On the Investigation of Cloud-based Mobile Media Environments with Service-Populating and QoS-aware Mechanisms

Fragkiskos Sardis, Glenford Mapp, Jonathan Loo, Mahdi Aiash, and Alexey Vinel

***Abstract*—Recent advances in mobile devices and network technologies have set new trends in the way we use computers and access networks. Cloud Computing, where processing and storage resources are residing on the network is one of these trends. The other is Mobile Computing, where mobile devices such as smartphones and tablets are believed to replace personal computers by combining network connectivity, mobility, and software functionality. In the future, these devices are expected to seamlessly switch between different network providers using vertical handover mechanisms in order to maintain network connectivity at all times. This will enable mobile devices to access Cloud Services without interruption as users move around. Using current service delivery models, mobile devices moving from one geographical location to another will keep accessing those services from the local Cloud of their previous network, which might lead to moving a large volume of data over the Internet backbone over long distances. This scenario highlights the fact that user mobility will result in more congestion on the Internet. This will degrade the Quality of Service and by extension, the Quality of Experience offered by the services in the Cloud and especially multimedia services that have very tight temporal constraints in terms of bandwidth and jitter. We believe that a different approach is required to manage resources more efficiently, while improving the Quality of Service and Quality of Experience of mobile media services. This paper introduces a novel concept of Cloud-Based Mobile Media Service Delivery in which services run on localised public Clouds and are capable of populating other public Clouds in different geographical locations depending on service demands and network status. Using an analytical framework, this paper argues that as the demand for specific services increases in a location, it might be more efficient to move those services closer to that location. This will prevent the Internet backbone from experiencing high traffic loads due to multimedia streams and will offer service providers an automated resource allocation and management mechanism for their services.**

I. Introduction

Cloud computing is a relatively new trend in Information Technology that involves the provision of services over a network such as the Internet. The cloud services offered are divided in three categories: Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS) as illustrated in Fig. 1. SaaS delivers software applications such as word processing over the network. PaaS delivers a host operating system and development tools that come installed on virtualised resources. Such Cloud services are now being used to support Video-on-Demand (VoD) services which have much more demanding Quality of Service (QoS) constraints. Finally, IaaS offers raw resources such as a number of virtual machines or processors and storage space and leaves it up to the user to select how these resources are used.

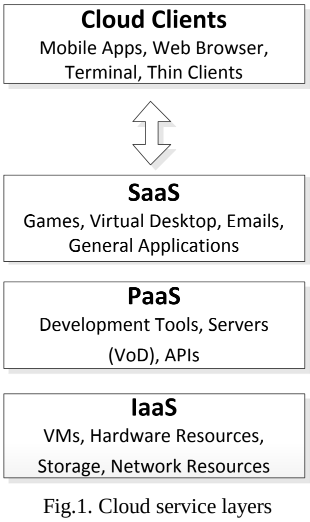
Cloud Clients
Mobile Apps, Web Browser, Terminal, Thin Clients
SaaS
Games, Virtual Desktop, Emails, General Applications
PaaS
Development Tools, Servers (VoD), APIs
IaaS
VMs, Hardware Resources, Storage, Network Resources

Fig.1. Cloud service layers

Cloud services are elastic in the sense that they are provided on demand. The provider manages the delivery of services and the clients can demand as little or as many resources as they require and are billed accordingly. From the client side, all that is needed is a computer with a web browser or a thin client with the ability to remotely connect to the Cloud. This simplicity of requirements for the client has created a high demand for Cloud computing and has paved the way for more Cloud-based research and development. The trend to

Manuscript received March 5, 2012, revised July 18, 2012, accepted November 13, 2012.

F. Sardis, G. Mapp, J. Loo and M. Aiash are with the School of Science and Technology, Middlesex University, London, NW44BT, UK (e-mail: {f.sardis, g.mapp, j.loo, m.aiasah}@mdx.ac.uk).

A. Vinel is with Tampere University of Technology, Finland and part-time with Halmstad University, Sweden (e-mail: vinel@ieee.org).

This joint work has been done in the framework of EU COST Action IC0906 “WiNeMO: Wireless Networking for Moving Objects”.

centralise processing and storage resources and outsource I.T. infrastructure management and maintenance has been the driving force for many big vendors to create their own Cloud services and offer them to businesses and individuals alike. Furthermore, this trend has negated the need for powerful client computers and has opened the way for smaller, lighter and more portable devices such as mobile phones and tablets. Some examples of Cloud-based products that are very popular nowadays are Amazon's EC2 and Apple's iCloud [1]. Each of these vendors has their own vision of Cloud-based services so their approach to it is different according to the market they are targeting.

Mobile devices nowadays come in different shapes and forms. Perhaps the most popular form is laptops, although they are not truly portable in the sense that we cannot operate one while on the move due to the size and form factor. This has created a demand for devices that are more mobile and easier to use for someone on the move and away from a power source. The devices that filled this gap and created a new trend in mobile computing are smart phones and tablet PCs. Unlike laptops and desktop computers, these mobile devices are made for a long-lasting battery life, a small size and weight, a simple user interface and run basic computing tasks using limited resources such as memory, etc. As such, they lack the hardware resources necessary to perform intensive tasks. The very nature of mobile devices dictates their form factor and prohibits the use of hardware with a wide range of capabilities. Due to the limited local resources on these devices the focus for future development on them is shifting towards always-on connectivity via the use of multiple network interfaces such as Wi-Fi [2], GSM [3], 3G [4] and LTE [5] so that they do not have to rely on local resources for storage and processing but instead access resources remotely via a network.

