Exploring a New Framework to Build Mobile QoS-Aware Applications and Services for Future Internet

Onyekachukwu A. Ezenwigbo[1], Yonal Kirsal[2], Vishnu V. Paranthaman[1], Glenford Mapp[1], and Ramona Trestian[1]
[1]Faculty of Science and Technology Middlesex University, London, NW4 4BT
[2]Electrical and Electronic Engineering Department European University of Lefke, Lefke, Turkey
Email: {OE130, V.Paranthaman, G.Mapp, R.Trestian}@mdx.ac.uk; ykirsal@eul.edu.tr

Abstract—Mobile users are making more demands of common networks by running applications such as network streaming of audio, video as well as immersive gaming that demand a high Quality of Service. One way to address this problem is by making services mobile, such that the services will move closer to the users as they move around. This ensures that low latencies are maintained between the client and server resulting in a better Quality of Service. At present, new architectures such as the Y-Comm framework, attempt to provide a platform to support intelligent service migration mechanisms. However, what is also needed is to provide QoS mechanisms in order to facilitate efficient service migration. This requires techniques to measure the QoS in terms of bandwidth, latency and burst characteristics at various locations to which the server could be migrated. In addition, the emergence of Software Defined Networking as well as new end to end control mechanisms such as the Network Management Control Protocol and the development of new transport protocols will allow a new framework to support mobile QoS-Aware applications and services that will be a key part of the Future Internet. This paper explores the development of a new applications and services framework for Future Internet that replaces the traditional IP framework. New mechanisms are developed to decide when and where to move services and a video on demand scenario is analysed. An analytical model is investigated to provide results based on bandwidth and latency. The results show that this approach is valid and should lead to better QoS and better Quality of Experience for mobile users.

Index Terms—Quality of service, cloud computing, software defined networks, mobile services, low latency protocols.

I. Introduction

We live in a rapidly changing networking environment. Firstly, we are seeing the emergence of new types of networks. Vehicular Networks are being developed to support Connected and Autonomous Vehicles (CAVs). Tactile networks, conveying a sense of touch over the network, will also increase network use. These types of networks require much lower latency compared to current networks and in the case of vehicular networks much higher bandwidths. Secondly, we now have a world of heterogeneous devices. These entities, called HetNet devices, have several network interfaces and so can be always connected to the Internet using vertical handover techniques. Finally, there is currently a wide and diverse range of applications with different Qualities-of-Service (QoS) that must be supported on various hardware platforms. Some applications such as network audio streaming require low delays and even lower jitter. Other applications such as file transfers can tolerate large delays but require total reliability.

The need to support these diverse set of applications with different Quality-of-Service (QoS) requirements means that support for QoS needs to be incorporated into future networks. In this new environment, QoS will also include security requirements and the system should be able to balance between the security and QoS concerns. Y-Comm [1] is an architecture that has been designed to build heterogeneous mobile networks. It attempts to integrate communications, mobility, QoS and security into a single platform. It divides the Future Internet into two frameworks: the Core Framework and the Peripheral Framework.

The emergence of Software Defined Networking (SDN) and Network Function Virtualization (NFV) [2] is helping us meet these challenges in terms of providing networks that are better managed by allowing finer control of data flows in the networking infrastructure such as in routers and switches. By centralising the high-level connectivity and routing control, an SDN controller monitoring several data switches can therefore have a much wider and almost instantaneous view of what is going on in its network that would take a normal router considerable time to develop. NFV allows the installation of in-band management routines, which can be used to optimise various flows across the network. These developments are very positive. However, there are very few end-to-end mechanisms defined in SDN and hence the default action is to fall back on the Internet Protocol (IP) Framework.

A. The IP Application Framework

Though IP has been a very successful network protocol, it is inadequate to meet new challenges because of several deficiencies such as the lack of support for multi-homing, no agreed QoS model and inflexibility in coping with rapid network changes; for example, fast handover in mobile environments such as vehicular

Manuscript received March 15, 2018; revised September 20, 2018.
doi:10.12720/jcm.13.10.559-573

networks. A new platform needs to be considered to deal with these challenges.

In addition, applications use transport endpoints and thus require transport-level protocols. For Internet Applications there has been a limited number of protocols being used for communication: the Transmission Control Protocol (TCP) is used to provide total reliability through checksums and retransmissions. While on the other hand, the User Datagram Protocol (UDP) provides no reliability but delivers packets between endpoints using ports. The Real- Time Protocol (RTP) and its variants provide time-stamped, unreliable connections. Recent protocols such as the Stream Control Transmission Protocol (SCTP) [3], are used to address specific issues such as multi-homing.

Though these protocols have served us well, they are unable to deal with future requirements. As previously mentioned, new applications now run on networks that require low latency. In addition, applications now want more direct control of their transport provision for various reasons. Firstly, transport protocols provide end-to-end connectivity, however, the data path model can be too restrictive. So TCP provides reliability but does it in a stream-like manner: there is no support for message boundaries making transactional or client/server support harder to implement. It is therefore very difficult to put hooks into this data path to deal with different situations; this can now be done very easily using NFV. Secondly, TCP uses the slow start and congestion avoidance algorithms to implement in-protocol flow control. However, applications now require more direct QoS support at the transport level. Hence there must be better cooperation between the application/server and the transport protocol that is been used.

This therefore explores the development of a new framework to support mobile QoS-aware applications and services by bringing together new technologies to provide an implementation platform for future internet.

