Dynamic traffic management for interactive cloud services: localising traffic based on network throughput and user mobility

Dynamic Traffic Management for Interactive Cloud Services

Localising traffic based on network throughput and user mobility

Fragkiskos Sardis, Glenford Mapp, Jonathan Loo and Mahdi Aiash
School of Science and Technology
Middlesex University
London, UK
f.sardis, g.mapp, j.loo, m.aiash @mdx.ac.uk

***Abstract*—Traffic localisation is an important aspect of traffic management on the Internet. The caching of content and its distribution from localized servers is one mechanism that enables traffic localisation and reduces transit costs for network providers while enhancing service performance. However, personalized, dynamic user content is often un-cacheable due to its nature. As we move towards a world of constantly connected mobile devices, these devices will focus more towards Cloud services as a means of adding capabilities that would otherwise be impossible to have in a small mobile package. This leaves us with the potential problem of having to deal with un-cacheable traffic from real-time interactive Cloud services that causes congestion on a global scale. In this paper we present a solution that considers the scenario of an interactive, personal service running on the Cloud and accessed by a mobile user. We attempt to localise traffic by moving the virtual machine in response to the user's estimated network dwell time. Results gathered from a prototype platform are presented as proof of the concept.**

Keywords—cloud services; traffic management; service localisation; live migrations; service portability

I. INTRODUCTION

Cloud computing has become a de facto standard in delivering services more efficiently and centralising resources in such way that reduces costs for service providers and clients alike. The vast amount of scalable and elastic resources available in the Cloud has greatly enhanced the capabilities of end-user devices by giving them access to Cloud-supported services such as storage and processing. This has been a great advantage for mobile devices where local resources are limited. Modern mobile devices feature multiple network interfaces of different technologies and in the future, seamless handovers between different networks will be possible in order to guarantee constant connectivity [1]. This will introduce a new generation of mobile thin-clients where all the processing will be done in the Cloud and the device will only be used to display information [2]. Depending on the application, different levels of Quality of Service (QoS) are required from the network for good service delivery. For a thin-client, a high level of QoS is mandatory for a good user experience because the device itself has very little capabilities and all the processing has to be done on the Cloud. Therefore any limitations of the network will immediately appear as limitations in the user experience. This puts network technology and traffic management in the forefront when it comes to delivering Cloud services, especially when featuring interactive multimedia content which often requires high bandwidth and low latency connections.

Traffic localisation is a term that covers various methods of keeping network traffic within an autonomous system. It is a form of traffic management that focuses on putting content as close to the users as possible. The concept behind it is to cache data or run services within a network so that clients of that network will not have to request them from third party networks. The incentive behind it has two aspects. The first aspect is economic and can be described as Economic Traffic Management [3], where the main concern is to reduce transit costs for network operators by reducing the amount of traffic that exits their network and incurs transit costs from other providers. The second aspect is related to performance enhancement by eliminating long network paths to a service or content.

Content Delivery Network (CDN) technology was developed as a caching and content distribution solution for the Internet [4]. A CDN consists of multiple datacentres that cache published content in various locations. Each location peers with different networks and delivers content to clients of these networks. This kind of localisation keeps the data within a network and its peers and thus improves the QoS for its clients while reducing transit costs for the network operator. While CDNs are very effective at delivering frequently-accessed static content and multimedia content such as video streams to a broad audience, it is impossible to use them for caching dynamic and interactive content.

Therefore, the model of Cloud-supported mobile thin-clients presents new challenges for networks and traffic localisation. Each user's virtual device will have a different setup in terms of computing resources, operating system, applications and user preferences. Consequently, this kind of environment is not homogenous enough for caching and localised distribution. Even if we consider that each user's virtual device starts with a default software package, any customisation done by the user will render it unique and therefore making it an unlikely candidate for caching. This means that CDN technology cannot contribute in localising this type of content and services. As a result, there can be a varying amount of inter-domain data pushed onto the networks between a thin-client and a virtual device depending on the application.

Multimedia applications such as games can generate large amounts of traffic for images and audio. The impact of delivering such a service to a mobile thin client is more easily understood by envisioning the example of a mobile user with a thin-client accessing a service via their home network (Wi-Fi) and subsequently leaving their home and connecting to an LTE network as they travel to a different location. Unless their Wi-Fi and LTE networks are from the same provider, there is a possibility that there are different transit networks that carry the data between the Cloud and the client. There is even a possibility that for the same provider, different parts of their network are served by different transit connections. In this case, if we wish to localise traffic, we will have to move the user's session to a different Cloud that is part of, or peering with the user's current network as shown in Fig. 1.

