exceed the threshold, the cloud manager will start new instances to serve these new exceeded requests (illustrated in Fig. 4).

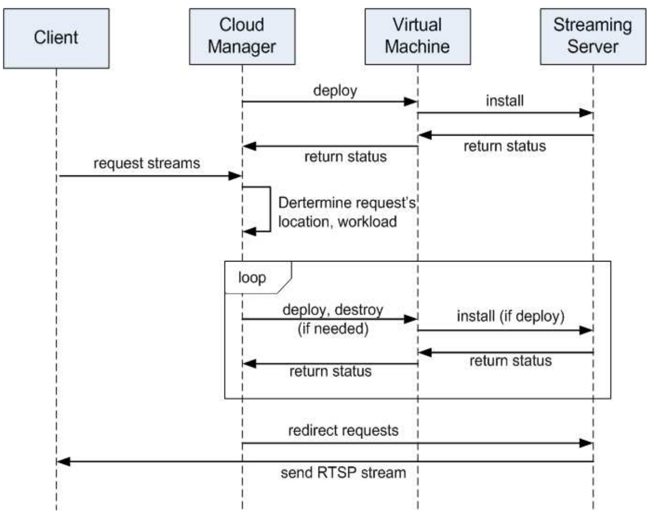

Figure 4. Working mechanism of proposed streaming service

Vice versa, at certain times when the number of requests is decreased so that they need fewer servers than the current available servers, there will appear one or several servers which are idle - doesn't serve any client. Thus, these servers therefore would be terminated to both release resources for another application and reduce operation cost (illustrated in Fig. 4).

C. Advantages and Evaluation

The proposed streaming cloud service above can guarantee both requirements rose in design rationale section: ensure good user experience while scale up easily.

Firstly, it is clearly seen that it has exploited telecom operators' strengths to ensure good user experience. In this architecture, the streaming server is brought out of a centralized data center to the far away sites which are closer to the user. This is here where the strengths of operator show up. With the widespread network of telecom sites, i.e. 15-20 sites per country (section II_A), operators can provide global reach with domestic and international streaming points of presence. This is not mention the access network with very large numbers of Mobile stations (MS) and Base Station (BS) which give operators a strong local presence. Meanwhile, Amazon CloudFront for example provides only 17 edge locations world-wide as of December, 2010 (10 in USA, 4 in Euro, 3 in Asia). The short distance between operators and users would reduce round-trip time, package loss and avoid congestion. As a result, multimedia streams are delivered with high quality and thereby enhance user experience. Besides, it is also easy to address relevant concern such as trust or security...

Secondly, with the described dynamical scaling characteristic, the Telco streaming cloud service in the architecture above allows operators to handle workload efficiently. By controlling the number of streaming server virtual machines, the cloud manager can allocate resources in both private Telco clouds and public clouds to meet the changing needs of the streaming service. This brings great benefits to operators. Firstly, they can exploit resources the most. At every moment, each service will get only required resources and leave the rest for other services; therefore, there isn't any resource that will be left underutilized. Additionally, by automation of deployment and configuration, operators can reduce time and effort for operating and maintaining applications also. More importantly, the streaming cloud service allow operators to avoid the over provisioning issue in handling peak load. They don't need to worry if the peak load is out of their capacity because they can rent resources from public clouds. There are no long-term contracts, no additional charges; they simply pay for what they use.

Thus, together the Telco Clouds and Cloud manager make it possible to manage the demand for computing and network resources instantaneously, and to meet changing service needs more quickly and efficiently than today. The proposed service architecture, therefore, sharpen the new business opportunity for operator in offering competitive cloud services.

IV. PROTOTYPE

In this Section, we describe the setup of a prototype to demonstrate for the dynamically scaling characteristic of the streaming service in Section III: Operators could deploy streaming servers dynamically into distributed telecom sites

ISBN: 978-988-18210-9-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

located worldwide as well as public clouds to meet the needs of clients. This prototype is a complete live streaming system to provide live streaming service to the end-user. The signal from satellite is encoded to stream and sent to the streaming servers to serve requests from remote clients.