The rest of the paper is structured as follows: Section 2 outlines the related work. Section 3 looks at a service migration framework for mobile services. Section 4 discusses the new framework being proposed. Section 5 details the QoS-aware applications and services environment. Section 6 explains the experimental set-up. Section 7 shows the experimental results. Section 8 introduces the analytical model for intelligent server migration while Section 9 shows the Markov model of the proposed system. Section 10 examines VoD case study. Section 11 concludes the paper.

II. Related Work

A. Analysis of IP Framework

The IP application framework is shown in Figure 1 and consist of five layers. Though very successful, it will be difficult to meet these challenges. Firstly, mobile nodes now have several interfaces. This means that in IP each interface has an IP address; hence a heterogeneous device with many interfaces will have several IP addresses and is very hard to determine these interfaces are co-allocated on the same device. This leads to unnecessary router caching problems since the system is unable to detect the co-location of several interfaces.

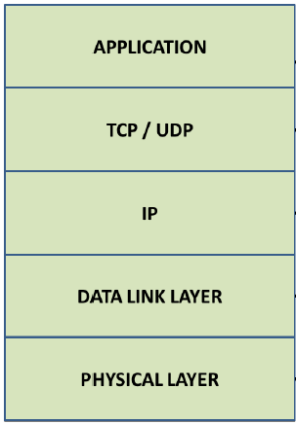


Fig. 1. The IP framework.

This can be solved with the use of a location ID server. The second issue is that there is no agreed QoS model; the two models Intserv and Deffserv have failed to gain universal acceptance leading to poor resource allocation. Finally IP is unable to work seamlessly in high speed vehicular environments where the point network attachment is constantly changing [4].

B. Transport Protocols

There has been a lot of work done in the area of transport protocols over the years. Most of the original work centred on trying to allow specific applications to operate in an asynchronous manner. For example, the protocol for the Network File System (NFS) originally ran over UDP to keep the server end entirely stateless [5]. Other research efforts into transport protocols tried to provide end-to-end connectivity in high speed networks such as ATM [6]. Interest in userspace transport protocols was heightened with development of the Xpress Transfer Protocol (XTP) [7], which was fast and was designed to support demanding applications such as multimedia. This effort led to the design of the A1 transport protocol which was used to show multimedia video over ATM [8]. However, as noted previously, without enough CPU cycles it was difficult for these efforts to take hold. During this period there were also efforts to implement a user-space version of TCP [9]. These efforts revealed that implementing user-space TCP was a non-trivial exercise [10].

C. The Y-Comm Framework

Y-Comm integrates communication, mobility, QoS and security. Y-Comm is divided into two frameworks: core network and peripheral network. It also introduces the idea of the Core End Point (CEP). The CEP is located at the edge of core network and connects different

peripheral networks to the core network. This type of architecture shown in Fig. 2 enables edge computing.

According to [11] the CEP has a communication function such as providing Cloud facilities and it also has a computing function used by services which can run on the CEP. The core network consists of a super-fast backbone and fast access networks which are attached to the backbone. The backbone network is made fast by the use of optical switches while the access networks are upgraded using Multiprotocol Label Switching (MPLS) techniques.

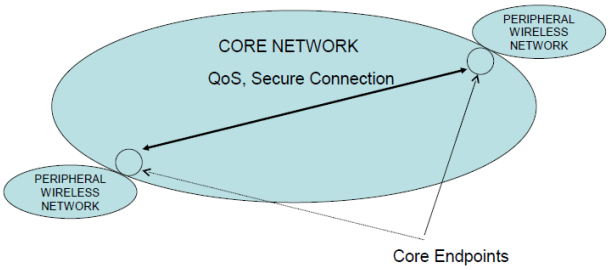


Fig. 2. The concept of core-end points.

On the other hand, the peripheral network will be dominated by the deployment of wireless technology. This means that the characteristics of the core network will be very different from the characteristics of the peripheral wireless network on the edge. By considering the above described changes in the network structure, different research efforts such as the Daidalos II architecture [12], the Mobile Ethernet framework [13] and the International Telecommunication Union (ITU-T) [14] have been working on defining a new architecture for heterogeneous networks.

Other network architectures for mobile systems such as Hokey [15], Ambient Networks [16] and Mobile Ethernet [13] have also been explored. HoKey looked at issues of secure handover in heterogeneous networks while Ambient Networks concentrated on supporting seamless connectivity in diverse networks. Mobile Ethernet adopted the Core/Peripheral structure like Y Comm but assumed an Ethernet-type Core. A comparison of these systems indicates that Y-Comm offers the most functionality and flexibility [17] while integrating various key mechanisms [18], [19].

D. Mobile Edge Computing

In order to analyse the effects of edge computing on reducing web response time authors in [20] derived a formula that reduces the response time of web pages by delivering objects from edge nodes. They investigated the effect of edge computing in different web categories such as sports and news. They were able to achieve this with their numerical evaluations using the data obtained by browsing about 1,000 web pages from 12 locations in the world.

Furthermore, authors of [21] proposed a model for system latency of two distribution processing scenarios by analysing the system latency of edge computing for multimedia data processing in the pipeline and parallel processing scenarios. They confirmed that both models can follow the actual characteristics of system latency. With regard to delay constrained offloading for mobile edge computing in cloud-enabled vehicular networks, the authors in [22] proposed a type of vehicular framework for offloading in a cloud-based Mobile Edge Computing (MEC) environment. They were able to investigate the computation offloading mechanism. The latency and the resource limitations of MEC servers were taken into consideration which enabled the proposal of a computation resource allocation and a contract-based offloading scheme. The scheme intends to exploit the utility of the MEC service provider to satisfy the offloading requirements of the task.