With Cloud-based services on one side offering affordable and centralised computing resources, and mobile devices on the other side, demanding for a centralised pool of resources to make up for their lack of processing power, we now see a connection between those two technologies that will allow future development in both areas of research.

In this paper we present a potential scenario in the future that can create traffic congestion problems on the Internet due to high bandwidth media services and user mobility. We use an analytical framework to investigate the factors that affect the Quality of Experience (QoE) and QoS for VoD services in such a mobile environment. Finally, we investigate a service delivery framework that can overcome such a problem by the use of service populating techniques and Cloud services.

The paper is outlined as follows: Section II presents the current state of some recent research in the area of Cloud services. In Section III we present a service delivery scenario of the future. In section IV we look at how QoS can be affected in a mobility scenario. In Section V we present the investigated framework for service delivery. Section VI examines some of the mechanisms of the framework and how they relate to the use case. Section VII presents potential applications of this framework and Section VIII concludes this paper.

II. State-of-Art of Cloud-Based Services

The development of Cloud-based service delivery is now moving rapidly as existing Cloud service providers attempt to revolutionise the concept while new vendors attempt to enter their market with their own versions of the technology. Three popular vendors are Amazon, Apple and Microsoft, while many more offer similar services or simplified versions of the same services.

Amazon's EC2 [6] is a Cloud solution that offers IaaS and bills the clients according to the time and resources have been using. In addition to services, EC2 offers storage that is accessible from anywhere on the Internet. Amazon's service offering are highly elastic, starting from micro instances that offer a small amount of virtualized resources, enough to cover very basic computational needs, to Cluster Compute solutions that allocate physical processors permanently to the clients. In addition to the above, Amazon also offers Cluster Graphics Processing solutions that are suitable for rendering and media processing applications.

iCloud, Apple's Cloud offering, is different type of Cloud compared to what Amazon is offering. Apple's solution provides storage services and the ability to synchronize files across multiple clients, including mobile devices. This gives clients the ability to store their calendars, contacts and emails, as well as iWork documents to the Cloud and have any changes in them consistently propagate to all their Apple devices. A new feature for iCloud is its ability to track geographically devices of a user which helps in finding lost devices although such features often raise privacy concerns regardless of service provider.

Microsoft is also offering a wide variety of Cloud-based services [7]. Their implementation of Cloud services apart from offering SaaS in the form of Office 365 is also offering PaaS in the form of Azure and also IaaS in the form of their Private Cloud implementation. Microsoft also offers a Cloud solution that acts as a central management point for the clients. Windows Intune is a Cloud solution that allows central management of all the connected client computers in ways such as malware detection, application deployment, software update rollouts and centralized software license tracking.

Regardless of vendor and the type of services offered, Cloud computing is used to centralize processing in a highly scalable and cost efficient manner. In fact, many Cloud providers are able to offer their services for free or at a very small cost to their clients. However, it is also important to look at the development of Cloud technology itself and not only at the development of services that run on top of it.

Researchers at the University of Minnesota are developing a migration technique for virtual machines within a Cloud that incorporates heterogeneity and dynamism in network topology and job communication patterns to allocate virtual machines on the available physical resources [8]. Their aim is to bring physically closer any virtual machines that exchange a lot of traffic with each other. This way, they can make use of faster connections within the same network hierarchical level instead of letting traffic go through slower connections between levels. Since what we call "Cloud" is actually a network of computers with a hierarchical structure, it becomes obvious that sometimes, there can be a lot of traffic between different hierarchical levels, depending on where data is stored and processed within the infrastructure. Moving virtual machines that carry out individual parts of a bigger task, closer to each other, will reduce this cross-boundary communication which often goes through slower network links compared to the much faster links that exist within the same hierarchical boundaries. The benefit of this is faster communication for the two VMs, which improves the overall performance and less network congestion within the infrastructure. This makes the use of Cloud resources more efficient, which results in lower costs for the provider and more savings for the clients.

Another research project by the University of Minnesota involves the reshaping of the physical footprint of virtual machines within a Cloud [9]. The aim is to lower operational costs for Cloud providers and improve hosted application performance, by accounting for affinities and conflicts between co-placed virtual machines. This is achieved by mapping virtual machine footprints and then comparing them. When similarities are found in the memory footprints, the virtual machines are migrated to the same physical location and content-based memory sharing [10, 11, 12] is employed to achieve consolidation without inducing performance penalties. The aim is to build control systems for Cloud environments that employ such footprint reshaping to achieve higher-level objectives such as lower power consumption, higher reliability and better performance. This better use of Cloud resources will also reduce costs for providers and make Cloud services cheaper for clients.

Another recently proposed architecture aimed at improving the performance of Cloud technologies is called Media-Edge Cloud (MEC). It is an architecture that aims to improve the QoS and Quality of Experience (QoE) for multimedia applications [13]. This is achieved by a "Cloudlet" of servers running at the edge of a bigger Cloud. The aim of this is to handle requests closer to the edge of the Cloud and thus reduce latency. If further processing is needed, then requests are sent to the inner Cloud, so the "Cloudlets" are reserved for QoS sensitive multimedia applications. In essence, the aim is to divide the network hierarchy within the Cloud, in such a way that physical machines that are closer to the Cloud's outer boundaries will handle QoS sensitive services. Since these machines reside on the border of the Cloud, the data has to travel less distance within the Cloud before it is sent out to the clients. This not only improves QoE for clients but it also reduces network congestion within the Cloud.