A. Prototype Setup

The system architecture is presented as in the Fig. 5. It consists of two parts: the local data center and the public cloud Amazon EC2 [7]. In the local data center, we utilize the existing infrastructure that includes: the cloud manager which can allocate resources in hybrid cloud [2] and two private clouds: Eucalyptus [12] and VMware [13].

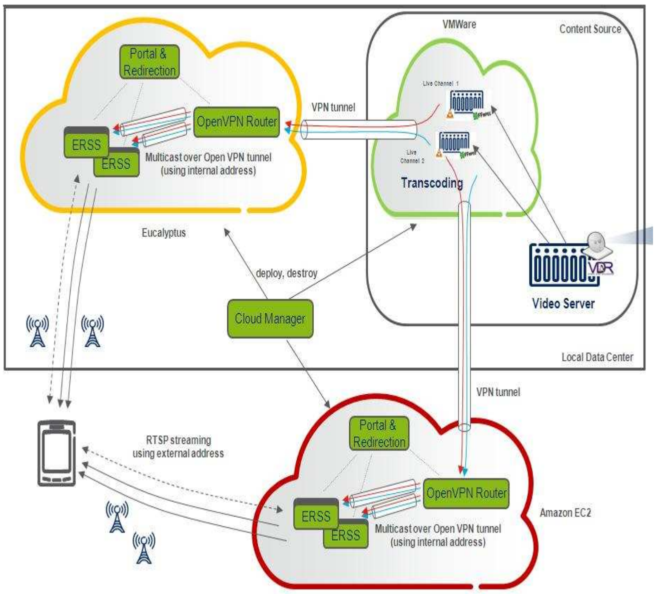

Figure 5. Prototype setup

Live content source: To provide live content source, we choose the solution of using a live broadcast TV program which can be fetched from satellite signal. One PC, equipped with a satellite receiver, is dedicated for that purpose. It is installed VRD (Video Disk Recorder), an open source application for Linux, in order to record and replay TV programming (in mpeg file format). Then this mpeg file will be transcoded into transport streams (mpegts) which is the required input of the Ericsson streaming server introduced in next section. In this prototype, we use FFmpeg [14] transcoder.

The transcoder is deployed in the VMWare cloud for the purpose of easy scalability and multiple format support capability. In the picture we have, for example, two encoders to encode for two live channels. We can even deploy more encoders to increase encoding capacity or support more channels.

Streaming server nodes: The streaming server node is a virtual machine (VM) which is installed the Ericsson streaming server (ERSS). This streaming server receives RTP/RTCP stream as the input and provide RTSP stream as the output. The output RTSP protocol defines control sequences which are useful in controlling multimedia playback, i.e. SETUP, PLAY, PAUSE, RECORD...

To add a live source, an SDP-file describing the incoming streams must be put in an appropriate directory. The SDP file determines which incoming ports the server should listen to. The server periodically scans for new or changed SDP-file and starts or stops listening to the appropriate ports. It then starts to buffer the media, and indicates Intra pictures[1] (or in general, Random Access Points) for where to start a client session. Thereby, a new user always starts at an Intra picture.

To demonstrate the dynamically scaling characteristic of the system based on the demand as in Section III, the cloud manager will dynamically deploy streaming server nodes to private cloud Eucalyptus or public cloud Amazon EC2.

OpenVPN server nodes: While the content source comes from FFmpeg transcoder in local data center, the streaming server can be placed anywhere, private cloud in local center or public cloud in the internet. Therefore, we consider two noticed points in sending the live content stream from FFmpeg transcoder to streaming server:

- First, because of the timely delivery and package-loss-tolerance requirements of real-time multimedia streaming applications, we choose to use VPN tunnel to send content stream from FFmpeg to ERSS instead of TCP/IP protocol.
- Second, if we send the stream from FFmpeg directly to each ERSS server follow unicast model, it would cause bottle neck at the FFmpeg encoder and also cost more money for data transfer into Eucalyptus and EC2 cloud (where place all the ERSS server).