Given the significance of increased research in combining networking with MEC to support the development of 5G, the authors in [23] investigated the conceivable outcomes of engaging coordinated fiber-wireless to get networks to offer MEC abilities. More predominantly, imagined plan situations of MEC over Fi-Wi networks for typical Radio Access Network (RAN) advancements were explored, representing both network architecture and enhanced resource management.

Moreover, authors of [24] showed the architectural description of MEC platform along with the key functionalities. They agreed that the radio access network is enhanced by the computation and storage capacity provided using MEC. The primary benefit of MEC is to allow significant latency reduction to applications and services as well as reduced bandwidth consumption. The enhancement of RAN with the MECs capability can rely on its edge server cloud resources to provide the context-aware services to nearby mobile users in addition to conducting the user traffic forwarding.

For performance evaluation of edge cloud computing systems for big data applications, acceptable performance was revealed in [25] using Hadoop to build a visualisation machine for small clouds. In [26], [27] and [11], the intended functioning of the projected system has been presented in an attempt to determine if the migration of a service is required. The proposed model allows services to migrate from one cloud to another.

III. Service Migration Framework for Mobile Services

In a traditional cloud to cloud service migration, each Cloud is uniquely identified by a Cloud ID due to their rendered services. Each Cloud will have a number of resources which will actively advertise to the Orchestration servers. The Service Orchestration is a key component that is used to verify the identities of all servers on the network. In addition, the Service Orchestration knows the Service IDs and Public Keys for each service cloud.

It is pivotal to advance a novel service architecture that permits services to be organised, derived or moved to

support mobile users in an attempt to deliver a comprehensive fixed number of mechanisms to support mobile services. In order to achieve this, the system allows for algorithms that integrate the organisation of traffic and the QoS requests of the flow. As illustrated in Fig. 3, this novel framework which has six layers was recommended in [11], and they include:

- **The Service Management Layer:** The function of this layer is to identify the tasks of the service, catalogue the service in a service registry and obtain an exclusive service ID. In essence, it controls the provided service as it determines the minimum assets required by cloud and networking infrastructure in order to run the service, including network QoS and storage needs as well as computing resources.

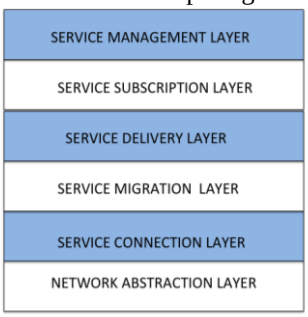


Fig. 3. Service-oriented framework for mobile services.

- **The Service Subscription Layer:** This layer allows clients to subscribe to services as it takes care of the actions needed by universal clients to access the service. Furthermore, it allocates for a new subscriber an exclusive client ID, a given Service Level Agreement (SLA) as well as determines accounting and payment tools.
- **The Service Delivery Layer:** The layer grants a given client access to the service. It does this by mapping the SLA to a given QoS and then certifies that the designated server and attendant networks can match the needed QoS. The service also accepts notifications and prompts regarding handovers and either duplicate or moves the service closer to the user based on received notifications.
- **The Service Migration Layer**: Migration or movement is usually undertaken at the command of the Service Delivery Layer. Here, this layer is in charge of duplicating or moving services to varied cloud platforms to encourage good QoE for the mobile user.
- **The Service Connection Layer:** This layer handles the ongoing connection between a client and the service and feeds back alterations in network and transport parameters, the likelihood of bandwidth or interruption to or suspension of the Service Delivery Layer.
- **The Network Abstraction Layer:** Subject to the network architecture and addressing, this layer oversees the function of getting a service to interface with varied kinds of networks as it maps into IP networking with TCP/IP. The ability to do this is split between the QoS and Transport Layers in Core and Peripheral Frameworks in more progressive systems like Y-Comm.

IV. The New Application Framework

Fig. 4 shows the new Application Framework. It consists of five layers which are detailed below:

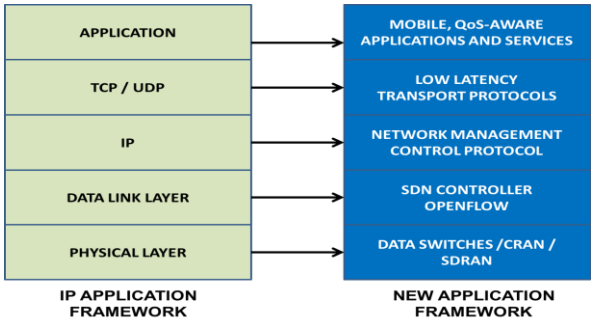


Fig. 4. New framework for building future applications.

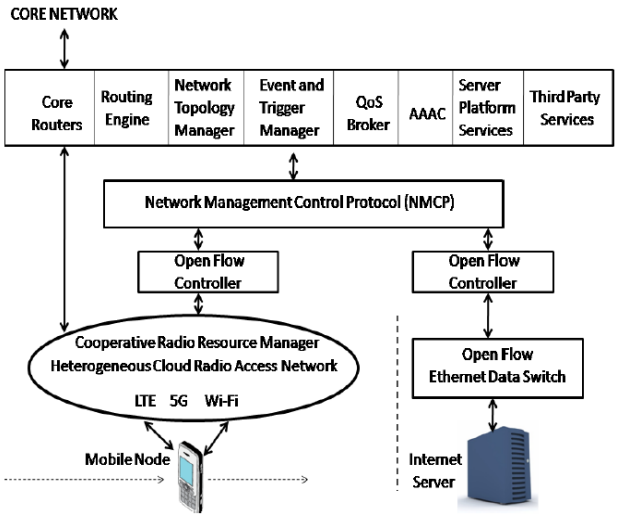


Fig. 5. The operational structure of a core endpoints.

