On-Demand Service Delivery for Mobile Networks

Fragkiskos Sardis
School of Engineering and
Information Sciences
Middlesex University
London, UK
f.sardis@live.mdx.ac.uk

Glenford Mapp
School of Engineering and
Information Sciences
Middlesex University
London, UK
g.mapp@mdx.ac.uk

Jonathan Loo
School of Engineering and
Information Sciences
Middlesex University
London, UK
j.loo@mdx.ac.uk

Abstract—**Support for mobility has become a key requirement for computer networks. Users now expect to be connected at any time and from anywhere. At the network level, this will be done using vertical handover techniques across multiple network technologies. However at the service level there is no agreed mechanism by which to support mobile users. Current service delivery techniques that depend on overprovisioning are no longer valid as they are inefficient in terms of network resource management. Furthermore, mobile users now want access to demanding applications such as multimedia services, i.e., iPlayer, YouTube and 3D-TV. These services often have constraints in terms of bandwidth and latency that need to be properly supported in the mobile environment. This paper outlines the challenges involved in the design of a service delivery model for mobile nodes with high Quality of Service requirements. The proposed approach uses service migration techniques that take into account user mobility and network conditions so as to ensure efficient use of network resources. In this paper, we introduce the novel concept of user clustering to help us decide when and where services should be migrated. We also show how this idea can be used to support a video streaming service.**

Keywords- mobile; services; clustering; migration; NMS

I. Introduction

Technological advancements in recent years have made mobile devices more accessible to a large number of people. Smart-phones and tablets are increasingly becoming more common and users now expect to be connected to the Internet while they are on the move. In addition, these devices now come with multiple network interfaces such as Wi-Fi, High-Speed Downlink Packet Access (HSDPA) and Bluetooth which allow them to have network connectivity at all times. The concept of vertical handovers will allow these devices to stay connected to the Internet as they move from the range of one network technology to another [1]. Transport mechanisms including Mobile IPv6 [2], Stream Control Transport Protocol (SCTP) [3] and Multipath TCP [4] attempt to support ubiquitous connectivity at the network level. However, service delivery issues in mobile environments also need to be addressed.

Legacy networks use an overprovisioning approach to the delivery of services. This approach relies heavily on allocating resources in anticipation of user requirements over long timeframes. Such an approach is unable to adapt, in an appropriate way, to a mobile environment where users are constantly moving around and results in significant waste of network resources. Hence a new approach is needed.

Furthermore, in recent years, the popularity of audio and video streaming, as well as browser applets and HTML5 has made the Internet more multimedia-centric. Multimedia applications have strict temporal and Quality of Service (QoS) requirements that have to be continually supported in this mobile context. Continual service provision for these applications requires that we keep response times to a minimum. 4G technologies such as Long Term Evolution (LTE) offer more bandwidth to the users and higher QoS, which in turn, increase the need for fast service delivery on the server side.

One way to address these problems is by creating on-demand instances of a service. These instances can run on multiple servers and in different geographical locations in order to balance the load and where possible, put a service closer to mobile users. Better load balancing and better QoS can be achieved through this service delivery scheme compared to existing methods.

This paper addresses the issues of Service Migration in the context of delivering services to mobile devices. The concept of User Clustering in which we group users into a cluster is introduced. This cluster is tracked in order to determine when and where a service migration should occur. Bringing these two concepts together allows us to create a service delivery platform capable of creating service resources based on user movement and demand. The rest of the paper is outlined as follows: Section II discusses related work. Section III details the key areas of the problem. In section IV we propose solutions for each area. Section V introduces the test platform we will use to carry out our proposed solution. Section VI demonstrates how we aim to use a working model of this technology to achieve video streaming. The paper concludes at Section VII looking at future aims.

II. Related Work

A service-centric networking platform has been developed at Princeton University. The SCAFFOLD [5] architecture is capable of providing flow-based Anycast with moving service instances. Rather than retaining their addresses as the hosts move, SCAFFOLD allows end-point addresses to change dynamically. This enables hosts to migrate across Layer-2 boundaries. When end-points move,

ISBN: 978-1-61208-164-9

in-band signaling is performed to update the remote end-points of established flows. Thus, when a service moves, the network automatically directs new requests to the new location. This architecture is aimed at maintaining service availability in the event of server failure but it can also work in our context.

