Dynamic edge-caching for mobile users: minimising inter-AS traffic by moving cloud services and VMs

Dynamic Edge-Caching for Mobile Users

Minimising inter-AS traffic by moving Cloud services and VMs

Fragkiskos Sardis
SaT School
Middlesex University
London, UK
fs254@live.mdx.ac.uk

Glenford Mapp
SaT School
Middlesex University
London, UK
g.mapp@mdx.ac.uk

Jonathan Loo
Sat School
Middlesex University
London, UK
j.loo@mdx.ac.uk

Mahdi Aiash
SaT School
Middlesex University
London, UK
m.aiash@mdx.ac.uk

***Abstract*— In recent years, Cloud technology has revolutionized the way services are delivered to end-users. The advent of truly mobile computing in the form of smartphones and tablets has also driven the demand for Cloud resources in order to compensate for the inherent lack of local resources on these devices. Furthermore, modern mobile devices are equipped with multiple network interfaces and in combination with the rapid deployment of wireless networks, it is expected that they will always have Internet connectivity and access to Cloud resources. In this paper we will focus on traffic management for interactive multimedia services accessed by a mobile user by means of dynamic migration of a Virtual Machine. Network performance measurements are taken from a network of virtualization-enabled hosts that perform live migrations of a Virtual Machine which hosts multimedia content. The data is used as input to an equation that determines whether a migration would be beneficial in terms of traffic localization based on a user's mobility characteristics and network usage patterns. The contribution of this paper lies in the proposed mechanism of managing traffic for interactive services in the context of mobile cloud computing. This helps alleviate the increased network costs introduced by dynamic migrations driven by Quality of Service parameters and may result in increased network traffic for the benefit of improved QoS.**

Keywords- virtualisation, live migration, cloud gaming, traffic management, mobile cloud

I. INTRODUCTION

Cloud computing has facilitated the offering of new types of services such as Video-on-Demand (VoD) and Cloud Gaming and as a result we now see examples of multimedia content offered from Clouds such as video streaming from YouTube and Netflix and video gaming such as Onlive and Coreonline. However, multimedia is particularly sensitive to Quality of Service (QoS) conditions and a good network connection is always a requirement when it comes to accessing such content and especially in High Definition (HD) formats. With Cloud gaming, QoS is even more important since there is a level of interaction involved in the process of gaming. Unlike VoD, gaming cannot afford to buffer video frames since the content on display is heavily influenced by user interaction which is unpredictable. As a result, we must always have a good connection between the client and the Cloud in order to provide a smooth gaming experience.

Content caching is one way of improving the QoS while decreasing the amount of traffic generated by services. Localised copies of content can be distributed on the edge routers of autonomous networks in order to better serve their clients. This approach works well when we are not dealing with interactive content. Photos, videos and some non-interactive services can benefit from edge-caching but when dealing with interactive services, it becomes apparent that it is not possible to cache any part of their content. For example, a user's virtual machine (VM) that is used for playing games on the Cloud or otherwise interacting with multimedia applications is not cacheable content. In fact, it may not even be desirable to cache such content on multiple datacentres since we are talking about highly personalized data that is only used by a single person or by a small group of users. Regardless, such applications are still QoS sensitive and they can generate a lot of traffic due to their interactive nature and multimedia content. This means that a new caching solution may be necessary as the Internet moves towards Cloud-based personalized interactive services.

The above problem becomes more severe when we consider a scenario of mobile users. We often find mobile users changing networks by connecting to various Wi-Fi hotspots, even when they are not physically mobile. Hence, they are not bound to a specific AS that can be used as a caching target for personalised content. The importance of this observation becomes more apparent if we consider that mobile devices inherently lack local resources and often rely on Cloud services for storage and other functions. More importantly, one of the selling points of Cloud gaming is that it makes it possible to play games on any platform, including mobile phones [1].

The content and results presented in this paper are a continuation of the research presented in [2,3]. In the following sections we evaluate in terms of traffic management, how desirable it may be to dynamically move a VM based on a mobile user's interaction with it. We will look at how much data is being sent to the client when viewing a video and when playing a game and we will also examine how much traffic is generated by moving that VM to a new location. The measurements are used to test an algorithm that estimates if a VM migration can bring traffic savings based on the mobility of a user. The rest of the paper

is structured as follows: In Section II, we will have a look at some background information in the fields of edge-caching, mobile devices, Cloud technology and wireless connectivity. In Section III we will present the methodology of the experiment and in Section IV we will present and analyse the results. Finally, Section V contains the planned future work and conclusions.