However, these new concepts and research into improving Cloud performance, do not take into account user mobility. Media delivery on mobile clients is the new trend in computing and mobile devices are the most likely to make use of Cloud resources in the future. Furthermore, all the research at present assumes that only one entity (the provider) is in control of a Cloud and as a result different providers cannot "share" resources in a manner that can improve the utilisation efficiency of their hardware. This can potentially lead to problems in the future as mobility and multimedia-rich content becomes more popular and high bandwidth data streams will have to travel great distances and reach moving targets. Cloud providers may find themselves in situations where their hardware resources are not adequate and they may have to create more Clouds to handle the load and relieve network congestion.

III. Envisioned Cloud-Based Service Scenario

In order to understand possible problems that may arise in the future of Cloud computing we will look at an example of a common use of Cloud resources. We will first look at how services are delivered at present and how this is bound to change.

At the moment, the Internet and networking in general works in a resource-centric way. This means that clients get services by contacting a physical resource directly and then asking for a service. By typing a URL for example, we essentially type the name of a server on the Internet. The name is resolved to an IP address and we then connect directly to that server in order to retrieve the service. Cloud services at the moment work in a similar fashion. Clients connect to the Cloud and they are presented with possible services they can access. The disadvantage of this approach is that users still have to know the name of a physical resource in order to reach a particular service and that if the physical resource offering the service is experiencing problems then there is little room for redundancy. Big corporations are able to address the redundancy problem by running multiple servers and using DNS [14] techniques for failover and load-balancing purposes. However, it is not a viable solution for smaller entities who want to offer a service at low cost

In the future of Cloud services, we envision the ability for clients to request services directly from the network rather than asking for physical resources that offer these services. This will simplify the process for end-users and open the way for other changes. In this service-oriented approach, we expect clients to simply request a Service ID and the network infrastructure to find where the service is running and connect the clients. This gives the possibility of running a service in multiple locations and directing client requests to the most appropriate instance depending on their location and network status.

In order to take network status into account when delivering services, we need a QoS aware service delivery model. This means that the network infrastructure should take into account what the network status is between the client and the service. Service providers will want to give a fairly high and consistent QoS and QoE to their clients. In our example, clients of Cloud services at the moment will connect to the same Cloud no matter their location or network conditions. However if network conditions deteriorate and there is no redundant path, the service will be out of reach or severely affected. This results in the provider failing to meet their SLA standards and the clients not getting the best QoE possible at all times. The other disadvantage with the present Cloud-service model is that clients from any geographical location have to connect to the same Cloud to get services that run on it, no matter how far they are from the Cloud. This potentially overloads network interfaces on the Cloud and also creates higher processing load on the Cloud itself which can further deteriorate QoS. Cloud providers are not in a position where they can easily build multiple Clouds to service different geographical areas like they do with services that run on individual servers. It is also not possible to use regional caching techniques on entire services that have active content. Therefore, a new method for service delivery is required, that will take into account QoS in order to provide better QoE to the clients and better load management to the providers, as well as help reduce network congestion on a global scale.

With the above service delivery model we will have clients requesting a service and their requests will be directed to the physical location where the service is running and also fulfils QoS criteria. However, if we introduce mobility to the scenario, it becomes harder to direct client requests to a specific instance of a service. We could connect a client to a service instance based on their present location and network conditions but if the client moves to another location with different network characteristics, we may lose all benefits. In addition, if we come into a situation where clients are moving farther away from the service, we add to the network congestion and depending on the type of service, this can have a big impact on QoS for everyone on the same network. To address this, we could connect the client to a different instance of a service every time QoS parameters deteriorate, however, we cannot expect Cloud providers to have multiple Clouds in different locations only for the purpose of addressing mobility problems and network congestion.

Although a single Cloud provider may not own multiple Clouds in different geographical locations, we can safely assume that many Cloud providers are will have their Cloud installations quite far apart on a global scale or even down to a regional scale within a country. This gives us the opportunity to investigate the concept of Service Population across different Cloud provider boundaries. We envision a scenario where Service Providers will register their services globally and will not be tied to a specific Cloud provider. These services will be free to “populate” Clouds or “jump” to a different Cloud depending on QoS parameters and source location of service requests. To achieve such thing, it means Cloud providers will have to “open” their Cloud boundaries so that services can move in and out of their Clouds depending on demand. This will make a big change in the business model of Cloud and Service providers. A service provider will simply register their service with a Service Level Agreement that will define expected QoS parameters. Cloud providers will be in competition to provide the best QoS so that the service will populate their Clouds and generate income for them. However this does not mean that the biggest Cloud provider will always take all the services, since location and network congestion parameters are taken into account. So we may see services moving out of a bigger Cloud and propagate into smaller ones in order to keep network congestion to a minimum and move itself closer to its clients. Clouds should also have the ability to decline a service if they are already under heavy load. This process should be automated and completely transparent to the users. It should also happen in real time without administrative intervention in order to provide a streamlined resource management solution for Cloud providers.

In order to address the problems identified in the example above, a new service delivery framework is necessary. This framework should be QoS aware and support active Cloud population with services.

IV. Analytical Approach in a Mobile Environment

In this section we will attempt to provide a basic analytical framework to analyse how mobility and network attributes affect the provision of a multimedia service such as Video-on-Demand (VoD) services. In VoD systems, entire videos are placed in memory on the server and client requests are serviced from this in-memory cache.