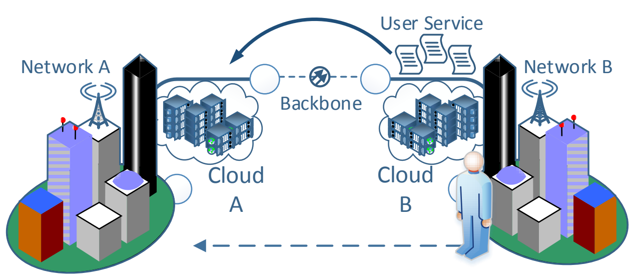

Fig. 1. Multi-network scenario for service migration.

In this paper we consider a Cloud-supported mobile thin-client under different usage and mobility scenarios. We develop a mathematical solution that localises network traffic by means of moving services. The inputs to this solution are the user's time on the network, the amount of inter-domain traffic that the service is generating and the size of the service itself. The unique contribution of this paper lies in the refinement of the work presented in [5] and [6]. We introduce new equations for traffic localisation in more complex scenarios along with a process flow diagram for making decisions on when to move a service. We also present the results from prototype testing platform that applies the proposed solution.

The rest of the paper is outlined as follows: Section II presents background information related to traffic management in the context of Cloud services and mobile-thin clients. Section III presents the developed mathematical equations used for traffic management. Section IV introduces the prototype implementation and the results gathered from experimentation. Finally, Section V suggests future work and concludes the paper.

II. Background

A. Cloud Context

In order to understand some of the terms and functions mentioned in this paper, we will first look at Cloud technology and present some relevant information.

A Cloud is a network of loosely interconnected computers and its infrastructure can be divided into three main sections: The first section is the Storage Area Network (SAN), where persistent storage for the Cloud and its services is provided. The second section is the Compute Cluster where all the processing for the services is carried out. The third section is the public-facing portal servers where clients connect to gain access to Cloud resources.

Clouds are based on virtualisation technology that obscures physical hardware from the software through the use of hypervisors. A hypervisor allows us to create a pool of virtual resources and manages instances of Virtual Machines (VM) that have access to these resources. The amount of virtual resources allocated to a VM is determined by the user or by the Cloud's administrator. These resources have the form of Virtual Hard Disks (VHD) in the NAS, virtual memory and processor cores in the compute cluster and virtual network interfaces that connect these virtual devices to each other through virtual switches. The virtual switches can connect to physical networks through bridging with physical interfaces.

A VM may run on any physical node in the compute cluster and hypervisors have the ability to migrate the VMs from one physical node to another for reliability and load balancing purposes. The migration can occur even while the VM is running and without interrupting its function. In this case it is called a live migration [7] and it involves transferring the entire working set of the VM over the network. Live migrations are achieved more easily when strict QoS requirements are met by the network interconnecting the nodes. This is due to the fact that a live migration transfers memory pages between two nodes while the VM is running and accessing those pages. It is therefore a race condition of copying and synchronising memory pages over the network at a rate higher than that of the VM changing them. At present, migrations can only occur between homogeneous Clouds, however, interoperability standards that will allow migrations between heterogeneous Clouds are currently in development [8].

In the context of this paper, we are particularly concerned with two traffic flows originating from the VM. The first traffic flow is between the VM and its (VHD) where the operating system, applications and user data are stored. The VM needs constant access to its VHD, otherwise it ceases to function. Again, there are strict QoS requirements from the network in order to provide a good connection between the VM and the VHD so that the functions of the VM will not be affected.

With the above information in mind, we will make two assumptions that apply to the context of this paper. The first assumption is that migrations occur between peering datacentres where the peering connection's QoS is controlled and is enough to support live migrations. The second assumption is that the VM has a home datacentre where its VHD resides and can access it through a connection that does not affect the performance of the VM. The VM's location may change as long as the above assumption is satisfied. We therefore assume a dedicated backbone connection between multiple peering datacentres that is dedicated to exchanging data between the Clouds.

B. *Mobility Context*

In the context of a mobile thin-client accessing a private VM, it is becomes harder to localise traffic when the user is constantly switching networks. Even when not moving, a user's device may switch from LTE to Wi-Fi or vice versa, depending on the conditions of the connection or through user intervention. Y-Comm [1] attempts to predict user mobility and the resulting connectivity changes through the use of network mechanisms that detect a user's speed and direction. This information is then placed on an overlay map of wireless networks and their area coverage over the user's path as demonstrated in Fig. 2. Y-Comm also provides solution for seamless handovers between heterogeneous networks as well as multi-homing solutions for service portability [9]. For this paper, we will assume that the network is able to detect a moving user or service and update the routing information between the service and its client in a seamless fashion.

In [10] Mapp et al. show that when a user's velocity and a network's area coverage are known, we can estimate for how long the user will be within the network's boundaries. The result is an estimation of the user's Network Dwell Time (NDT) which is then used for the purpose of allocating network resources proactively. NDT can also be used to inform a Cloud about the time that a user is going to be connected to a particular network. The NDT in conjunction with information about datacentres local to or peering with the network as presented in [11] can also help us calculate whether or not it is possible and desirable to move a service and localise its traffic.