II. BACKGROUND

A. Edge-caching and CDN

We define as a Content Delivery Network (CDN) a group of servers and datacentres that cache content by a single or multiple providers with the purpose of redistributing it to clients. The aim of a CDN is to improve the QoS and availability of the content that is being cached. The CDN also has the beneficial side-effect of offloading traffic from the publisher's servers and therefore relieving them of traffic costs. On this basis, edge-caching works on the principle that a CDN within one particular AS will be able to offer content at a better QoS and with lower traffic costs than a CDN or provider outside the boundaries of the AS [4].

B. Media-edge Cloud

Although Clouds can allocate resources to tasks in an elastic and hence quite efficient manner, when it comes to QoS sensitive applications they face some challenges. A Cloud is essentially a group of machines that communicate and share resources over a network which has a hierarchical structure. Like in any other network, the deeper into the hierarchy a server resides, the longer it takes for the data to reach the edge of the network. The Media-Edge Cloud (MEC) [5] framework presents a method of optimizing the location of QoS sensitive applications within a Cloud by placing them closer to the edge of the infrastructure and therefore closer to the user. The argument made is that these applications generate a lot of traffic and placing them in the deeper layers of the network only causes further congestion within the Cloud. Furthermore, by placing these services near the Cloud's edge, the QoS can be improved since the traffic has to go through fewer network interfaces before finally reaching the client.

Essentially, this concept is about expanding the efficiency of the Cloud to include optimizations in traffic management (and by extension to QoS) within the Cloud.

C. Cloud Interoperability

Cloud Interoperability is now a popular topic within the industry. In 2012, IEEE announced two Cloud interoperability groups tasked with providing a draft standard for workload portability and Intercloud Interoperability [6,7]. Workload portability is essentially a set of standards for developing file formats, APIs and management interfaces that will allow platform-agnostic tasks to be developed for Clouds and essentially guarantee that a specific task or software developed for one Cloud can run on another even if it is using a different Cloud platform. Intercloud Interoperability is the term used for defining the ability of a Cloud platform to share resources and co-operate with a different Cloud platform. We believe that the development of these technologies will provide us with an improved solution to edge-caching where entire services and virtual environments can be localized depending on QoS conditions and traffic demands.

D. Mobility and QoS

The Y-Comm framework [8] is aimed at providing a solution for heterogeneous networking in mobile computing. The fundamental approach is to divide the Internet into Core and Peripheral networks. The Core network is the Internet's backbone where we find very fast connections offering low latency and high bandwidth. In the peripheral networks we find slower network connections of different technologies such as Wi-Fi, 3G, LTE and DSL. In these networks we have less bandwidth and higher latency depending on the network technology used. Y-Comm uses mechanisms in the core and peripheral networks to assist devices in performing seamless vertical handovers such as between Wi-Fi and LTE. This means that as a user moves, the mobile node can "jump" from one network to another, always seeking the best possible QoS or the lower cost connection. Consequently, a mobile node can be constantly switching between different network providers since the entire network is constructed around the concept of an open approach instead of being locked to a particular mobile network provider like it is at present.

One of the side effects of this technology is that mobile clients may receive fluctuating QoS depending on which networks their device is connecting to. This is because different providers have different network equipment and structures. These factors can greatly affect a connection and they are impossible to track in a way that we can use in order to predict what may happen to individual connections and data streams when a vertical handover occurs. The important thing however is that this type of network mobility can lead to problems with edge-caching, especially when dealing with high traffic applications that are also QoS sensitive.