We start by defining the time to prefetch p blocks of data, which is given by:

$$T_{prefetch} = L + C \times p \tag{1}$$

In this equation, L is the network latency and $C$is the per-block time of copying data between the in-cache memory and network buffers. Ideally p should be at least equal to the number of blocks required to display a video frame of data. On a lightly loaded wired network we can consider these values constant for each link. However, in a mobile environment, L changes as the client moves and the number of network links increase. We can express L as follows:

$$L = F_{n,s,\theta} + F_{cloud} + F_{protocol} \tag{2}$$

where $F_{n,s,\theta}$is the latency incurred by the number of links(n) between client and service, the network bandwidth on each link(s_i)and the network load on each link(θ_i); F_{cloud} is the Cloud latency caused by the network topology and hierarchy within the Cloud [13] and F_{protocol} is the latency caused by the transport protocol.

If the time to prefetch p blocks is larger than the time it takes for the device to consume them, then we have jitter. This can be expressed as:

$$T_{prefetch}(p) \leq T_{cpu} \times p \tag{3}$$

where T_{cpu} as the time it takes for a device to consume a number of blocks by playing them as audio and video frames. T_{cpu} is therefore dependent on the type of video being displayed and the hardware capabilities of the mobile device.

We now substitute for $T_{prefetch}$ in Equation 3 with the expressions in Equations 1 and 2. Rearranging, we get:

$$F_{n,s,\theta} + F_{protocol} + F_{cloud} \leq \left(T_{cpu} - C\right) \times p \tag{4}$$

Exploring network latency in detail, for each link we have transmission delay D_i and queuing delay Q_i. Therefore, the total network latency will be the sum of the latencies for each link between client and service. Hence, we can express $F_{n,s,\theta}$ as:

$$F_{n,s,\theta} = \sum_{i=1}^{n}(D_i + Q_i) \quad (5)$$

If we denote the transport block size as b, then the time to transmit p blocks over a link is equal to the number of blocks multiplied by the block size and divided by the bandwidth of the link. Thus, the transmission delay for p blocks over link i is $\frac{p\times b}{S_i}$, where S_i is the bandwidth of the link; hence:

$$F_{n,s,\theta} = \sum_{i=1}^{n}\left(\frac{p\times b}{S_i} + Q_i\right) \quad (6)$$

So using Equation 6, we can expand Equation 4 as follows:

$$F_{cloud} + F_{protocol} + \sum_{i=1}^{n}\left(\frac{p\times b}{S_i} + Q_i\right) \le \left(T_{cpu} - C\right) \times p \quad (7)$$

On a lightly loaded system, we consider $F_{protocol}, F_{cloud}$ and Q_i to be negligible. A simplified version of the above equation for this scenario becomes:

$$\sum_{i=1}^{n}\left(\frac{b}{S_i}\right) \le T_{cpu} - C \quad (8)$$

This equation shows that as mobile users move away from a service and more links are added between them, then the QoS can deteriorate and if it exceeds the threshold $T_{cpu} - C$ for video, this will result in a degradation of QoE.

Therefore, $H_L = T_{cpu} - C$ represents a QoS hard limit which must not be crossed in order to avoid jitter. To avoid reaching this hard limit, we introduce the concept of a soft limit which acts as a trigger for our migration mechanism.

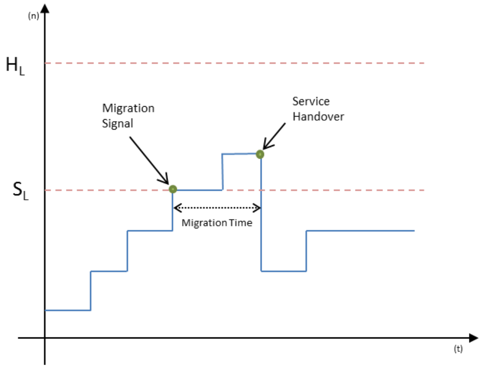

Fig. 2. QoS degradation diagram

Let S_L be the soft limit that we are aiming for in order to prevent jitter and M_t is the migration time. So the difference between the hard limit and the soft limit is:

$$H_L - S_L = a_l \times M_t \quad (9)$$

where a_l is the rate of network latency increase as the number of network links increases. We can calculate a_l at the mobile device and we can also find M_t between two Clouds. H_L is given by the mobile device, so we can calculate S_L to find where to set out QoS trigger for service migration.

In Fig. 2 we visualised how the increasing number of links between a user and a service can bring the connection near the QoS limit and how we can use a soft limit to trigger service migration in order to prevent this. We can also see that for a given migration time, we need to adjust $S_L$so that during the migration the QoS will not reach the H_L.

V. Envisioned Cloud-Based Service Framework

The framework we are investigating at Middlesex University is service-centric with focus on maintaining QoS by means of moving instances of services across Cloud boundaries. Different approaches are being investigated in terms of mechanisms for this framework. To facilitate a service-populating model we are introducing the idea of an Open Cloud. Unlike existing Cloud implementations where the Cloud is private and only runs services controlled by its owner, an Open Cloud allows services from third party providers to populate it. It is important to note however, that Cloud providers still have administrative control over the Cloud. To differentiate from the existing “closed” Cloud model, we can think of “open” Cloud as a “Resource Pool” in order to emphasize the fact that anyone can use these resources to run their services and in fact anyone can provide such a resource pool and accept services from other providers to run on it hence the need for a new service framework.