A. Data Switches/CRAN/SDRAN

The proposed system will evolve with the development of new mobile technologies as shown in Fig. 5. This evolution will allow the smooth management of local heterogeneous networks by the Heterogeneous Cloud Radio Access Network (H-CRAN) and the Cooperative Radio Resource Manager (CRRM). H-CRAN will be used to access and control individual networks which CRRM will be used to optimize the overall radio access environment. CRRM will also support OpenFlow and hence the upper layers of the architecture can remain unchanged. The use of NFV and SDN at the Core Endpoint will also facilitate the softwarisation of radio technologies as proposed in 5G with the deployment of Cloud-RAN [28] at the Core Endpoint.

B. SDN Controller Open Flow

As shown in Fig. 5 the SDN controller controls access to both the H-CRAN and OpenFlow Ethernet data

switches. The controller interfaces to the upper layers using NETCONF interface using the YANG data model [29].

C. Network Management Control Protocol

The Network Management Control Protocol (NMCP) is used to allow the high-level network management functions and services discussed above to control and manage networking infrastructure. NMCP can be implemented by directly translating it into OpenFlow commands or by using a number of emerging Northbound APIs. NMCP also supports various communication entities: An endpoint is a device that can send or receive data. Mobile nodes and servers are examples of endpoints. Endpoints can support a set of different types of addresses. So an endpoint can have an Ethernet address, an IPv4 address, an IPv6 address, a SIM number as used in a mobile phone, etc. A link is a direct connection between two entities. A path is a connection between two endpoints. A path is made up of one or more links. A data-flow is the movement of data between two endpoints. Data-flows allow us to specify the actual data flowing along physical links. NMCP commands are divided into 5 groups:

- Link commands are commands to create and remove links as well as to activate and deactivate links. A link must be activated in order to forward packets
- Path commands are used to create, modify and delete paths
- Data-flow commands are used to create, modify, delete as well as to merge and demerge flows
- Parameter commands are used to get and set parameters
- Events Notification commands are used to set and delete event notifications

Connections in NMCP work by specifying links between endpoints and core network elements such as Core Endpoints as shown in Figure 6. Each link involved in the connection is specified using a TUPLE which specifies end points on the link as well as a forward connection label (fcl). An fcl is needed to forward any packet along a link and is treated like a capability and hence cannot be tampered with. The fcl specifies which addresses should be used to communicate over a link. Once the links between the end points are specified, it is possible to create a path using the TUPLEs specified. The data flow between endpoints represents the data being exchanged and is specified as a flow along a specified path. Once this is done the connection can be activated and the two end points can send data with each other.

The improvement of NMCP over IP is that it is not bound by a specific address format. The system decides which network technology can be used on each link, and this arrangement is able to readily adjust to changes to network topology. In addition, by making data-flows first class objects we also need not associate them with any network technology that allows us to implement things such as vertical handover because we can easily specify that a flow can be changed to go on a different path. Finally, the fcl can be used to specify a given quality of service required by applications using a given connection. This means that it is easy to find out if the QoS on the network is being broken.

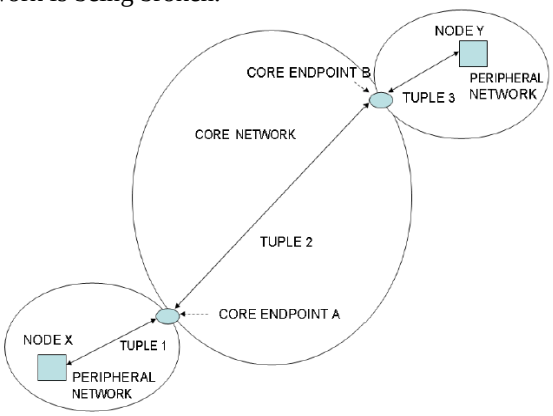


Fig. 6. Connection setup using NMCP.

D. Low Latency Protocol

New networks such as VANET networks require low latency and high bandwidth. In addition, new features can be used to fine tune transport protocols to application requirements. These includes:

- Running Efficiently in User Space: Since the transport protocol should be under the direct control of the application, it must run efficiently in user-space. Though running protocols in the kernel has advantages such a lower latency and guaranteed CPU cycles as kernel code normally executes at a higher priority than user space, running transport protocols in the kernel results in a huge amount of cross talk for all applications. So this means that a reliable fast video stream could be affected by activities from other applications. Secondly, because of the development and proliferation of multiprocessor architectures, which are now common even on PCs, it now very hard to argue that there is not enough CPU cycles in user-space to run transport protocols efficiently. Running in user-space will eliminate transport crosstalk and allow applications to be able to directly tune protocol parameters without the need for obscure socket system calls. Support for user space protocol processing is being actively pursued by several companies including TCP Offload [30] and the Data Processing Data Kit (DPDK) initiatives [31].
- Variable Reliability: Applications should be able to apply different reliability characteristics to different connections using the same protocol. So a transport protocol should be able to provide the entire spectrum: from totally reliable to totally unreliable connections.
- Selective retransmission by default: Protocols such as TCP use the go-back-n mechanism which can result in many packets being retransmitted even though they already been received at the other endpoint. So it is necessary that future protocols implement Selective Retransmission by default.