One argument against Y-Comm is that Mobile Operators (MO) will never allow their clients to roam freely to other operators; however, current research is showing demand for a solution that offloads 3G and LTE traffic to Wi-Fi for better management of the limited resources of mobile networks [9,10]. Therefore, we envision a future scenario of a mobile user roaming from his mobile network operator to different Wi-Fi hotspots and causing traffic to be generated at different networks along his path. So in this scenario, we go back to our previous conclusion that it would be almost impossible to provide edge-caching for this type of use model and at the same time the client could experience varying QoS conditions. It should also be noted that we are not considering a user accessing static content such as video streaming but rather a user accessing interactive services such as Cloud gaming that are personalized and therefore cannot be cached in a generic manner.

In this paper, we shall be using the concept of Network Dwell Time [11] (NDT) which is an estimate of how long we are expecting a mobile user to stay connected to an AS

before moving past its range and connecting to another network. If we know how much traffic a user is generating, we can calculate the load on the network for the duration of NDT and we can then adjust the caching dynamically to compensate for it. In section III, we shall present how the NDT is used to achieve this.

E. Cloud-based Service Delivery Models

In [12] Hoßfeld et al. argue that the future of Cloud services is going to be centred on the Quality of Experience as perceived by the user. They go on to argue that moving services to the Cloud does not provide a solution to QoS problems and what is really needed is the classification of content that is being served to clients and the adjustment of QoS accordingly, in order to maximize the Quality of Experience (QoE) of users. To achieve this they use an example of interactive multimedia services and they make a case that network conditions can severely affect the QoE and they are outside the control of the client and the Cloud provider. As a result, what is proposed is a form of edge-caching that takes QoE and QoS parameters at its input and accordingly locates services and content to datacentres that are most capable of serving them according to the user's needs from a network perspective. One added benefit of this approach is that by moving a type of service or content closer to the user (much like in edge-caching) a lot of traffic is kept within an AS as opposed to having long connections over several networks.

However, if we are to enable the migration of entire Cloud applications and services across Clouds then we need a new service delivery framework that monitors QoS fluctuations between services and clients and negotiates the best location for services. Cloud interoperability mechanisms will also ensure that heterogeneous Clouds are able to accept each other's services and hence, any Cloud within an AS will be able to host any service. One such framework was proposed in [2.3], where a simple example of a mobile user is given as a means of explaining how a mobile user's perceived QoE can be monitored by the mobile device and reported to service delivery mechanisms that will instruct Clouds to negotiate resources with each other and move services to locations that are more capable of providing a good QoE to the user. However, as we see in [13], socio-economic aspects are equally as important and as a result, one important parameter to providing a good QoE is the cost to the entities involved (client, service provider, and network). In the following sections we shall attempt to answer two questions: The first question is, how much traffic is generated by a user accessing a VM remotely for the purposes of watching a movie and playing a game? The second question is, if we decide to move that VM, how much extra traffic are we generating and is that amount of traffic compensable in any way in a mobility scenario? In the following section we shall present the test platform and testing methodology.

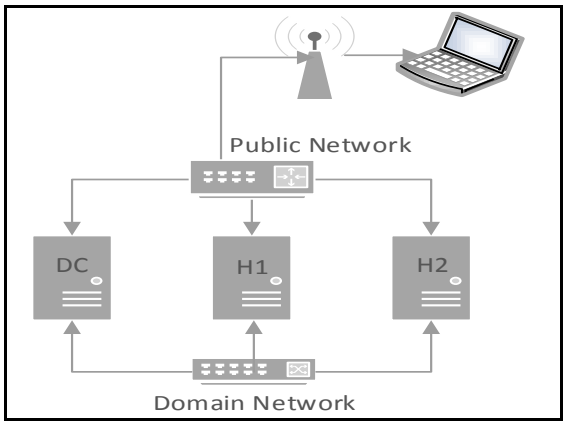

Figure 1 Physical network diagram

III. METHODOLOGY

A. Experimentation Platform

The main data we want to gather from this experiment is the amount of traffic generated by the VM migrations and also the user traffic when accessing the VM via remote desktop and playing games or videos. To achieve this we used three physical hosts running Windows 2012 Server in a domain configuration. One of the hosts played the role of Domain Controller and DNS while the other two hosts played the role of Hyper-V platforms. Although part of the same domain and connected to the same LAN, for the purpose of this experiment we can consider the two Hyper-V hosts as being two separate "datacentres" that host virtualisation services. We will be moving a VM between these two hosts and measuring how much traffic is being generated by the live migration. At this point, we are not focusing on the performance of the hardware and how long it takes to do the migration. The main interest is to identify the exact amount of data being transferred so that we can understand if there are any overheads that we need to take into account.