The Y-Comm framework [6] is a network architecture that supports vertical handover. Y-Comm uses two frameworks: the first is the Peripheral framework that manages different functions on the mobile terminal. The second is the Core Framework which deals with operations in the core network to support different peripheral networks. As shown in Fig. 1, these frameworks are brought together to represent a future telecommunications environment that supports heterogeneous devices, disparate networking technologies, network operators and service providers. Although, the two frameworks share the first two layers, they diverge in terms of functionality but the corresponding layers interact to provide support for heterogeneous environments. In the context of this paper, we are interested in the service and application environment layers which are used to support services and their delivery to mobile nodes. In Y-Comm, the service platform layer is independent of the underlying networks but is used to facilitate service level handovers in an integrated fashion.

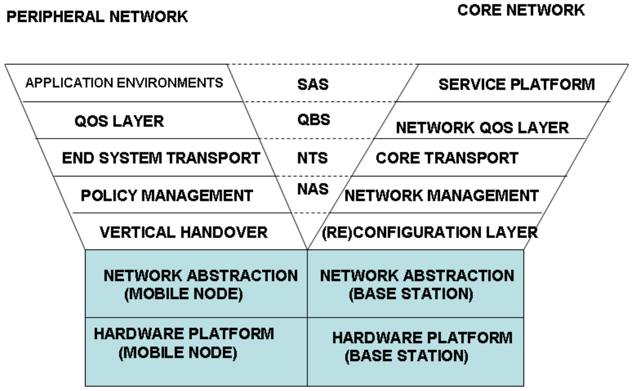

Figure 1. The Y-Comm Framework

III. PROBLEM ANALYSIS

Existing service delivery technologies rely on providing a fixed amount of resources that is greater than the expected service requests within a timeframe. In a scenario where the requests exceed the capacity of the service, some users are not served or in the worst-case scenario, the entire service fails. This does not happen frequently but there are examples due to flash crowds or Denial-of-Service (DoS) attacks. The continuously increasing popularity of always-online mobile devices results in a higher number of constantly connected users. Moreover, as mobile devices gain more processing power, users are able to multitask in the form of connecting to multiple services concurrently i.e. weather updates, video streaming, social networking and file sharing. Hence, a more efficient resource management scheme should be used in the future.

Using a proactive service model allows us to create resources on-demand and gives us the ability to move services closer to users, thus improving QoS and decreasing routing costs. The key challenge is the integration of service migration strategies with models of user mobility. Service migration solutions should also be aware of the network status and available resources before attempting to move a service. Therefore, an efficient service migration strategy must take into account network and server resources as well as user mobility in a scalable fashion.

In terms of user mobility, it is necessary to be able to track the movements and service usage patterns of mobile nodes. To achieve this, we need to be able to track a node’s location. In order to efficiently allocate service resources to a geographical area, we need to group similar users into a cluster. The cluster size and its movement will define when service migration is desirable. The reason we are clustering similar users together is so that we can attach a service instance to the group and use it as a mechanism that triggers service migration. An added advantage of this method is that we can treat user clusters as Multicast groups for services such as Internet Television and Internet Radio.

There are also network factors that determine when service migration should take place. Latency, congestion and bandwidth are some of the metric that we should consider before a service is moved between locations. QoS requirements for each service are also an important factor that will determine when migration is desirable and beneficial. Similarly to monitoring users, we need to be able to monitor these resources in real-time. For example, if the network reports higher congestion at a service migration point, there will be little benefit in moving the service to that point. So network statistics are vital to the system.

Another aspect of the problem is the handover at the service level. This needs to be integrated with the network-level handover in order to support seamless connectivity. In order to replicate a service to a new server, we need to transfer relevant user information and service content. There are two types of service-level handoff occurring in sequence. The first type of handoff is between the instances of the service. This transfers user and service data such as active sessions, user files and data caches. The second handoff is between the clients and the service. It binds the client nodes to the new service instance by registering new service IDs and follows with a network-level handover that reroutes all connections from the old service’s IP address to the new one’s. We assume that a transparent addressing scheme is used in which devices always have the same IP address, even if they move across heterogeneous networks [7].