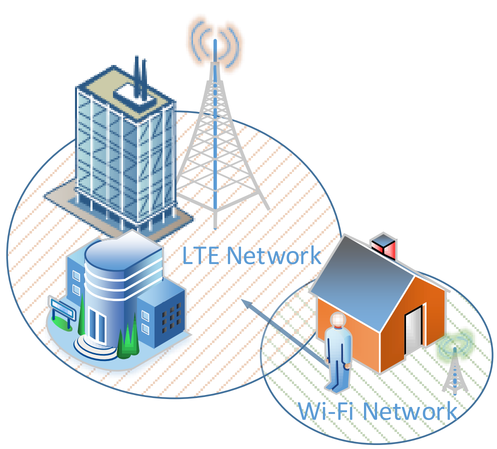

Fig. 2. Mobility-induced handover between heterogeneous networks

C. *Traffic Management Context*

To identify when traffic localisation is desirable we must first know the characteristics of the traffic. From a traffic localisation perspective, the main quality of interest is the network throughput of the service. There are two throughputs to monitor in this case: The first is the throughput between the service and the client and the second is the throughput between the service and other services such virtual hard disks. In the case of a VM for example, we will need to monitor the Remote Desktop Connection (RDC) throughput and the Virtual Hard Disk (VHD) throughput. In more complicated scenarios such as in Service-Oriented Networks, we may have to monitor additional traffic such as the connections between the composite services and its component services. The main goal, regardless of the scenario, is to eliminate the largest amount of inter-domain traffic flow by moving the service that generates it within a network.

Measurement can be done in three different ways. The first method expects the client device to report the RDC throughput to the service while the service itself is responsible for monitoring its own back-end throughputs. The second method involves the service doing the measurements for everything. The third method uses new transport protocols such as the Simple Protocol (SP) [12] that has the capability of reporting network metrics (i.e. latency and throughput) for individual flows. This information, combined with the NDT can tell us how much traffic a service will generate towards a user's network and help us identify cases where localisation is desirable.

D. *Migration Throughput*

Moving a VM will add data to the network so localising services is a process that actually adds to the inter-domain traffic for a brief period of time and should therefore be accounted for. So the first thing to consider is the size of the VM that needs to be moved. As mentioned previously, we will assume that Federated datacentres or even datacentres of the same provider have high QoS interconnections that will not interfere with the migration operation. VM migration over Wide-Area Network (WAN) links is still under investigation [13] but there are developments towards more efficient mechanisms that reduce the QoS requirements for a successful migration [14]. However, even when considering a private peering facility with guaranteed QoS, it would be unrealistic to assume constant throughput for the migrations. Therefore, once again, we need a mechanism that can probe the network and determine a realistic value of throughput which will determine how long it takes to move a VM on its memory size. Another factor that affects the migration time is the rate at which memory pages of the VM are being altered. The higher the rate of change, the longer it takes to sync memory pages between two hosts. Therefore, one additional condition to consider is that the rate of copying pages must be higher than the rate at which pages are being changed. This can be done by querying the hypervisor for a VM's memory access patterns and comparing that to the throughput of the connection between the source and destination hosts.

III. TRAFFIC MANAGEMENT EQUATIONS

Before attempting to expand on the equations presented in [6], we shall first analyse them by considering the assumption and findings of the original paper. The main assumption is that the entire VM along with the VHD is migrated and hence the traffic balance equation becomes:

$$\theta_{RDC} \times t_{NDT} > C_{VM} + (\theta_{RDC} \times t_{mig}) \quad (1)$$

Where, θ_{RDC} is the throughput of the RDC, t_{NDT} is the NDT of the user for a particular network, C_{VM} is the size of the VM (including VHD) and t_{Mig} is the migration time for the VM. This equation tells us that if the total amount of inter-domain traffic ($\theta_{RDC} \times t_{NDT}$) is larger than the size of the VM plus the amount inter-domain data transferred by the RDC during the migration ($\theta_{RDC} \times t_{mig}$), then a migration is desirable and will lead to traffic savings the through localisation of RDC traffic. This may also be applicable if the migration of the VM does not include its VHD, but only under the additional assumption that the traffic throughput between the VM and the VHD is zero; an unlikely scenario.