Addressing these two points, we simulate the multicast model by introducing an intermediate OpenVPN server [15] node as in Fig. 5. This node acts as a router for all streaming servers deployed in the Eucalyptus cloud or EC2 cloud. It receives stream from transcoder through OpenVPN tunnel and then re-distribute/broadcast received stream to all Streaming Server (ERSS) nodes also through OpenVPN tunnels. This model helps to reduce the outgoing traffic at transcoder and also reduce data transfer in/out of Amazon EC2, so it's cheaper.

To route all packages follow the scenario above, we deployed 2 VPNs as in the Fig. 6: The first one, OpenVPN1, includes all streaming servers ERSS which connect to OpenVPN server through virtual interface 0. The second one, OpenVPN2, includes all transcoder which connect to OpenVPN server through virtual interface 1. The packages will be routed as below: Firstly, the transcoder in VPN2 will send stream to OpenVPN server through tap 1. Then, OpenVPN server has to forward that streams from tap 1 to tap 0, and forward to all ERSS in VPN1.

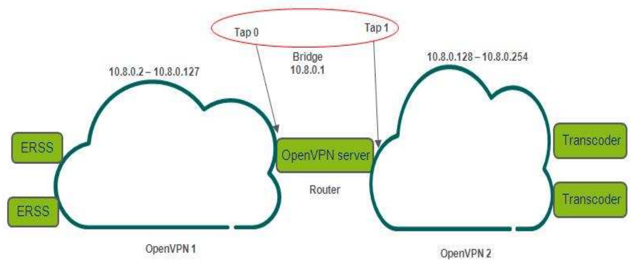

Figure 6. OpenVPN connection

[1] In video compression, Intra-Picture (I-Picture) is one of 3 different picture types. The others are Predictive-coded Picture (P-Picture) and Bidirectionally-coded Picture (B-Picture). I-Pictures are coded using information present only in the picture itself and not depending on information from other pictures. Therefore, they provide a mechanism for random access into the compressed video data.

Working flow: The working flow of the system could be summarized as following. Firstly, the video server receives the satellite signal and converts it into digital media (mpeg). After that the FFmpeg transcoder in the VMware cloud will encode that media file into transport stream format (mpegts) before sending it to OpenVPN server node. To its turn, OpenVPN server node will act as a router; it receives transport stream in one interface and forward it to another interface. Next, the transport stream is broadcasted to all ERSS streaming servers which are connecting to the second interface. Subsequently, each streaming server receives transport streams from OpenVPN server and processes them to generate RTSP stream. Here all the streaming servers are deployed into either the local private cloud Eucalyptus or the distributed edge sites of public cloud Amazon EC2 located worldwide. Finally, when user comes to portal and sends request for playing, requests will be redirected to closest ERSS and will be served from there.

B. Demonstration and Analysis

After all the components above were set up and configured successfully, we have run a demonstration. In our demonstration, first of all we deployed one OpenVPN server, one FFmpeg encoder, and one streaming server ERSS. It took only 7 - 9 minutes to deploy all of these three nodes, and 2-3 minutes for each node. Then users can start playing streaming from mobile or PC. To terminate a VM, it is even faster: it took just 1 minute.

To demonstrate the scaling capability of the service as in Section III (Working Mechanism), we assumed that the number of requests increases, so we deploy several streaming servers ERSS more to serve new requests. To deploy an ERSS server, it took only 3 minutes. Additionally, assuming the number of requests goes down, we destroyed some streaming server. To destroy a server, it took only 1 minute.

To illustrate how the service could move the peak load onto public clouds, we assumed that the Telco Cloud has been run out of capacity while new requests are still coming. In this case, operators need to rent external resources to serve these new requests; otherwise, the service will not work. Thus, instead of deploying the OpenVPN server and streaming servers onto the private cloud Eucalyptus, we deployed them onto the public cloud Amazon EC2. The deployment was performed successfully within similar time as above (2-3 minutes/server).