- Support for Forward Error Correction (FEC) functionality: Most transport protocols provide check summing and retransmission of packets to assure reliability. However, for applications that require low latency, retransmissions are seldom beneficial. In this situation, FEC techniques are used to ensure reliable reception. So streaming network audio could use FEC rather than just dropping corrupted packets.
- The ability to tune specific aspects of the protocol: This becomes very relevant for certain operations. So one parameter that should be changeable is the window size of a given transport protocol. This may be due to buffering issues but it could be used to support other communication events such as handover [31]. So when a handover to another network begins, the protocol closes its window preventing other packets being sent until the handover takes place where after its window size can be re-opened. Other parameters can be indicated are the maximum message size, etc.
- Support of priority for different end-to-end data flows: This has become a key issue as different types of data flows are being transported and so there might be times when you want to send packets on certain connections with different priorities.
- Up calls from the transport protocol into the application: Most transport systems use PUSH-PULL mechanisms developed by the traditional socket layer libraries where senders transmit or PUSH data towards the client while receivers retrieve or PULL the data from the underlying socket for the connection. However, in many cases, a server may wish to simply provide an upcall on the receipt of a service request message from the client.
- Providing alternative for flow control: In current transport protocols, applications have no say how flow control is done. TCP uses a sliding window based on congestion and receive window parameters as well as slow start and congestion avoidance algorithms. These mechanisms have proven to be effective but at times have been too conservative.

E. Simple Lightweight Transport Protocol (SLTP)

SLTP is an example of a low latency lightweight protocol that has been designed to support the new QoS-Aware framework. This motivation for designing SLTP came for the need to support research into services using Cloud based environments [32] as well as to provide low latency and tuneable support for Vehicular and Haptic Networks.

The SLTP Header

Fig. 7 shows the Diagram of the SLTP while Table I shows the length of the individual fields.

SLTP Packet Types

SLTP supports a number of packet types as shown in Table II.

SLTP Flags

SLTP FLAGS comprises a field containing 8 bits. Their functions are detailed in Table III.

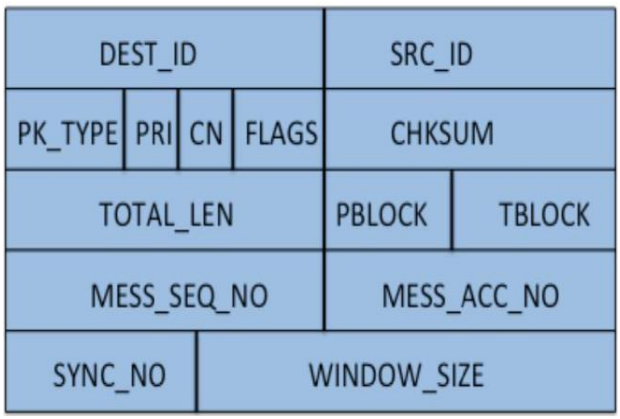


Fig. 7. The structure of SLTP header (Total Size is 20 bytes).

TABLE I: HE FIELDS OF SLTP AND THEIR FUNCTIONS

FIELD	BITS	FUNCTION
DEST_ID	16	Connection_Id at the remote end
SRC_ID	16	Connection_Id at the local end
PK_TYPE	4	Type of packet
PRI	2	Priority of the packet
CN	2	Congestion Notification Indication
FLAGS	8	Indicates actions needed to process the packet
CHUKSUM	16	Uses the TCP Checksum
TOTAL_LEN	16	Total length of the packet
PBLOCK	8	Current block or fragment
TBLOCK	8	Total number of blocks in the message
MESS_SEQ_NO	16	Sequence number of the last message sent
MESS_ACC_NO	16	Sequence number of the last message received
SYNC_NO	10	Random number to prevent replay attacks
WINDOW_SIZE	22	The Receive Window Size

TABLE II: PACKET TYPES AND THEIR FUNCTIONS

PACKET_TYPE	FUNCTION
STRAT	First packet transmitted on a connection
REJECT	Signals that the connection request has been rejected
DATA	Data packet
ACK	Acknowledgement (ACK) packet
NACK	Used for selective retransmission
END	Used to close a connection
FIN	Final packet sent
ECHO	Used to measure RTT
ECHO_1	First back-to-back packet
ECH0_2	Second back-to-back packet
STATUS	Used to maintain flow control
IDLE	Sent when there is no data to send
CWIN	Used to change the window size

In SLTP, when a connection is started, each side measures the bandwidth and burstiness of the connection through a modified packet-pair approach in which two

packets of a given size are sent back-to-back and the round-trip times of each packet is measured as well as the time-difference, d, between the packet replies as shown in Fig. 8. In SLTP, ECHO 1 and ECHO 2 packet types are used to perform this test.

The diagram in Fig. 8 shows how bandwidth is measured in SLTP, here two packets are sent from source to destination. Here, t1 and t2 are the times when the packet has started being sent from the source and t3 and t3 are the times when the ECHO 1 and ECHO 2 are received back at the source after being echoed by destination. The round trip time is given as (t3- t1) or (t4- t2) and therefore the bandwidth will be given as S/(RTT/2) where S is defined as the size of packet. From Figure 8, d is time difference between the two replies received by the source i.e., t3 and t4 for packet A and B respectively. Furthermore, we work out the maximum burst for a connection by saying that if the packets get separated by d, then the maximum number of packets you can burst is (RTT/d). Therefore with this formula we calculate our burst size to be (S*RTT/d), where, S is the size of packet. And so we set the maximum unacknowledged packets to burst hence, when this value is reached, the sender should stop sending and wait for an acknowledgement because sending more data will likely result in packet loss.