Experimentation results from the same paper suggest that moving the VHD is not very efficient given the large size of the VHD and therefore the large amount of data it will transfer over the network and consequently the prolonged migration time. So a better approach would be to decouple the VHD from the VM and consider moving only the VM. This approach adds a new variable to the equation which is the throughput between the VM and the VHD. We can now show that the more general equation is as follows:

$$\theta_{RDC} \times t_{NDT} > C_{VM} + \left(\theta_{RDC} \times t_{mig}\right) + \theta_{VHD} \times \left(t_{NDT} - t_{mig}\right) \quad (2)$$

Where θ_{VHD} is the VHD throughput and $\left(t_{NDT} - t_{mig}\right)$ is effectively the time that the user will be connected to the same network after the migration completes. Thus, we have now accounted for inter-domain traffic that the VM will generate after the migration if we decouple it from its VHD. However, this only covers the scenario where the VM is moving from its home location. It does not cover the scenario where the starting point of the VM is already on a remote location and therefore generates inter-domain traffic both on the front-end and the back-end. The two different scenarios are visualised in Fig. 3 and Fig 4.

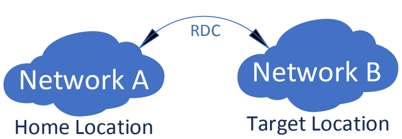

Fig. 3. Dual-network scenario for VM migration.

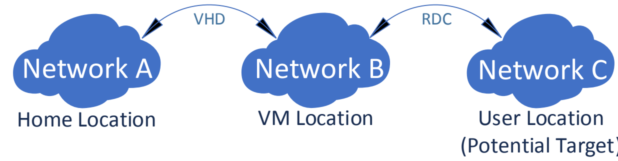

Fig. 4. Multi-network scenario for VM migration.

To address traffic management in the scenario of Fig. 4, we need a new set of equations. The first equation is applicable when the RDC traffic is greater than the VHD traffic. In this case we attempt to localise RDC traffic and allow the VHD traffic to cross domain boundaries.

We have:

$$t_{NDT} \times (\theta_{VHD} + \theta_{RDC}) > C_{VM} + (\theta_{RDC} + \theta_{VHD}) \times t_{mig} + \theta_{VHD} \times \left(t_{NDT} - t_{mig}\right) \quad (3)$$

Where $t_{NDT} \times (\theta_{VHD} + \theta_{RDC})$ is the amount of non-localised data put on the network from RDC and VHD for the full NDT duration, $(\theta_{RDC} + \theta_{VHD}) \times t_{mig}$is the amount of non-localised data put on the network by RDC and VHD while the VM is migrating and $\theta_{VHD} \times \left(t_{NDT} - t_{mig}\right)$ is the amount of data crossing network boundaries after the VM has migrated and for the remainder of the user's NDT. Using this equation we compare the amount of non-localised data that is put on the network for the duration of the NDT, with the amount of non-localised data that we will transfer during and after the migration. We can simply (3) as follows:

$$t_{NDT} \times \theta_{RDC} > C_{VM} + \theta_{RDC} \times t_{mig} \quad (4)$$

Next, we have to cover the opposite scenario, where the VHD traffic is greater than RDC. In this case the above equation is slightly modified to reflect that it is the RDC traffic that will not be localised.

$$t_{NDT} \times (\theta_{VHD} + \theta_{RDC}) > C_{VM} + (\theta_{RDC} + \theta_{VHD}) \times t_{mig} + \theta_{RDC} \times \left(t_{NDT} - t_{mig}\right) \quad (5)$$

So we have $t_{NDT} \times (\theta_{VHD} + \theta_{RDC})$ which is the non-localised data on the network for the user's NDT and $\theta_{RDC} \times \left(t_{NDT} - t_{mig}\right)$ which represents the amount of non-localised RDC traffic after the migration. We can simplify (5) as follows:

$$t_{NDT} \times \theta_{VHD} > C_{VM} + \theta_{VHD} \times t_{mig} \quad (6)$$

The last thing to consider is when the RDC throughput is equal to VHD throughput. In this case, the best choice is to default the VM back to its home location in order to save resources by involving only one Cloud in service delivery and also enhance disk performance by accessing the disk directly rather via WAN. In this case, the applicable equation is (5) or (6) as they express the elimination of VHD traffic.

Based on these equations, we now have a more general solution to localising traffic for a single-user VM that also takes into account the scenario presented in Fig. 4. The advantage of this approach is that it is more realistic than the one originally proposed in [6] because it does not require moving the VHD. Finally, when there are no traffic savings to be made, the priority is given to returning the VM to its home location to release resources on third-party Clouds. The complete flow chart is presented in Fig. 5.