The domain network is running over gigabit Ethernet (GbE) and it is the subnet used for moving the VM between the two hosts. In essence, this forms our domain's backbone network. The public network used by our client to access the VM is running over Fast Ethernet which also has a 802.11b/g wireless access point. The client connects either by Ethernet or Wi-Fi to the network and accesses the VM via Remote Desktop Connection (RDC). From this connection we are monitoring how much traffic is generated by the RDC when watching a video or playing a game. The VM is running Windows 8.1 Professional and the clients connect with RDP v.8.1. To carry out the experiments we copied a video file to the VM and we also installed Pinball FX2. The size of the VM's hard disk was 15.6GB. The VM was also configured to have 1GB RAM and 2 virtual CPUs. In Fig. 1, we visualize the network layout.

B. Test Plan

The first phase of the experiment was to initiate a complete move to Host 2 (H2) while the VM was running. Then we repeated the move back to H1. We gathered data in

terms of traffic generated for the complete move of the VM. We will be referring to this as a "full" migration and it represents a case where an entire service or VM has to be transported in real-time to another physical host or datacentre that is not initially "aware" it in any way (different domain/ownership). The VM is transported seamlessly and connectivity to the client is not affected. The process was repeated multiple times and the average transfer rate was recorded.

In the second phase of the experiment we moved only the Virtual Hard Disk (VHD) to H2 and monitored the amount of traffic generated. In this phase we also attempted to move the VM between hosts without actually moving its storage from H2. We recorded data on the traffic generated by moving only the VM. We will be referring to this as a "light migration". Once again, the migration is seamless with minimal or no impact in connectivity. A diagram of the two types of migration is presented in Fig. 2.

The last phase of the experiment had two usage scenarios. In both cases, H2 was hosting the VHD and H1 was hosting the VM. In the first case, we monitor the amount of traffic between the VHD and the VM while the user is watching a video file stored on the disk. At the same time we are recording the traffic between the user and the VM (RDC traffic). In the second case, we repeat the experiment but the usage scenario now becomes a video game (Pinball FX2). The resolution we selected for the remote desktop session was Full HD (1920x1080) on the basis of collecting data for what is currently considered a standard in entertainment.

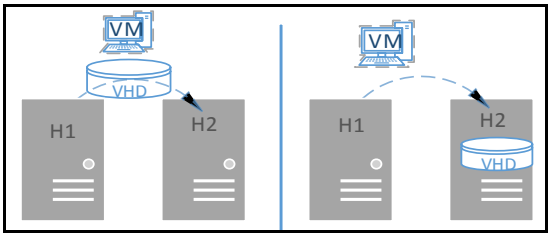

Figure 2 Migration types

C. Equations

After gathering performance data, we constructed some equations that help us identify when and if moving a VM is desirable based on the client's mobility. As mentioned in the previous section, the size of a full migration is given by:

$$C_{VM} = VHD + VM_{RAM} \quad (1)$$

The time to move a VM over a dedicated backbone network with throughput ϑ_b is given by:

$$t_m = \frac{C_{VM}}{\theta_b} \quad (2)$$

We will now start considering user mobility. We introduce NDT as the time a user is estimated to remain connected to the network. Based on the above, for a deterministic NDT, we have:

$$t_{sw} = t_{NDT} - t_m \quad (3)$$

Where t_{sw} (*"savings window"*) is the time duration after a migration that RDC traffic will be contained within the AS. So to carry out a VM migration and reduce inter-AS traffic we need to fulfil:

$$\theta_r \times t_{NDT} > \theta_b \times t_m + \theta_r \times t_m + \theta_d \times t_{sw} \quad (4)$$

Where θ_r is the throughput of the RDC to the VM and θ_d is the throughput between VM and VHD if a light migration is performed. We are comparing the theoretical amount of data that would have crossed the edge of the AS without a migration to the size of the VM being migrated, plus the amount of data that will cross the AS edge while the migration is being carried out and any data that may flow between VM and VHD if the VHD is not moved along with the VM. If the second part of the equation is smaller than the first, then a migration has the potential of reducing the amount of Inter-AS traffic for a specific user. For a full migration scenario $\theta_d = 0$ since the traffic will not cross AS boundaries. For a light migration we consider $C_{VM} = VM_{RAM}$

When (4) is true, moving a VM inside an AS will reduce the amount of traffic passing through the edge of that AS and therefore improve the performance on edge routers and reduce the traffic costs for that AS. We will use data gathered by the experiments as inputs to (4), in order to find out, how the changes in parameters can affect the decision to move a VM.