Finally, node tracking introduces privacy concerns and there may be users who wish to opt-out from tracking but still want to access services. Furthermore, some services may hold sensitive data and may have security requirements such as encrypted file systems and encrypted connections. The system should consider these factors and services must not migrate to servers that do not support adequate security levels.

IV. SOLUTION APPROACH

A. User Clustering

In order to make correct service decisions, we need to be able to track users as they move and the services to which they are currently subscribed. The concept of user clustering attempts to group similar users together by allocating them into clusters based on their location and their service subscriptions. User tracking can be done either by Global Position System (GPS) capabilities on the mobiles nodes or by GSM antenna tracking. Another possibility is Wi-Fi hotspot tracking as used by Google for Street View. Hotspot tracking is similar to GSM antenna tracking in which the mobile node can estimate its location by scanning for nearby Wi-Fi hotspots and comparing the results to an online database that holds hotspot names and locations. These technologies provide varying accuracy in user position ranging from a few meters to the size of a city block but for the purposes of our system, they all provide enough accuracy.

A user cluster is defined as a group of users subscribing to the same service and sharing the same approximate location. In our initial investigation we are proposing the use of simple parameters. For example, we define a cluster as having a Centre (c) and a Radius (r). Furthermore, for the purposes of scalability, we will limit the maximum Cluster Population (p). A cluster is first created when the first user subscribes to a service at a specific location. When a cluster reaches maximum population, another cluster is created in the area. An example of the clustering mechanism is shown in Fig. 2.

The concept of clustered mobility introduces two kinds of dynamics that need to be tracked: The first type of dynamics involves the collective movement of a cluster as its members move from one location to another. This can be defined by Velocity and Acceleration vectors (u) and (a) respectively. The second dynamic involves users joining and leaving a cluster either by subscribing or unsubscribing from a service or by crossing the boundaries of a cluster. This is defined as Cluster Entropy (E). We should also consider the merging of clusters in cases where two clusters attached to the same service come close together and the sum of their members is less than the Maximum Population (p).

The parameters discussed above need to be tracked in real-time so as to form a correct model of the movement of a cluster. Using that model we are able to predict the probable speed, direction and location of a cluster at a point in the future. Finally, all cluster data such as cluster ID (CID) and others mentioned above will be held in a database and updated in real-time so that the service migration logic can process it.

B. Network Resource Monitoring

By using the data from the clustering database and combining it with real-time network metrics we can create an algorithm that will instruct a service to migrate to an appropriate location. Real-time network metrics will include data such as network congestion, latency, and available bandwidth. Additionally, we need to know of candidate servers that can accept a migrating service. Metrics such as available CPU and storage resources as well as security parameters will be used by the migration logic to decide which server is best suited for the type of service that is being moved.

Network metrics are a good indication of the state of a particular network or subnet. Knowing the available bandwidth and latency will allow the Migration Logic to make correct decisions on where to move a service or if it is worth moving the service at all. For example, if a cluster is about to move to an area where wireless signals are not strong and latency is increased while bandwidth is decreased, the Migration Logic will issue an urgent migration of the service to a server as close as possible to the cluster in order to improve QoS as much as possible. In a scenario where network conditions are good the service will be migrated more casually. Similarly, if a server reports an overloaded status, the Migration Logic may decide to move a service to another location in order to balance the load evenly across servers.

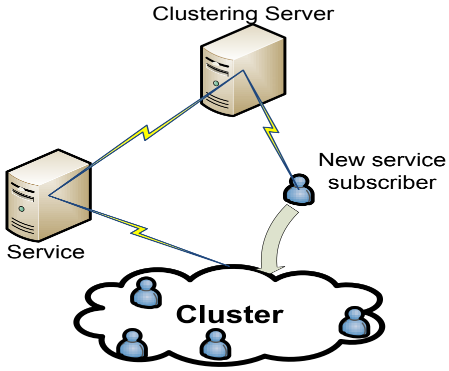

Figure 2: User clustering example.