Fig. 3 shows the layers of the architecture and how they relate to the OSI model. The proposed framework and the OSI model share the same level of abstraction in terms of network technologies and protocols and this makes it easy to use the OSI as a reference to our model as opposed to using the TCP/IP model. The service architecture is not meant to map directly to some of the OSI layers. Some of the functions performed in the proposed layers can interact with OSI layers to perform network-level operations while other layers do not present any functions that directly interface with the OSI and are therefore considered extra layers. Finally, to better understand what each layer does, we will relate it to the previous example.

The Service Management Layer (SML) deals with how services are registered in a Cloud. This also includes the overall Service and Security Level Agreement (SSLA) between the Cloud providers and the service providers and the unique Service ID. In this layer, billing information between resources and services providers is also processed. The SML can be considered as part of the Application Layer in the OSI since it defines the applications themselves and how they use resources.

In our example, when a service provider wishes to publish a service, they have to define security and QoS parameters. In essence what they define is the requirements to run the service to a level that the provider considers adequate. To achieve this, each service must have a list of parameters which must agree with the parameters offered by the Cloud. This list is also used in the migration negotiation to find appropriate

Clouds that can accept the service. However, the SSLA is not rigid and if a service requires more resources, it can be given extra and the service provider will be billed accordingly. So the SSLA acts in a way as a minimum requirement set by the service provider. Upon defining the SSLA, the service is given a unique ID by which clients are able to make requests. The SSLA contains the primary parameters considered when a service migration is negotiated between Clouds. If the target Cloud fails to meet these parameters, then the migration is aborted and a more suitable target is selected.

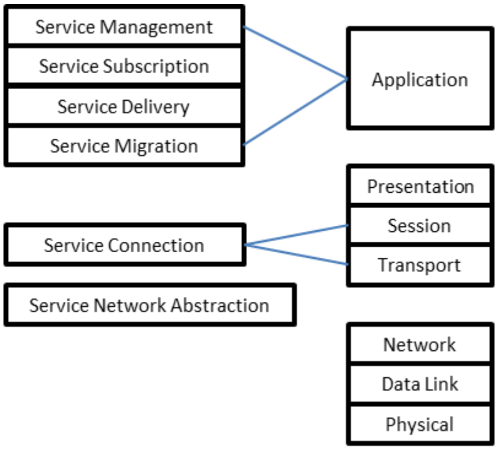

Fig. 3. Envisioned cloud-based service framework

The Service Subscription Layer (SSL) deals with the subscription of clients to the service and holds information that handles the subscriptions such as User IDs, the list of services subscribed to by individual client and the associated client SLAs between clients and services. It should be noted here that billing information between clients and services should be processed at this layer. This should not be confused with the billing information between service providers and Cloud owners described in the SML. The SLA at this level gives us the ability to provide different service terms to each client. This layer can give instructions to the Presentation Layer in the OSI in order to handle user specific service parameters such as encryption or CODECs in video streams.

In our example when a client requests a service, it is treated as a subscriber to that service. This term does not necessarily mean a long-term membership or imply that the client is billed for the subscription. It can be used merely as a record keeping function in order to keep track of how many clients at a given point are accessing a service and from where.

The Service Delivery Layer (SDL) is responsible for the delivery of services to individual clients. The layers below receive instructions from this layer with regard to connecting to individual clients as well as populating Clouds.

In our example, the logic that processes all the data regarding QoS characteristics and user mobility resides in this layer. It uses data from the overall SSLA and the client SLA and checks if the requirements are met by using network QoS data given by the layer below. Such data can be fed to this layer by the mobile devices themselves either in the form of a process running separately or through a QoS-aware protocol that can report latency and bandwidth between two end points. If it finds that SLA and SSLA requirements are not met, it marks the service as “ready to migrate” and seeks out a target Cloud that can meet the agreement requirements. To find such Clouds the first condition is to minimise the distance between the service and the location of the client. The Clouds that cover this requirement are given the SSLA list of the service. The Cloud that fulfils all the parameters in the SSLA list and can provide better QoS than the others can then proceed to the Migration process in the layer below. Fig. 4 shows the process of service migration in a scenario including client mobility.

The Service Migration Layer (SMiL) is responsible for the Migration of services between Clouds. It deals with resource allocation across Clouds to facilitate service population. It also holds the mechanism that performs the handover of client connections between services.

In order to make a service populate a Cloud we must first make sure that the target Cloud can accept the service. We assume that Clouds are able to report whether or not they can meet client SLA and overall SSLA requirements in their present state and based on that, a decision is made at SDL on whether or not to move a service. It is now up to the SMiL to instruct the Cloud with regard to which resources need to be allocated to the service. It therefore acts as a handshake mechanism between the service and the Cloud. In moving services, resources are allocated and a service handoff is performed between the new and previous Clouds. Once the service has moved, this layer is also responsible for initiating a network level handover for the subscribed clients.

The Service Connection Layer (SCL) monitors connections between clients and services. It is up to this layer to handle issues such as client mobility and inform the upper layers of changes in connection status which in turn might trigger service migrations. This is done by gathering QoS data from the network and from client devices. Some of this layer’s functions map directly to the Session Layer in the OSI model. For instance, this layer monitors active sessions by gathering QoS data from the transport layer.