IV. Experiment Results & Analysis

While experimenting with moving a VM, we realised that the GbE network was not saturated by the traffic. In fact, the bottleneck was at the performance of our storage system. This was confirmed by using different types of storage such as solid state disks and hard disks. We found that migration times varied depending on the read/write capabilities of the storage device in each host.

A. Live Migration Results

The two VM hosts were equipped with different SSDs and therefore their performance was different depending on if a host was receiving or sending and hence reading or writing to the SSD. Table (1) shows the network performance measurements taken during full migrations for H1. For H2, the performance numbers were the same but the traffic direction was reversed.

TABLE I. FULL MIGRATION THROUGHPUT OVER GBE

Traffic Direction	Highest	Lowest	Average
Sending (MB/s)	45	41	43
Receiving (MB/s)	64	57	60.5

Because the bulk of a full migration is the size of the VHD with the size of the VM's RAM being very small comparatively, we see that the speed of a full migration is mostly dependent on the performance of the storage system. In Fig. 3, we can see that while the VHD was being copied

over the network, the entire bandwidth was not used. However, during the context transfer (at the end) which is purely a RAM copy operation, we see a spike in the bandwidth utilisation. Therefore, it is very important to ensure maximum performance of the storage system in a Cloud in order to make dynamic live migrations a viable solution to managing traffic. The size of the VM is also very important and we can see that the smaller the size of the VHD and vRAM, the sooner the process will finish.

Before performing a "light" live migration, we decided to move only the VHD while the VM was active and identify exactly how much traffic this would generate and what would be the network access pattern during the process. Our observation during a storage migration is that it looks like a full migration but lacks the context transfer at the end. What we see instead is an operation that closely resembles a file copy in terms of network and disk utilization. We find the throughput to be the same as when performing a full migration which hints us that the most influencing factor in the performance of a full migration is the disk.

Finally, for "light" migration, we moved only the VM without moving its VHD and since there is no disk activity for a vRAM copy, we achieved 117MB/s throughput which is effectively 93% of the theoretical GbE bandwidth. The memory system of modern computer is many orders of magnitude faster than the fastest network interfaces we have. Therefore, we will always have a bottleneck at the network when performing a context transfer.

B. RDC Measurements

When viewing video over RDC, we found that the amount of traffic generated by the RDC is less than the amount of traffic between the VM and the VHD where the video is stored. For games however, disk access is more intermittent. It typically occurs when a level is loaded from the disk. Until the level is finished, there is very little disk activity. However the RDC traffic is much higher since frames are constantly generated, user interaction is being transmitted and the content is not easily compressible in real-time So while gaming, the disk activity is very small while the RDC traffic is quite high. In Table (2), we present the averages of our observations.

TABLE II. OBSERVED RDC & DISK THROUGHPUT

Activity	RDC (MB/s)	Disk (MB/s)
Video	0.52	1.8
Game	4.4	1.3

For the video scenario we see that the disk traffic exceeds the traffic of the RDC. This is caused by overheads in disk access protocols used by the operating system of the VM. It is also caused by the operating system accessing the disk for various functions that may not be directly related to the user's activity. The disk traffic in the gaming scenario is lower than in video playback which tells us that during gaming, the disk is mostly idle and randomly accessed by the operating system for other functions. We also find that the RDC traffic for gaming is far greater than that of video playback. This tells us that RDC is not capable of compressing dynamic content as efficiently as it does with static content. We also know that gaming is an interactive application where user input is also transmitted as opposed to watching a video which is a more passive process.