To gather this data we can use Simple Network Management Protocol (SNMP) and routing protocol information. We can also query participating servers for available resources. This data will be held in a database and updated at frequent intervals. This information allows us to know where best to move a service. In addition to the above, the Network Resources Database (NRD) will also hold Security Level Agreement (SLA) data that define what level of security a participating server offers and whether or not it can accept sensitive services.

C. Migration Logic

The Migration Logic should be able to process cluster mobility data and attempt to predict the future location of a cluster. Eventually the centre of the cluster is going to exceed a distance threshold from the location of the service. At that point, it will instruct a service to replicate itself to a location as close as possible to the predicted location of the centre of the cluster. This will give users lower network latency and will also decrease congestion on a large scale. An additional effect will be decreased costs to the service provider due to decreased packet routing and switching. If the Migration Logic predicts that a service migration is not going to improve QoS, it will attempt to recalculate the data after a time period. This process will be repeated until the distance threshold is not exceeded anymore, a successful migration occurs or the cluster itself ceases to exist.

If the system fails to predict the future location of a cluster due to erratic movement or other factors, the algorithm will only instruct a service to replicate to a server closer to the cluster's present location. Other parameters that will affect such decisions include the cluster's population, the type of service and whether or not it would be cost effective to replicate an entire service.

When the criteria are met for a replication event, a call packet will be sent by the replication logic to the instance of a service that needs to be replicated. The packet will include a flag for migrating or replicating a service and the address of the target server. If the call is for migration, then the service will make a copy on the target server and delete itself from the initial location. This scenario applies when a cluster moves to a new location as a whole and is demonstrated in Fig. 3 below.

A replication call takes place when a new instance of the service needs to be created without deleting the old instance. In this scenario, the new instance of the service is created with a new service ID and bound to the cluster ID. Replication calls typically apply to load balancing events when more resources are needed. Finally, a kill signal can be sent to a service if it is not needed anymore or a service can be set up so that it shuts down if there are no service requests after a defined time period. If the service is then needed, the requests will be routed to the nearest live service instance and a replication event will occur.

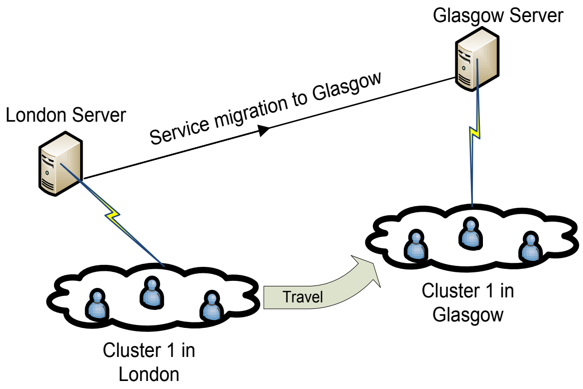

Figure 3: Service migration example.

D. Migration Mechanism

The migration mechanism should be capable of moving a service and its context from the memory of one server to the memory of another server. In addition to moving the service itself, user data should also be moved to the new location. The mechanism should also be capable of launching the service in the new location and terminating the one in the old location. Service level handover should take place once the service has migrated to the new location.

Depending on whether the service in question is stateful or stateless, a handover at the service level may not be needed. Services that rely on stateful user sessions will need to pass session information to the new instances. Otherwise, a stateless service, such as streaming a video, does not need this information passed on, as it is up to the client to request the next block ID of the video. These are called lightweight handovers.

After the service has been migrated, the Cluster Logic can Multicast to all the members of a cluster the new server address. At that point the connection level handover takes place and new connections are initiated while connections to the old service instance are terminated. In order to explore service replication, we will look at replicating instances of the Network Memory Server (NMS) detailed below.

V. NETWORK MEMORY SERVER

A. NMS Features

The Network Memory Server [8] is an example of a simple, stateless service; it stores blocks of data from clients in its memory (RAM). Clients can create, read, write and delete blocks of data. The NMS is primarily made as a storage platform for mobile users. In order to provide support for mobility, the NMS is divided into two parts: The Mobile Memory Cache (MMC) and the Persistent Storage Server (PSS). The MMC initially runs on the same network as the mobile client. If a client moves to another network then the MMC is migrated in order to achieve better performance. The PSS offers permanent data backup for the MMC and there is a level of redundancy implemented so that an MMC can be backed up in multiple instances of the PSS. This is achieved by a multicast call to all the associated PSS. Furthermore, the NMS stores data at the block level in order to provide maximum flexibility in terms of storage and access. The client is responsible for any added-value abstractions, for example, a file-level abstraction. An independent low-level socket interface is used for the server and so the overall interface as seen by applications on client machines is unstructured and can be manipulated as necessary for the needs of the client.