TABLE III: FLAGS AND THEIR FUNCTIONS

BIT	NAME	FUNCTION
0	W_VAL	Window-Size is valid
1	ST_CKS	Checksum this packet
2	ST-RTR	Retransmission is permitted
3	ST_RET	Indicates a retransmitted packet
4	REMOTE_RESET	Connection reset by the other side
5	REPLY_REQ	A reply is requested
6	REPLY	Reply to a previous request
7	EOM	Last message was correctly received

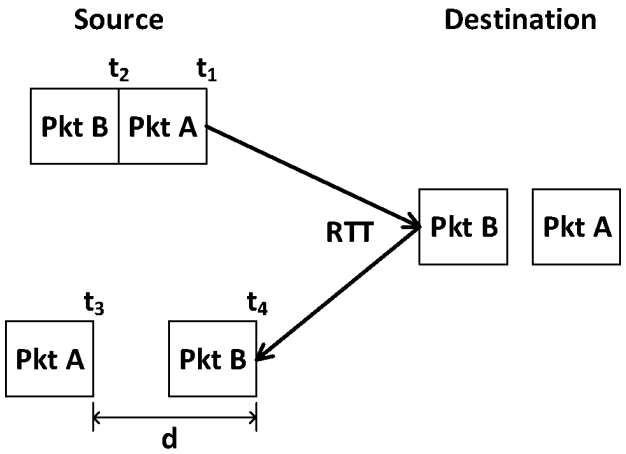


Fig. 8. SLTP bandwidth and burst size calculation.

F. Preliminary Result for SLTP

In order to fully analyse the effect of SLTP we obtained set of results that looked at the cost of communication between a client and server. This is directly dependent on the transport protocol being used. Results were obtained when the client and server are connected via an Ethernet Switch and when they are connected via a router.

Since SLTP runs over UDP, the size of a single SLTP packet can be up to (64KBs - 8 bytes (the size of the UDP)). However, for testing we wanted to ensure that SLTP packets could fit into a whole number of Ethernet packets which can carry a payload of 1500 bytes. For larger UDP packets we took into account IP fragmentation over Ethernet. So we varied this parameter as follows:

- SP 0 represents (1500 - (IP header size (20) + UDP header size (8) + SLTP Header Size (20))) = 1452 bytes
- SP 4 represents (SP 0 + (4 * (1500 - 20)) = 7372 bytes or 7.2 KBs
- SP 8 represents (SP 0 + (8 * (1500 - 20)) = 13292 bytes or 12.98 KBs

Finally, for these tests we used a window size of 144 KBs;

We performed our benchmarks by using two PCs equipped with the following hardware:

- Processor: Intel(R) Core(TM) i5-3770 CPU (4 cores).
- RAM: Both PC with 16GB DDR3
- Storage: Both PC with 320GB HDD
- Network: 1 Gigabit Ethernet Cable
- OS Type: Fedora 25 64-bit
- Router: CISCO 1941 Series
- Switch: NETGEAR Gigabit desktop switch GS108

Fig. 9 shows the results when the connection is going through an Ethernet Switch and Figure 10 shows the results when the connection is going through a router with the client and server on two different networks of the router. These results show that SLTP generally performed better than standard TCP. These results at least indicate that modern systems now have enough resources in terms of CPU, memory and networking to allow user-space protocols to run efficiently.

V. QOS-AWARE APPLICATION AND SERVICES

In order to build QoS-Aware applications and services, it is necessary to periodically measure the bandwidth between client and servers. This can be done using an intelligent agent which may not be scalable. The other option is to use a mechanism built into the transport networking system. For example, the ICMP system in IP is used to measure the round trip time between endpoints. However, it is very difficult for applications to make use of that mechanism. Since the proposed low latency protocol, SLTP, runs in user space it is much easier to use this mechanism as a gauge of how much QoS is on the connection. Given that it is possible to measure the QoS between the application and server, we can therefore we can look at efficient sever migration to reduce the latency between client and server. So it is possible to determine

the best possible place to run the server as shown in the VANET scenario in Fig. 11.

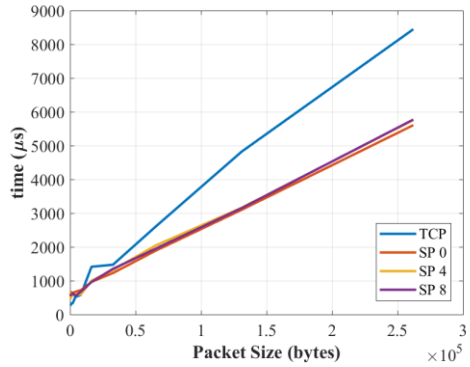


Fig. 9. Switch –network time: TCP vs SLTP packets size.

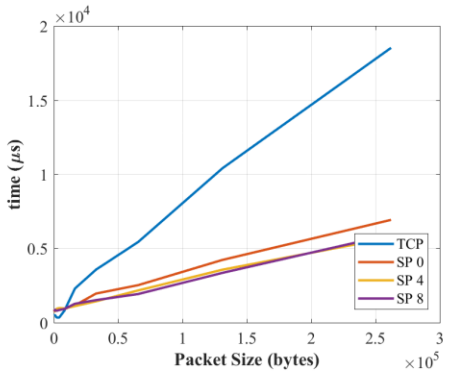


Fig. 10. Router –network time: TCP vs SLTP with different SLTP packet sizes.

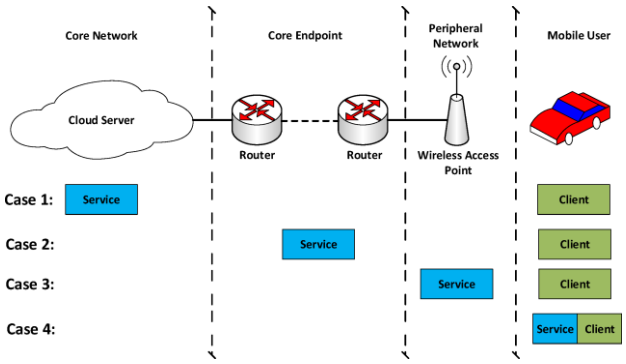


Fig. 11. Service migration scenario.