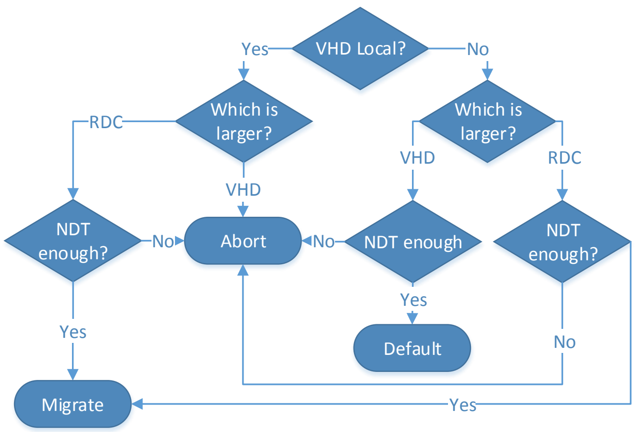

Fig. 5. Traffic Management Flow Diagram

IV. Prototype Implementation and Results

To test if the implementation of these equations is possible and if network throughput data can be measured and used along with these equations a prototype experimentation platform was set up. We will first describe the technical characteristics of the platform, followed by the test setup, the methodology of the experiments and finally the results.

A. Platform

The test platform uses two physical servers running Windows Server 2012 Enterprise in a domain configuration and Hyper-V enabled. The first node features and Intel Core-i7 920 CPU with 12GB of memory and a Kingston Hyper-X SSD for the operating system and the VHDs. The VHDs are placed on a network shared folder so that they can be accessed when the VM is moved to the second server. The server also features two Gigabit Ethernet interfaces, one acting strictly as a back-end connection for server administration and VM migration akin to the private network of a Cloud and the other interface acts as a front-end network which is used by the client to connect to their VM. The back-end uses a gigabit switch while the front-end uses a Wi-Fi access point 802.11n with a built in 100Mb/s switch. The second node features 4GB of memory with a Crucial RealSSD for the operating system and an Intel Core2Quad Q6600 CPU. It also features a Gigabit interface for the back-end connection and a 100Mb/s interface for the front-end. Finally, we have an admin console connected to the back-end network for the purpose of remotely controlling the servers and running the script manually. The physical setup is presented in Fig. 6.

There are two VMs configured on the network. The first VM acts as a domain controller with 1GB memory allocated and 2 virtual cores. It has a virtual connection to the back-end through the network interface and resides permanently on the first node. It is also running Windows Server 2012 Enterprise. The second VM is configured with 4 virtual cores and 2GB of memory and has a direct access to the front-end network via the network interface of the servers. It has not access to the back-end and it is running Windows 8.1. Finally, the client device is a laptop connected to the front-end access point using W-Fi with a constant bandwidth of 54Mb/s. Access to the client VM is achieved through Microsoft's Remote Desktop Connection.

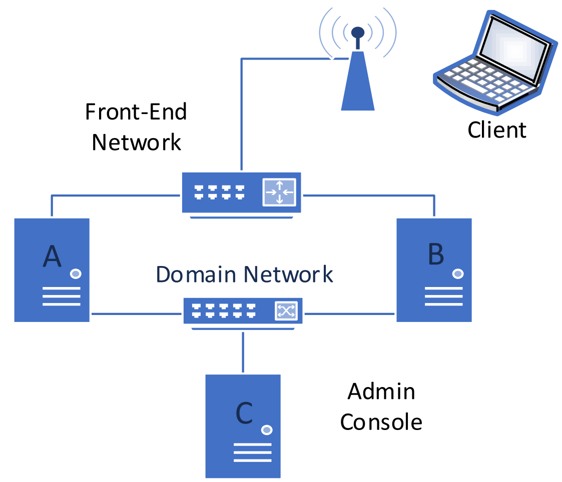

Fig. 6. Physical platform setup

B. Test Setup

The client VM is configured with a static IP on the same subnet as the client device. For testing purposes in order to generate RDC and VHD traffic in a controllable manner, two applications were installed. The first application is the game Pinball FX, and the second application is ATTO Disk Benchmark [15]. The Pinball game was selected as it is an example of a casual game that a user may play on a mobile device. It features 3D graphics and lighting effects that cause a high RDC throughput. Similarly, the Disk Benchmark stresses the VHD connection by sending/receiving a large amount of data to the VHD. To fall in line with current technology capabilities, the RDC resolution was set to full-HD (1920x1080) which is also the native resolution of the client device.

The equations were implemented in a PowerShell [16] script running on the physical servers. The first operation of the script is to read the hostname of the physical server and then attempts to match it to the network path of the client VM's VHD. The name of the client VM is manually input by the user. If a match is found the script assumed that the VM and the VHD reside on the same physical host. In this case, the script will measure the current size of the VM's memory and ask the user for an estimated migration throughput. It divides the VM size by the migration throughout to find the migration time. It also asks for the user to manually input the NDT since we don't have a mechanism or setup that can automatically calculate and provide this for us. Following this, the script starts measuring RDC throughput via the front-end NIC of the physical server and VHD throughput via the Hyper-V since the VHD is local to the VM. In case the VM is found to be on a remote location away from the VHD, it uses the back-end NIC of the physical server to measure VHD throughput.