In addition to the above, the MMC provides security to the clients by employing an access rights mechanism over the blocks of data. The owner of a block has full access to it. A user cannot read any blocks that do not belong to them unless given read access to the blocks. Two levels of encryption are also supported in the NMS. A lower level of encryption is used between the client and server as they are on the same network and a higher level of encryption is used between the MMC and the PSS because they are in different networks.

B. NMS Migration

At the moment, the NMS is not capable of migrating from one server to another and only a simple prototype of the PPS is implemented. We will initially explore NMS replication by trying to transfer storage blocks from one MMC instance to another. Because the NMS is stateless, this is a good starting point for our work.

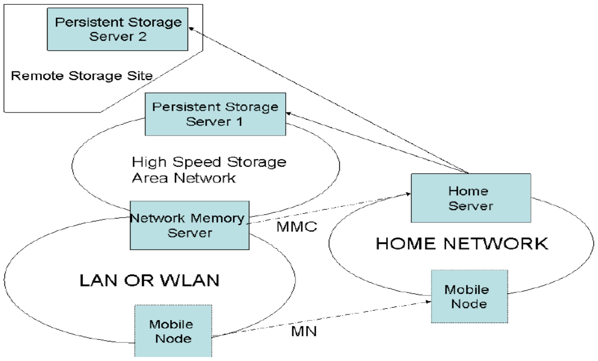

Figure 4: Example of NMS work scenario.

Additionally, we will attempt to implement the PSS in order to have a level of data redundancy during replication. The above will be implemented and tested on a blade server and service migration will be explored across the independent blades. Fig. 4 shows an example of service migration with the NMS using the MMC as the mobile front-end between the WLAN at work and the Home Network. This migration is done using the Context Transfer Protocol (CXTP) [9,10] as shown in Fig. 5 below.

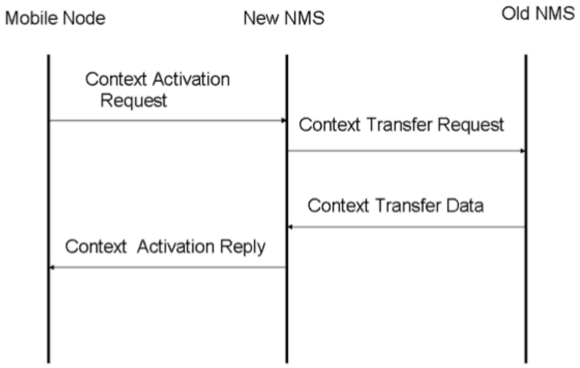

Figure 5: NMS migration model.

VI. Mobile Video Streaming Service

The NMS example detailed above is a simple mechanism for network storage delivery and will be the first service we will attempt to migrate. We would like to support a mobile video streaming service in which the streaming service itself, as well as the video file streamed, is migrated across a network. Fig. 6 shows the various components of a framework to support mobile video streaming. The entities and their interactions are described as follows:

The Clustering Mechanism keeps track of User Clusters as they move and the location of the services for each cluster. Once the distance between a user cluster and the service location exceeds a threshold, the clustering mechanism reads the Server Location Database for an available server closer to the location of the cluster. It then sends a signal to the Service Replication mechanism to move the service, suggesting a target migration point.

The Service Replication mechanism queries the target server for available resources and upon a positive response, triggers the replication of the servlet to the new location. Once the servlet is moved to the new location, a signal is sent to the Storage Migration mechanism to copy the relevant video files to an NMS server that is close to the new servlet. The Storage Migration mechanism then uses the CXTP to transfer the cached videos from the old storage server to the new storage server. Once this is completed, it sends a signal back to Service Replication.