In our example, this layer is where we gather data about the network status and the location and mobility characteristics of the users. Any QoS events recorded in this layer are pushed up to the SDL in order to evaluate the conditions and decide if a service needs to move. Events can be anything from a change of bandwidth and latency, to complete change of network technology such as going from WiFi to GSM. Such changes can be detected by the transport protocol itself if it has a QoS tracking mechanism or by a separate service running on mobile devices and recording network metrics. The SCL is also responsible for the network handover between clients and services after a service moves. This information is given to this layer by the above layers and is then passed on to the clients in order to initiate connections to the new Cloud where another instance of the service is running.

Finally, the **Service Network Abstraction Layer (SNAL)** makes the network technology transparent to the upper layers in order to simplify and unify the process of migration. The function of this layer is to act as a common interface between the service delivery framework and the underlying network architecture such as IP overlay network or new technologies such as Y-Comm [16] which divide the Internet into a Core network surrounded by Peripheral wireless networks.

The SSLA is regarded as a list of QoS and resource requirements for the service. The concept of the SSLA is not business-minded in the sense that two providers sign a contract that will then enable services to populate Clouds by a specific provider. If any Cloud can fulfil SSLA requirements, then a service can populate it. Within the Cloud we perform **SMiL** functions in order to handle service migrations.

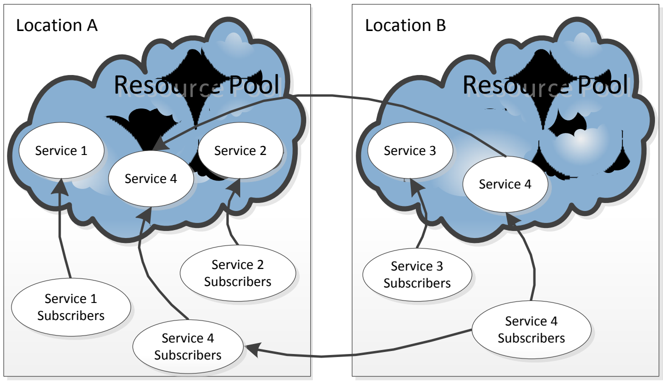

Fig.4. Service migration example

VI. Implementation Mechanisms

In order to gather QoS data and know the network conditions in a specific area, we are using another mechanism that we call the QoS Monitor. It is considered to be part of the **SCL** and acquires such data by querying the clients for network conditions.

At this point we are assuming a mechanism that can resolve human-friendly service names to unique Service IDs. For service delivery purposes in the **SDL** we need mechanisms that will connect service subscribers to the correct instance of a service. Service Tracking and Resolution or STAR keeps a record of Service IDs and in which Clouds their instances are running and also uses input by the QoS Tracking. Using this information, STAR will make a decision on which Cloud is better suited to service a client request based on the location of the client. To achieve this functionality, STAR can look up routing tables in order to identify which Cloud is closer to a user. A choice is always given to a service to reject the new client and forward them to another Cloud if possible. This gives control to service providers and also becomes a contingency mechanism in case STAR makes a wrong decision. The STAR server can be scaled similarly to the DNS system since it is essentially the same type of service albeit with some extra parameters. Once a Cloud ID is found, then the ID is resolved into the IP addresses of the Cloud controllers that the client can contact to access the service. The process is shown in the Fig. 5. It should be noted that alternatively the Cloud ID can be returned to the client, at which point, the client will have a choice of which DNS to use to find the IP addresses.

Finally, Fig. 6 illustrates a simplified global infrastructure for user mobility and service population. Global Service Population Authority (GSPA) also performs **SDL** functions and makes decisions on when to populate a Cloud based on all the factors given by the aforementioned mechanisms. The instruction to move a service will be given after the target Cloud has agreed with the SSLA of the service at which point the next function of GSPA is to update STAR records with new instances of services. We should note at this point that the GSPA can also be implemented as part of each Cloud so that each Cloud will manage QoS statistics for its own clients. Using this method we can leave it up to individual Clouds to negotiate service migrations instead of receiving instructions from a global mechanism. This allows for a more self-managed design but lacks the central management capability of the GSPA.