As explained in Y Comm architecture a service can be moved to three different locations namely: core network, core end point and Peripheral wireless network. In addition, a service can also be moved to the client. In order to facilitate real time applications running in highly mobile environments such as VANETs, it is necessary to very low latency. Hence we need to consider new approaches other than traditional cloud migration. Moving the server to different locations would reduce latency and yield better QoE [24]. For example, let us consider a scenario as shown in Fig. 11 where the cloud server is the core network, routers as our core end point, wireless access point as the peripheral network and the mobile user as the client. There are four possible cases as described below:

- Case 1: A mobile user is traveling in a car, the user experiences a service offered by the core network and the client receives it.
- Case 2: The service is running from the Core Endpoint therefore, the delay between the Core Endpoint and the client should be lower than the case 1.
- Case 3: The client and service are closer when measuring from the peripheral network or access point. And now the delay seems to be lower.
- Case 4: Here we are running services within the client.

The total time taken for a service to complete a given request can be split into two and they are Network Time (NT) and actual Service Time (ST). Here, NT is the time taken for a job sent from source to destination and back to the source. In essence, it is only the travel time of the job in a network. ST is the time a job is been serviced at the destination. Understanding of the NT and ST is very important to decide when and where the service has to be migrated to a network. For example, when a service is running in the cloud server as explained in Case 1, the NT is going to be large and ST is going to be small compared to other cases. This is due to the fact that the distance between cloud and the client is large, therefore, NT is large and cloud will have high computing power compared to a router or access point, therefore, ST is small. Hence, the ability to decide when and where to move the services based on NT and ST, which are in-turn based on user mobility, to ensure the best QoS for the mobile user is the focus of this research. The rise of fast computing processors and memory at a low cost allows these scenarios to be considered for a practical reason. Table IV below shows the possible outcomes of the considered location in the four cases in terms of NT and ST.

TABLE IV: SERVICE MIGRATION SCENARIO TABLE

Location	NT	ST
Server in Core Network	high	low
Router	high	medium
Access Point	medium	high
Mobile User	low	high

VI. EXPERIMENT SETUP

In order to completely investigate the support for QoS-aware applications. An implementation of Encryption as a Service (EnaaS) was implemented using SLTP, whereby a client or application can request EnaaS for a server to encrypt blocks of data before sending it over the network. Also, the server can decrypt the same block received over the network. The algorithm used for EnaaS is called XXTEA, which is being used in the Internet of Things (IoT). The server using SLTP is executed using the event mechanism and an upcall is used to process requests. The reading carried out for each test was done 7 times, the minimum and maximum values were rejected and the result was the average of the remaining five values. We got three sets of results: the cost of

communication; the cost of encoding the blocks and the cost of decoding the blocks. The results were based on the client and server on the same machine, client and server on two machines respectively connected via an Ethernet Switch connected to a Local Area Network (LAN) and client and server on two machines respectively connected via a Router.

Here, we used the same hardware configuration as was used for testing SLTP in Section IV. However, SLTP runs over UDP the size of a single SLTP packet can be up to (64KBs - 8 bytes (the size of the UDP)). However, for testing, we want an SLTP packet to fit into the payload of an Ethernet frame which is 1500 bytes. So the size of the SLTP packet used was (1500 - (IP header + UDP header + SLTP header)) which is equal to (1500 - (20 + 8 + 20)) = 1452 bytes.

The tests were sending commands to the EnaaS server to encode and decode packets of different sizes. There was also a NULL Command which was used to measure the Network Cost. So for any given size:

$$EncodeServiceTime = TotalEncodeTime - NULLTime$$

$$DecodeServiceTime = TotalDecodeTime - NULLTime$$

We passed different buffer sizes starting from 256 bytes then to the multiples of two till we got to 262144 bytes over the network. Using the NULL command, we measured the time it took each packet size to be transported from client to server then back again. This therefore represents the network cost entailed in any request which is NT.

Afterward the EnaaS server was asked to encode messages of different sizes. The total service time for a packet is the sum of NT and ST. Therefore, time taken for actual service which is ST is derived by subtracting the NT obtained for NULL service with the same packet size from the total service time.

While conducting experiments for server on the peripheral network, we repeated the same procedure as the first experiment but placed the client and server over two different machines while still connected over Local Area Network (LAN) with a gigabyte Ethernet cable.

VII. Results

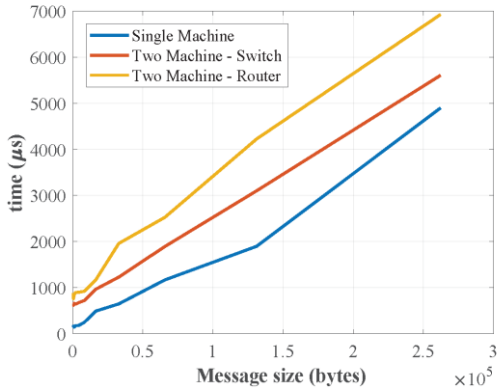


Fig. 12. Network time.

Fig. 12 shows the NT results for three different cases as explained in the previous section. The result shows that NT is significantly reduced when client and server are running on the same machine. In addition, NT is less for two machines connected via gigabit Ethernet switch compared to the router. We can also observe that the NT for all three cases increases as the massage size is increased.

In contrast, we can observe that Encode ST is high for two machines connected via a switch than the router and service running on the same machine as shown in Figure 13. The reason behind this outcome has to be explored in detail. Even in this case, ST for all three cases increases as the message size is increased.