In summary, the prototype has proved that operators completely could deploy streaming servers dynamically into distributed telecom sites located worldwide as well as public clouds to meet the needs of clients as described in section III, thereby provide a practical evidence for the proposed service architecture.

V. Conclusion And Outlook

In this paper, base on widely deployed mobile network topology and cloud computing trend in telecom industry, we have looked into the time when the Telco Cloud is fully established and proposed a multimedia streaming service on Hybrid Telco Cloud. On one hand, the service exploits telecom operator's strength of local presence to deploy streaming servers to secondary sites which are close to user. This helps to reduce round-trip time and avoid congestion, thereby enhance users' experience. On the other hand, the service can be scaled dynamically depend on the changing demand. Required resources (computing, storage, and network) in both private and public clouds are allocated for the service by the cloud manager through virtual machines. Therefore, operators can utilize their underlying hardware the most as well as avoid over provisioning by moving the peak load to public cloud.

In addition, a prototype is deployed to demonstrate for the dynamically scaling characteristic of the service. This prototype not only resolves related technical issues to make the service works but also considers applying multicast streaming model to reduce the total cost. It can be used as a basement for ongoing researches.

For future works, resource allocation can be optimized more. For example, find out servers' placement and serving schemes that brings minimized total cost. By understanding the traffic patterns, operators can apply different workload prediction strategy to let cloud manager know the anticipated demand and then scale application up and down with proper planning. In this way, the service can be scaled proactively, instead reactively as the moment. Besides, further components such as fault tolerant, SLA, QoS, etc are needed to determined, measured and developed as well.

References

[1] Borko Furht and Armando Escalante. "*Handbook of Cloud Computing*". Springer, ISBN 978-1-4419-6523-3, 2010.
[2] Jan Gabrielsson, Ola Hubertsson, Ignacio Ms and Robert Skog. "*Cloud computing in telecommunications*". Ericsson Review, Jan 2010.
[3] Jan Gabrielsson, Per Karlsson and Robert Skog. "*The cloud opportunity*". Ericsson Review, 2009.
[4] Arpit Joshipura. "The *cloud: what operators stand to gain or lose*". Ericsson, Feb 2010.
[5] Amy Larsen DeCarlo. Opinion: "*Don't enter the cloud alone*". Ericsson, Feb 2010.
[6] H. Zhang, G. Jiang, K. Yoshihira, H. Chen, and A. Saxena. "*Intelligent Workload Factoring for A Hybrid Cloud Computing Model*". IEEE 7th International Conference on Web Service (ICWS 2009), July 2009.
[7] "Amazon EC2". *http://aws.amazon.com/ec2/*.
[8] Kevin Fitchard. "*Streaming surpassing P2P in clogging mobile networks*". Connected Planet Website, 2009.
[9] Daniel Catrein, Bernd Lohrer, Christoph Meyer, Ren Rembarz, Thomas Weidenfeller. "*An Analysis of Webb Caching in Current Mobile Broadband Scenarios*". 4th IFIP International Conference on New Technologies, Mobility and Security (NTMS 2011), 2011.
[10] T. Nguyen and A. Zahkor. "*Distributed video streaming over the Internet*". In Proceedings of Multimedia Computing and Networking, San Jose, California,USA, 2002.
[11] Ma Lin. "*Techniques and protocols for distributed media streaming*". PHD thesis submitted for National University of Singapore, 2007.
[12] Eucalyptus. *http://open.eucalyptus.com/*.
[13] VMware. *http://www.vmware.com*.
[14] FFmpeg. *http://www.mpeg.org/*.
[15] OpenVPN. *http://openvpn.net/*.
[16] Duong Quoc Trong. "*Optimization and Evaluation of a Multimedia Streaming Service on a Hybrid Telco Cloud*". Master thesis under the cooperation of RWTH Aachen University and Ericsson GmbH, Germany, 2011.