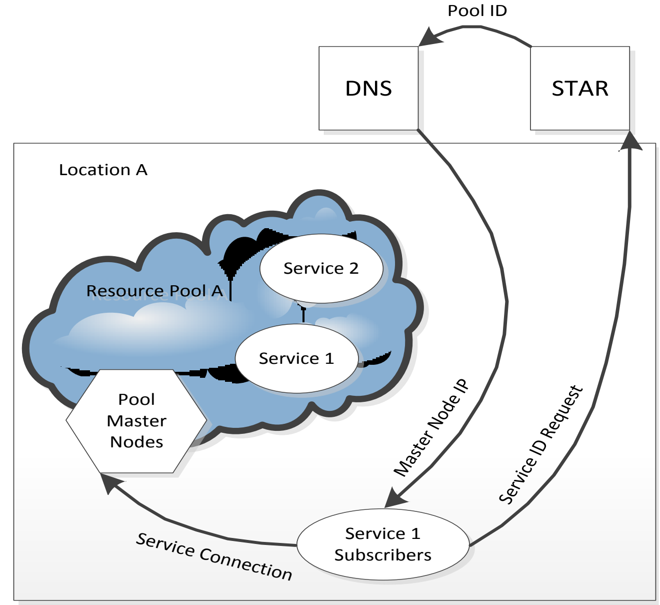

Fig.5.Service request and resolution

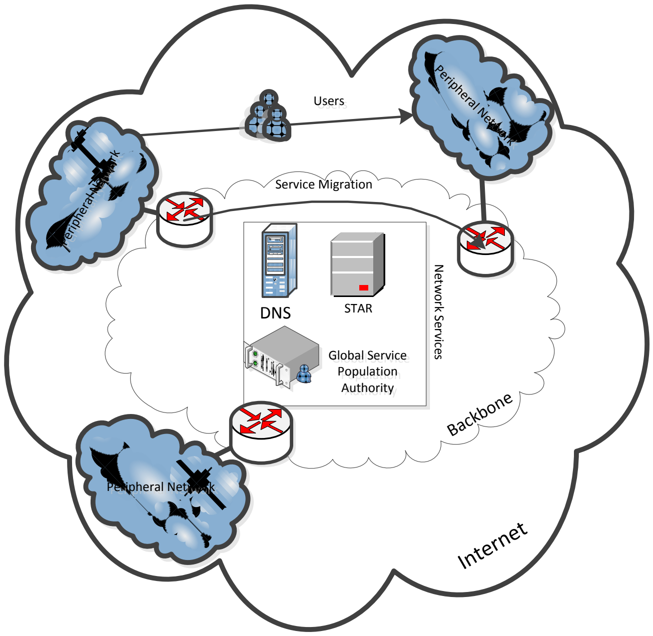

Fig. 6. Global service population infrastructure

Fig. 7 shows a handshake diagram on how a client requests a service and how all the layers work to deliver it. The first step is for the client to request a service ID from STAR. This service request includes the location of the client as well as the level of QoS required. STAR will then forward the client to a Cloud ID that hosts the requested service can honour the QoS level. While the connection is active, the client sends QoS metrics to the GSPA. If the GSPA detects that a QoS drops below a threshold, it will signal the Cloud to perform a service migration. When the service migration is performed successfully, the Cloud will also register the new instance of the service to the STAR.

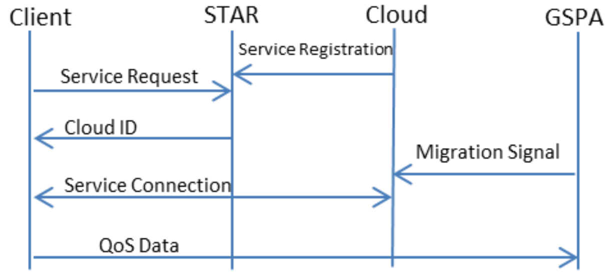

Fig.7.Service delivery handshake diagram

We have identified however, that moving a service can cause a large overhead on the network. The amount of traffic generated by the migration of a service depends on the size of the service itself and the user files it needs to copy. This means that aside from QoS criteria, any services that migrate gratuitously for unnecessary or minimal QoS gains can cause excessive congestion. A potential solution can be to prevent a service that migrated recently, from migrating again in a short time period. Such behaviour would congest a network with more traffic than letting clients connect over a large distance. This is currently an open issue in our research.

VII. Applications

A QoS aware service-populating model can bring many advantages to numerous types of services and applications. In terms of content delivery, migrating web services for example can reduce network congestion on a global scale for websites that are very frequently accessed or that have a lot of multimedia content. This position is further solidified by the trend of High-Definition media that consume a lot of bandwidth and in streaming scenarios, requiring consistent and high QoS. Furthermore, this type of service often has active content which is not possible to cache regionally, so moving the entire service closer to a geographical region is going to be of great benefit if there is high demand for that service in the area. Another benefit to web services using this framework is that load balancing becomes easier to manage. Services can be replicated or removed based on demand and this provides a highly adaptable resource allocation scheme.

From a computational perspective, Cloud providers can share their resources with other providers. This gives them the flexibility to request additional resource when their Cloud needs them or rent some of their resources to other providers that need them. By taking into account multimedia creation services such as rendering, we can see how such a scenario is applicable and how it can benefit clients and providers alike. Furthermore, if we combine the above scenario with mobile devices, we can see how in the future we may find ourselves in a position where rendering is done on the Cloud and the mobile devices only display the content. This can occur in applications such as games. In these situations, the proposed framework will not only balance the rendering load on Clouds but will also relieve networks from the high traffic generated by streaming video and audio. The distance reduction between clients and services caused by migrations will also decrease the latency and give users a more interactive feel to their multimedia application, thus improving the QoE.

VIII. Conclusion and Future Work

In this paper, we have outlined the challenges presented by user mobility in future networks. Current models of service delivery are inefficient and will not scale to cover the future needs of mobile users. We believe that the combination of Cloud technology and the proposed service delivery framework can bring a better solution to the efficient management of network resources while providing a high QoE for the clients.

To further develop our framework we are currently working on a method that calculates the rate of increase of latency as a user moves while streaming a video. We are also investigating how the number of clients can influence the decision making at the Service Delivery layer.We recognize that there is much to do and welcome feedback on this paper.

Acknowledgment

The authors would like to thank the guest editors and anonymous reviewers for their valuable comments.

References

[1] Apple, 2012. iCloud. [online] Available at http://www.apple.com/icloud/ [Accessed: 15 February, 2012].

[2] Postel J., Reynolds J., and ISI, RFC 948. A Standard for the Transmission of IP Datagrams over IEEE 802 Networks. IETF, February 1988.

[3] ETSI, 2011. Mobile Technologies GSM. [online] Available at http://www.etsi.org/WebSite/Technologies/gsm.aspx [Accessed: 15 February, 2012]

[4] Inamura, H., Montengero, G., Ludwig, R., Gurtov, A. and Khafizov, F., RFC 3481. TCP over Second (2.5G) and Third (3G) Generation Wireless Networks. IETF, February 2003.