Depending on the location of the VM and the balance between RDC and VHD traffic, the script uses the mathematical equations and the process flow diagram in Fig. 5 to determine if the VM should move.

C. *Methodology*

To determine an accurate migration throughput value, a set of experiments was carried out that involved the migration of the client VM while measuring the throughput at the back-end interfaces and the time it took to complete the operation. With the VM idle, the migration throughput was fully saturating the Gigabit bandwidth at 117MB/s. The throughput was reduced to 80MB/s while the test applications ran on the VM resulting in approximately 25seconds to fully transfer the VM to the other host. One thing to note is that the size of the VM varied during testing; however the actual throughput was always 80MB/s with a small deviation of ±2MB. This value was used as input to the script in order to estimate migration times.

The first part of the testing was focusing on gaming in order to stress the RDC connection. A game session was started and the player played normally while the script was launched on the host that contained the VM. Different NDT times were used and the results were recorded. The same experiment was repeated with the VM running on the second host machine, away from its VHD. The second part of the test focused on stressing the VHD connection. The game was closed and the disk benchmark application was launched at its default test settings. Once more, the same test was repeated with the VM residing on the second host, away from its VHD. Finally, the VM was allowed to idle for a few minutes to allow Windows to perform memory clean-up and the user launched productivity applications such as word and spreadsheet processors and the Internet browser. For this last part of the experiment, the front-end network was connected to an Internet router for the VM to gain access to the Internet. In order to differentiate the Internet traffic from the RDC traffic that goes through the front-end interfaces, the script was modified to count only the sent bytes, therefore counting only the traffic that is being transmitted by the VM to the client laptop. The sent traffic to the web was negligible during testing since the user was restricted to browsing news websites and therefore only a small amount of web requests was sent by the VM while a large amount of data was downloaded from the web which we didn't want to include in the measurements and hence we measure only sent traffic. The overall impact of the sent web traffic to the measurements was negligible as it was measured to be a few kilobytes as opposed to the megabytes transmitted by the RDC connection.

D. *Results and Analysis*

Before presenting the detailed results from the experiments, it should be noted that while running productivity applications the VHD and RDC were mostly idle. This always resulted in the script averaging the traffic to 0MB/s and aborting the migration as the throughput was never adequate to make any traffic savings within a realistic NDT for a mobile user. This is because the minimum NDT calculated was in the order of several days of consecutive use which would make it unlikely for any user to remain connected on the same network for this long and without interrupting the session at some point or switching applications. These results are not included in the tables in order to simplify the presentation. The main observation from this however, is that under light usage scenarios, migrations are unlikely to occur.

TABLE I. SCRIPT RESULTS - VM AT HOME LOCATION

VM (MB)	NDT (sec)	RDC (MB/s)	VHD (MB/s)	Result
1858	700	4.44	0.6	Moved to user's network
2048	500	4.4	1.5	Aborted: Insufficient NDT
1292	500	0.5	63.32	Aborted: VHD>RDC

We start by presenting the results of testing with the VM on the local host in Table 1. The higher values are highlighted in red. While gaming, RDC traffic was consistently larger than VHD traffic. The opposite result was found while the disk benchmark was running. During the gaming test, the disk activity was quite low which indicates that the game runs from memory. While it is true that more advanced games may generate higher disk activity as data is being streamed from disk to memory, in the case of casual games, that typically have a smaller memory footprint, it is normal to expect this behaviour. The findings were confirmed with other casual games available through the Windows Store such as solitaire minesweeper and mah-jong. The difference in these games is that they don't feature rapid changes on the display such as flashing effects or fast-moving 3D objects and as a result, their RDC activity is quite small comparatively. In the gaming test the migration was successful at an NDT of 700 seconds which is approximately 11.6 minutes; a realistic NDT for a mobile user who may have joined a network with a large area coverage such as LTE or for someone who stopped at a shop. The operation was aborted for an NDT of 500 as there was not enough time to make any traffic savings. The main insight from these results is that a casual game with rich 3D graphics and lighting effects is likely to trigger a migration due to its high RDC throughput and comparatively small VHD throughput.

Finally, in the disk benchmark scenario, we see that the VHD traffic greatly exceeds RDC traffic and because the VM is already at its home location, it would make no sense to move it to a remote node as the VHD traffic would have to be put on the network. One thing to note is the very low usage of the RDC connection when displaying the desktop and applications that don't feature moving objects. This confirms that for most simple applications that can be Cloud-supported and accessed by a mobile device, a migration is unlikely to occur as a result of the user interacting with them.

In summary, with the VM at home location, the only case where a migration was triggered was when playing a game and the user had an estimated NDT of 11.6 minutes.