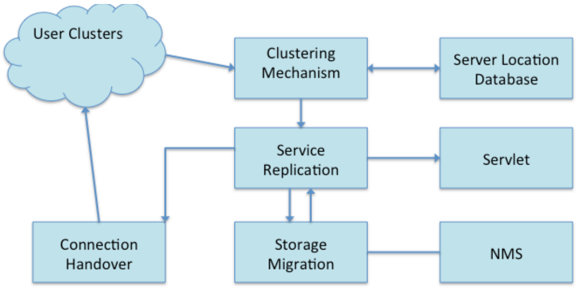

Figure 6: Replication framework for mobile video streaming.

Finally the Service Replication sends a signal to the Connection Handover mechanism, which in turns informs the mobiles nodes within the cluster of the new location of the videos.

In order to make the service efficient and keep it simple, mobile nodes pull the video streams directly from the Storage Server using prefetching algorithms [11]. This means that the Storage Architecture can be stateless, with the relevant state information stored in the mobile nodes. This also means that the actual data is streamed directly between the mobile nodes and the Storage Server.

VII. Future Work and Conclusion

In order to explore service migration, a test platform will be built here at Middlesex University. We initially aim to use the test platform to explore service migration on the NMS. The first step is to have a working prototype capable of replicating the MMC to a server in the local network. The replication signal will be sent to the MMC manually at first in order to simplify the development process.

The second step is to automate the replication signal using real mobile devices. The initial replication logic will be able to group users into a cluster and track them as they move around on campus. In addition, it will be able to remove users from a cluster if they leave the network or stop accessing the service.

To test this functionality, we are implementing a GSM network on campus that will allow us to use mobile devices to access the NMS. As we move around on campus, the devices will handover between GSM base stations and we will use those signals to trigger service migration between servers. In the long term we are aiming to use NMS service migration in order to achieve video streaming for support of mobile video server applications.

In this paper, we have briefly 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 User Clustering and Service Migration can bring a better solution to the efficient management of network resources while providing a high quality of experience for users. The authors recognize that there is much to do and would welcome feedback on this paper.

REFERENCES

[1] Mapp, G., Shaikh, F., Aiash, M., Vanni, R., Augusto, M., and Moreira, E. 2009. Exploring efficient imperative handover mechanisms for heterogeneous wireless networks. In: Duresi, Arjan and Barolli, Leonard and Enokido, Tomoya and Uehara, Minoru and Shakshuki, Elhada and Takizawa, Makoto, (eds.) Network-Based Information Systems, 2009. NBIS '09. International Conference. IEEE. ISBN 9781424447466

[2] Johnson, D., Perkins, C., and Arkko, J. RFC 2775 - Mobility Support in IPv6. IETF, June 2004.

[3] Dtewart, R., Xie, Q., Morneault, K., Sharp, C., Schwarzbauer, H., Taylor, T., Rytina, I., Kalla, M., Shang, L., and Paxson, V. RFC 2960 - Stream Control Transmission Protocol. IETF, October 2000.

[4] Ford, A., Raiciu, C., Hadley, M., and Bonaventure, O. RFC 6182 - TCP Extensions for Multipath Operation with Multiple Addresses. IETF, July 2011.

[5] Freedman, J. M., Arye, M., Gopalan, P., Ko, Y. S., Nordström, E., Rexford, J., and Shue D. 2010. Service-Centric Networking with SCAFFOLD. Princeton University, September 2010.

[6] Middlesex University, 2011. Y-Comm Research. [online] Available at http://www.mdx.ac.uk/research/areas/software/ycomm_research.aspx [Accessed: 25 July 2011].

[7] Mapp, G., Aiash, M., Guardia, C. H., and Crawford, J. 2011. Exploring Multi-homing Issues in Heterogeneous Environments. In: Proceedings of the 1st International Workshop on Protocols and Applications with Multi-homing Support in Biopolis, Singapore 22nd- 25th March 2011.

[8] 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

[9] Loughney, J., Nakhjiri, M., Perkins, C., and Koodli, R. RFC 4067 - Context Transfer Protocol (CXTP), IETF, July 2005.

[10] Patanapongpibul, B. L., Mapp, G., and Hopper, A. 2006. An End-System Approach to Mobility Management for 4G Networks and its Application to Thin-Client Computing. ACM SIGMOBILE Mobile Computing and Communications Review, ACM, July 2006.

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