[5] Motorola, 2012. Long Term Evolution (LTE):A Technical Overview. [online] Available at

[6] http://www.motorola.com/web/Business/Solutions/Industry%20Solutions/Service%20Providers/Wireless%20Operators/LTE/_Document/Static%20Files/6834_MotDoc_New.pdf [Accessed: 15 February, 2012]

[7] Amazon, 2012. EC2 [online] Available at http://aws.amazon.com/ec2/ [Accessed: 28 February, 2012].

[8] Microsoft, 2011. Cloud Computing [online] Available at http://www.microsoft.com/en-us/cloud/default.aspx?fbid=uzSXFYwh_d4 [Accessed: 28 February, 2012].

[9] Sonnek, J., Greensky, J., Reutiman, R. and Chandra, A. 2009. Starling: Minimizing Communication Overhead in Virtualized Computing Platforms Using Decentralized Affinity-Aware Migration. In Proceedings of the 39th International Conference on Parallel Processing (ICPP'10), San Diego, CA, September 2010.

[10] Sonnek, J., Chandra, A. 2009. Virtual Putty: Reshaping the Physical Footprint of Virtual Machines. In the Workshop on Hot Topics in Cloud Computing (HotCloud'09), San Diego, CA, June 2009.

[11] C. A. Waldspurger. Memory resource management in VMWare ESX server. In Proceedings of OSDI, 2002.

[12] T. Wood, G. Tarasuk-Levin, P.Shenoy, P. Desnoyers, E. Cecchet, and M. Corner. Memory Buddies: Exploiting Page Sharing for Smart Colocation in Virtualized Data Centers. In Proceedings of the 5th ACM Intl. Conference on Virtual Execution environments, 2009.

[13] D. Gupta, S. Lee, M. Vrable, S. Savage, A. C. Snoeren, G. Varghese, G. M. Voelker, and A. Vahdat. Difference Engine: Harnessing Memory Redundancy in Virtual Machines. In Proceedings of OSDI, 2008.

[14] Wenwu Zhu, Chong Luo, Jianfeng Wang, and Shipeng Li. 2011. Multimedia Cloud Computing [An emerging technology for providing multimedia services and applications]. IEEE Signal Processing Magazine, May 2011.

[15] Brisko, T. RFC 1794. DNS Support for Load Balancing. IETF, April 1995.

[16] Thakker,D.N. Prefetching and clustering techniques for network based storage, School of Engineering and Information Sciences, Middlesex University, PhD thesis, May 2010.

[17] Middlesex University, 2011. Y-Comm Research [online] Available at http://www.mdx.ac.uk/research/areas/software/ycomm_research.aspx [Accessed, 2 March, 2012].

Fragkiskos Sardis received his Bachelor (First Class Honours) from Middlesex University in 2008. He received a scholarship for a Master degree in Computer Networks at Middlesex University which he completed in 2009 with distinction. In 2010 he received a scholarship by Middlesex University for a PhD in the area of Cloud computing and mobile networks. Throughout his studies, he has worked as an I.T. administrator, network architect and I.T. consultant. His other areas of interest include, network security, wireless communications and distributed computing.

Glenford Mapp received his BSc (First Class Honours) from the University of the West Indies in 1982, a MEng (Distinction in Thesis) from Carleton University in Ottawa in 1985 and a PhD from the Computer Laboratory, University of Cambridge in 1992. He then worked for AT&T Cambridge Laboratories for ten years before joining Middlesex University as a Principal Lecturer. His primary expertise is in the development of new technologies for mobile and distributed systems. Glenford does research on Y-Comm, an architecture for future mobile communications systems. He also works on service platforms, cloud computing, network addressing and transport protocols for local environments. He is currently focusing on the development of fast, portable services that can migrate or replicate to support mobile users.

Jonathan Loo received an MSc degree in Electronics (with Distinction) and a PhD degree in Electronics and Communications from University of Hertfordshire, UK in 1998 and 2003, respectively. Between August 2003 and May 2010, he was a Lecturer in Multimedia Communications at the School of Engineering and Design, Brunel University, UK. During this period, he was also the Course Director for MSc in Digital Signal Processing. He is currently a Reader in Multimedia Communications at the School of Engineering and Information Sciences, Middlesex University, UK. His research interests are in the area of multimedia communications, which include visual media processing, video coding and transmission, wireless communications, digital signal processing, embedded systems and wireless network, protocols and security.

Mahdi Aiash received his Master and PhD degrees from Middlesex University, London, UK. Dr. Aiash is involved in many research efforts such as the Y-Comm research group and the Wireless Sensors group. His main interest is in the areas of network and information security, ubiquitous and pervasive communication. He is an Associate member of the IEEE and IEEE ComSoc since 2007 and a Programme Committee in many conferences and Journals.

Alexey Vinel (M’07–SM’12) received the Bachelor (Hons.) and Master (Hons.) degrees in information systems from Saint Petersburg State University of Aerospace Instrumentation, St. Petersburg, Russia, in 2003 and 2005, respectively, and the Ph.D. (candidate of science) degree in technical sciences from the Institute for Information Transmission Problems, Russian Academy of Sciences, Moscow, Russia, in 2007. He is currently a Researcher with the Department of Communications Engineering, Tampere University of Technology, Finland and a part-time Guest Professor at Halmstad University, Sweden. He has been an associate editor for IEEE Communications Letters since 2012. His research interests include multiple-access protocols and intelligent transportation systems.