In Table 2, we present the second round of test results, with the VM residing on the second node, away from its VHD. In this case, the first thing to note is that we see the opposite behaviour compared to the first scenario when it comes to the disk benchmark. Because the VHD traffic is larger than the RDC, the script decides to move the VM back to its home location in order to eliminate the VHD traffic from the network and enable faster disk access for the VM. An NDT of 500 seconds was sufficient in this case for the migration to occur and once again this proves to be an unlikely usage scenario for a mobile user as applications that may cause such high disk traffic will typically be office productivity applications or database access which will most likely occur in an office environment. An alternative in this case would be to consider moving the VHD but that would most likely require a very high NDT depending on the amount of data inside the VHD.

TABLE II. SCRIPT RESULTS - VM AT REMOTE LOCATION

VM (MB)	NDT (sec)	RDC (MB/s)	VHD (MB/s)	Result
2048	500	2.265	0.2	Aborted: Insufficient NDT
2048	1500	2.11	0.54	Moved to user's network
1924	500	0.5	5.46	Moved Home

In the game benchmarks, we see a similar behaviour to the previous scenario. The script will move the VM to a new target location if the NDT is sufficient or it will abort the operation if it is insufficient. One thing to note in this scenario is that the NDT for a successful migration was increased to 1500 seconds compared to 700 seconds in the first test. This is down to three different reasons: The first reason is that the VM size became larger by almost 200MB. This could be down to how the operating system manages the memory or because of the game caching more of its data into the memory. The second reason is that the RDC throughput became smaller after the VM moved to the second host. We will explain why in the next paragraph. The third reason is because we now also have the VHD traffic to compensate for while migrating. This leads to increased NDT according to (3).

The RDC throughput inconsistency between home and remote locations is not due to it running remotely from the VHD but rather because the CPU of the second host is of an older generation and could not render the game at the same frame-rate, resulting in smaller throughput. The game experience did not deteriorate in terms of latency but the texture quality was reduced and the audio was intermittent at times. Trying to confirm this claim, we looked at the CPU utilisation of the VM at each node. When running the game on the first node which has a more powerful CPU, the usage was approximately 45% while on the second node it reached 95%.

The last thing to note concerning this round of tests is that because only two nodes were available for this experiment, the script considered as remote location, the first node which also happens to be the home location. If a third node was available, it would have been the correct target host for the migration in the gaming scenario. The script is written in a way that differentiates this but for the test, in two different branches of the code, the first host was manually put as the target.

In summary, we find that service migrations in this scenario are not a common occurrence and only happen when the applications put a lot of data on the network and the user's mobility pattern is such that they will be part of a network for a reasonable amount of time. One traffic characteristic that affects the occurrence of migrations is the difference between RDC and VHD throughput. The greater the difference, the more likely it is for a migration to occur, either by sending the VM to the client's new network or by defaulting it to its home location. Due to this behaviour, we find that under light usage scenarios, migrations are unlikely to happen.

V. FUTURE WORK AND CONCLUSION

Preliminary results from the testing platform show that there are traffic savings to be made from this approach to traffic localisation in the context of personalized, interactive Cloud services. The script presented in this paper is reactive in its nature which means that it can be triggered manually through user intervention or reactively by the network when a handover is imminent. This means that some compensation needs to be made in the NDT to account for the script's running time which is typically around 20 seconds. We don't currently have a mechanism that can trigger the script automatically but in the context of Y-Comm we are developing such a solution. Furthermore, we are working on optimizing the SP so that it can report throughput values to the script. This will negate the lengthy process of sampling the traffic by the script itself. The SP is also in a better position to provide accurate throughput results as it can monitor individual flows as opposed to the entire traffic going through an interface. We also plan to rewrite the script in C++ and .NET languages to provide a package solution that can be launched more easily.

Considering the test platform, we are looking at adding a third physical node or transferring the entire setup to a blade server for a more realistic Cloud setup. To address the performance problem that results in varying throughput of 3D graphics between hosts, we can enable Remote FX [17] on Hyper-V. This requires compatible graphics processors that were not available at the time of testing.

We have also experimented with an alternative version of the script that features proactive network selection by accepting an input of a series of networks that the user is likely to cross in his path, and their respective NDTs. The script selects the first network with high enough NDT for a migration or measures the combined NDT of the user's path and estimates if the VM should be returned to its home locations when it already resides on a remote network and the VHD traffic throughput exceeds the RDC. We thus have a script that can plan its actions based on future actions of a user rather than on their current location and mobility. The input on the user's path and network NDTs is given by the network and the Y-Comm research group is working on developing such a mechanism. The results from experimentation with the proactive script will be published in a future paper.