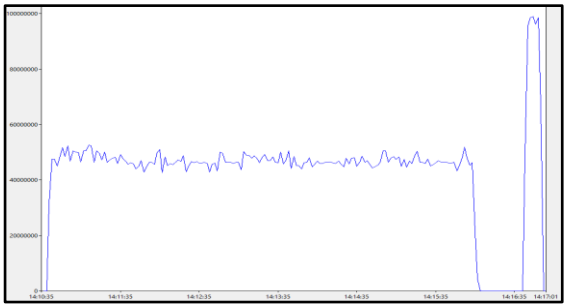

Figure 3 Full migration traffic

Based on the above observations we find that performing a full or light migration in a video playback scenario will bring negligible or no benefits, depending on usage patterns. We will analyse the game scenario because we see that the largest potential for traffic savings can be found there. We will try to estimate how much Inter-AS traffic can be reduced when dealing with a mobile user. By applying (4) to the traffic data from the previous section, we can find at what NDT we will have traffic savings. Table (3) shows the results for an average backbone bandwidth of 51.75MB/s which we derive by averaging the results for send/receive in Table (1).

TABLE III. FULL MIGRATION RESULTS

RDC (MB/s)	VM (MB)	Backbone (MB/s)	Savings (MB)	NDT (Mins)
4.4	15600	51.75	-8	64.08
4.4	15600	51.75	-4	64.10
4.4	15600	51.75	0	64.12
4.4	15600	51.75	5	64.13

We see that we start making traffic savings at NDT of approximately 64 minutes. This means that we cannot apply it to a user moving at high speed and connecting to small range networks. However, for a user that has a personal VM residing in a Cloud and accesses it from different networks at different times in the day (i.e. home or office), we can see that this method of dynamic migration can bring benefits. Next, we perform a light migration while maintaining the gaming scenario. Table (4) shows the results.

TABLE IV. LIGHT MIGRATION IN GAMING

RDC (MB/s)	VHD (MB/s)	VM (MB)	Backbone (MB/s)	Savings (MB)	NDT (Mins)
4.4	1.3	1024	117	-3.3316	5.63
4.4	1.3	1024	117	-0.2316	5.65
4.4	1.3	1024	117	2.86838	5.67
4.4	1.3	1024	117	5.96838	5.68

We see that the migration takes place much faster because the storage bottleneck does not apply. We also see that the VHD will still generate Inter-AS traffic but ultimately we are trading 4.4MB/s for 1.3MB/s.

V. Future Work & Conclusion

While it is true that Cloud technology has brought many advantages to the way we deliver services, it has also driven the demand for more multimedia and interactive services. Caching such content is not easily done and in some cases it may be impossible. However, traffic management for such scenarios is still a real problem that needs to be addressed if we are to provide networks that are resilient to the amount of traffic that Cloud services can generate. Mobility poses another challenge since mobile users can constantly switch networks leading to problems on where dynamic caching is best done.

In our experimentation, we tested a simple virtualisation setup that hosted a VM accessed by a single client. We compared the traffic generated between the VM and the client to the traffic generated between two independent hosts (playing the role of Cloud datacentres) while the VM was migrated between them. We also tested a scenario where the VM is running on separate host from where its VHD is located. We identified use cases where the traffic generated by the user's remote session is much higher than the traffic generated between the VM and its virtual disk. In these cases, it is almost always best to migrate the VM to the AS where the client resides without migrating its virtual disk. In other cases, the traffic between the VM and the user was actually lower than the traffic generated by the connection to the virtual disk. In these scenarios it is best not to move the VM.

From these results we can see that in order to achieve efficient dynamic edge-caching it is not only necessary to monitor QoS but we also need to monitor usage patterns. We cannot simply classify services in terms of QoS demands because this can lead to gratuitous caching which increases the network costs for very small benefit. We therefore need to monitor how each user accesses these services and how these services behave under different conditions. To achieve this, our preliminary opinion is that a global service performance monitoring scheme is needed that gathers information on how services behave, where they are located and where it may be best to move them in order to achieve a balance between Economic Traffic Management and QoS. The contribution of this paper is an algorithm that can be used in the Service Delivery Layer as proposed in [2,3] for the purpose of adding traffic localisation and management techniques to compliment QoS enhancements.

The next step for this project is to build a new test platform using a blade server that will allow us to repeat the experiments using WAN emulation [14] and ideally we would also like to repeat the experiments over the Internet using LTE connections on mobile devices. This will give us insights to how a migration will behave over the Internet and how RDC traffic is shaped over long distances. We are also planning of experimenting with the Network Memory Server [15] for Linux which will allow us to have a very fast storage system and remove the storage bottleneck from full VM migrations. As this is on-going research, we would appreciate any comments and recommendations that will help us further analyse the different aspects of this project and build a test platform that can provide meaningful results that we can share.

References

[1] OnLive, 2013. About the OnLive game service [online] Available at: http://www.onlive.co.uk/about [Accessed: 13 December, 2013]

[2] Sardis, F.; Mapp, G.; Loo, J.; Aiash, M.; Vinel, A., "On the Investigation of Cloud-Based Mobile Media Environments With Service-Populating and QoS-Aware Mechanisms", IEEE Transactions on Multimedia, vol.15, no.4, pp.769,777, June 2013 doi: 10.1109/TMM.2013.2240286

[3] Rodrigues, Joel J.P.C., Kai Lin and Jaime Lloret. "Mobile Networks and Cloud Computing Convergence for Progressive Services and Applications." IGI Global, 2014. 183-199. Web. 13 Dec. 2013. doi:10.4018/978-1-4666-4781-7

[4] Nygren, E., Sitaraman, R. K., and Sun, J. "The Akamai Network: A Platform for High-Performance Internet Applications", ACM SIGOPS Operating Systems Review, vol. 44, no. 3, July 2010.

[5] W. Zhu et al., "Multimedia Cloud Computing," IEEE Signal Processing Mag., vol. 28, no. 3, May 2011, pp.59–69.

[6] IEEE-SA, 2013. Guide for Cloud Portability and Interoperability Profiles. [online] Available at: http://standards.ieee.org/develop/project/2301.html [Accessed: 08 June, 2013].

[7] IEEE-SA, 2013. Standard for Intercloud Interoperability and Federation. [online] Available at: http://standards.ieee.org/develop/project/2302.html [Accessed: 08 June, 2013].

[8] Middlesex University, 2011. Y-Comm Research [online] Available at http://www.mdx.ac.uk/research/science_technology/informatics/projects/ycomm.aspx [Accessed, 2 July, 2013].

[9] Iosifidis, G.; Lin Gao; Jianwei Huang; Tassiulas, L., "An iterative double auction for mobile data offloading," Modeling & Optimization in Mobile, Ad Hoc & Wireless Networks (WiOpt), 2013 11th International Symposium on , vol., no., pp.154,161, 13-17 May 2013.

[10] Bennis, M.; Simsek, M.; Czylwik, A.; Saad, W.; Valentin, S.; Debbah, M., "When cellular meets WiFi in wireless small cell networks," Communications Magazine, IEEE , vol.51, no.6, pp.44,50, June 2013 doi: 10.1109/MCOM.2013.6525594

[11] Mapp, G., Katsriku, F., Aiash, M., Chinnam ,N., Lopes, R., Moreira, E., Porto Vanni, R.M., & Augusto, M. (2012). Exploiting Location and Contextual Information to Develop a Comprehensive Framework for Proactive Handover in Heterogeneous Environments. Journal of Computer Networks and Communications, 1-17

[12] Hobfeld, T.; Schatz, R.; Varela, M.; Timmerer, C., "Challenges of QoE management for cloud applications," Communications Magazine, IEEE , vol.50, no.4, pp.28,36, April 2012 doi: 10.1109/MCOM.2012.6178831

[13] T. Hoßfeld et al., "An Economic Traffic Management Approach to Enable the TripleWin for Users, ISPs, and Overlay Providers," FIA Prague Book. Towards the Future Internet — A European Research Perspective: IOS Press Books Online, ISBN 978-1-60750-007-0, May 2009.

[14] TATA, 2013. WANEM [online] Available at: http://wanem.sourceforge.net/ [Accessed: 30 July, 2013].

[15] Mapp, G., Thakker, D., and Silcott, D., 2007. The design of a storage architecture for mobile heterogeneous devices. In: Networking and Services, 2007. ICNS. Third International Conference on. IEEE Computer society. ISBN 0769528589