This approach to traffic management can be extended to QoS management. With the help of SP, we can retrieve QoS information for the connection between a user and by comparing this information to the required QoS values for good service delivery, we can determine if a service should move to a network closer to the client. The approach is more complicated than simply measuring the performance of the current network because we will also have to consider the performance of the target network before deciding to move a VM there. Such results can be gathered by probing the path to the remote network before the user actually connects to it and determining if a migration is going to bring any benefits to the QoS. This approach is currently in development as a theoretical model using queuing theory to determine if the service rate of a network can cover the service requests without pushing a client too high in the queue.

This novel approach to localizing traffic in the context of Cloud services for mobile users can be applied to other areas of research such as Service-Oriented Networks. Traffic can be measured between component services that make up a composite service and the component services can be dynamically allocated based on where most of their traffic originates or ends. In this paper we have proved that it is plausible to move services reactively based on the traffic they generate and on user mobility patterns. There are many open questions still such as the problem of VM migrations over WAN and the connection handover process when the VM moves to an entire different subnet or administrative domain in the context of Federated Clouds. The Y-Comm Research group is focused on providing solutions for network-related problems in the context of QoS and mobility for the next generation of network. We welcome any input that can help us introduce Cloud technology and service portability to the framework as a means of enhancing QoS and traffic management at the WAN scale.

REFERENCES

[1] Middlesex University, 2014. "Y-Comm Framework" [online] Available at: http://www.mdx.ac.uk/research/science_technology/informatics/projects/ycomm.aspx [Accessed: 11/07/2014]

[2] Patanapongpibul L, Mapp G.E and Hopper A. "An End-System Approach to Mobility Management for 4G Networks and its Application to Thin-Client Computing" Published in the ACM SIGMobile, Mobile Computing and Communications Review (MC2R), July 2006.

[3] Hoßfeld, T., Hausheer, D., Hecht, F. V., Lehrieder, F., Oechsner, S., Papafili, I., & Stiller, B. (2009, May). An Economic Traffic Management Approach to Enable the TripleWin for Users, ISPs, and Overlay Providers. In Future Internet Assembly (pp. 24-34).

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

[5] 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," Multimedia, IEEE Transactions on, vol.15, no.4, pp.769,777, June 2013.

[6] Sardis, Fragkiskos; Mapp, Glenford; Loo, Jonathan; Aiash, Mahdi, "Dynamic Edge-Caching for Mobile Users: Minimising Inter-AS traffic by Moving Cloud Services and VMs," Advanced Information Networking and Applications Workshops (WAINA), 2014 28th International Conference on, vol., no., pp.144,149, 13-16 May 2014.

[7] Microsoft, 2013. "Virtual Machine Live Migration" [online] Available at: http://technet.microsoft.com/en-gb/library/hh831435.aspx [Accessed: 11/07/2014]

[8] IEEE, 2014. "InterCloud – IEEE Cloud Computing" [online] Available at: http://cloudcomputing.ieee.org/intercloud [Accessed: 15/07/2014]

[9] Mapp, G.; Aiash, M.; Guardia, H.C.; Crowcroft, J., "Exploring Multi-homing Issues in Heterogeneous Environments," *Advanced Information Networking and Applications (WAINA), 2011 IEEE Workshops of International Conference on* , vol., no., pp.690,695, 22-25 March 2011

[10] G. Mapp, F. Katsriku, M. Aiash, et al., "Exploiting Location and Contextual Information to Develop a Comprehensive Framework for Proactive Handover in Heterogeneous Environments," Journal of Computer Networks and Communications, vol. 2012, Article ID 748163, 17 pages, 2012.

[11] Krishnan, Rupa, et al. "Moving beyond end-to-end path information to optimize CDN performance." Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference. ACM, 2009.

[12] Riley, L., and Mapp, G., 2014. "yRFC3: The Simple Protocol Lite (SP-Lite) Specification" [online] Available at: http://www.mdx.ac.uk/Assets/yrfc3-SP-Lite.pdf [Accessed: 11/07/2014]

[13] P. Ruth et al., "Autonomic Live Adaptation of Virtual Computational Environments in a Multi-Domain Infrastructure," Proc. 2006 IEEE Int'l. Conf. Autonomic Computing,Washington, DC, 2006.

[14] S. Hacking and B. Hudzia, "Improving the Live Migration Process of Large Enterprise Applications," Proc. ACM Int'l. Wksp. Virtualization Tech. in Distrib. Computing, New York, NY, 2009, pp. 51–58.

[15] ATTO, 2014. "Disk Benchmark" [online] Available at: http://www.attotech.com/disk-benchmark/ [Accessed:18/07/2014]

[16] MSDN, 2014. "How Windows PowerShell Works" [online] Available at: http://msdn.microsoft.com/en-us/library/ms714658.aspx [Accessed: 11/07/2014]

[17] MSDN, 2014. "Microsoft Remote FX" [online] Available at: http://technet.microsoft.com/en-us/library/ff817578(v=ws.10).aspx [Accessed 11/07/2014]