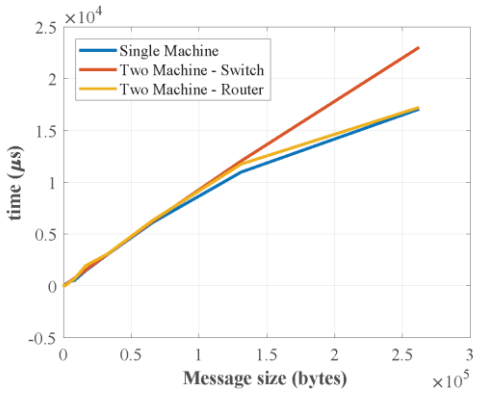


Fig. 13. Encode service time.

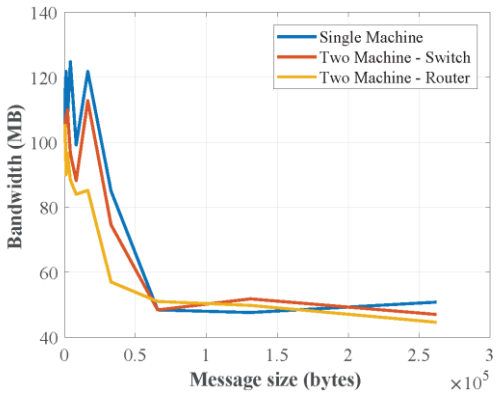


Fig. 14. Bandwidth measurement.

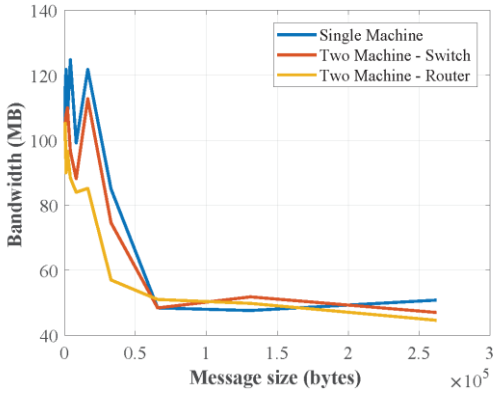


Fig. 15. Bandwidth received as a percentage of maximum capacity (1Gbit/s)

From Fig. 14 and 16, it is evident that the bandwidth and burst change significantly as the message size is increased. Bandwidth reaches its minimum (50MB/s) at 70,000 bytes (approx.) and stays in the same range as the message size is further increased.

Furthermore, as shown in Fig. 17, there is peak increase in the latency as packet size is increased for all the three cases until 70,000 bytes (approx.) and does not vary much as the packet size is increased further. This is due to the fact that the bandwidth does not vary much beyond that size. In addition, we can observe that latency for the router is higher compared to other two cases.

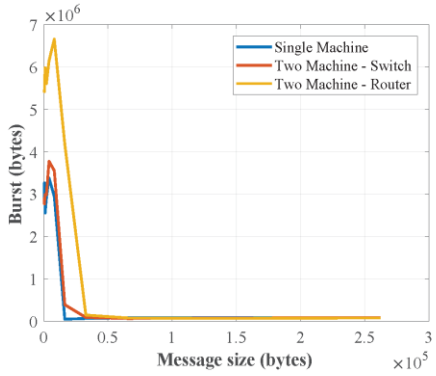


Fig. 16. Burst measurement.

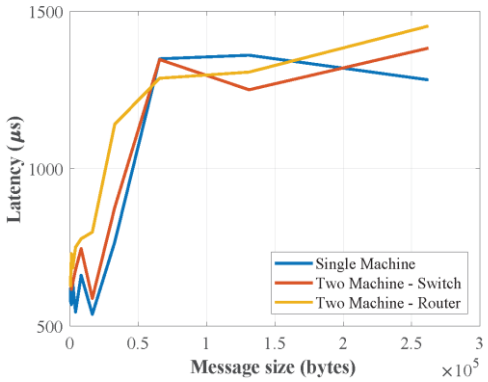


Fig. 17. Latency measurement

VIII. The Proposed Analytical Model for Cloud-Based Intelligent Service Migration

In this section, the proposed analytical model of Cloud-based intelligent service management is introduced considering the user mobility in heterogeneous environments. In order to optimize use of the network for mobile users, a novel queuing model is presented. This is achieved by end-to-end network slicing when the service localisation as well as the advanced handover are desirable.

A. The Proposed Model

In this section, the proposed system is described as analytical modelling approach to evaluate the QoS of Cloud-Based intelligent service migration using Markov chains. The proposed model is shown in Fig. 18. The mobile user's trajectory (AB) is clearly observed from Fig. 18. Hence, the mobile user moves across through point A to destination B. At the location A mobile user connected to core network, CN and there are several radio access networks, RAN (i.e., hotspots) between A and B such that the mobile user will always be connected. The main scenario taken here is the video application that user wants to use during its movement. Hence, the challenge here is to use intelligent service migration by handover techniques and server localisation to maintain a reasonable QoS. Fig. 19 shows the state diagram of the proposed system. The states of the system are described by i and j, specifying the *networks configuration* and number of *network-based requests* in the system, respectively.

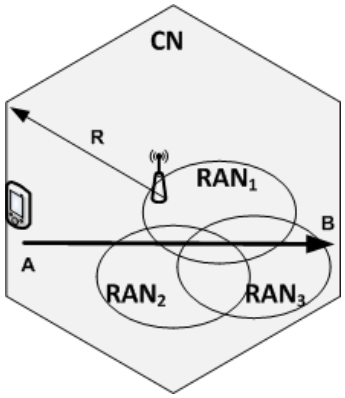


Fig. 18. Mobile user's trajectory for the proposed scenario.

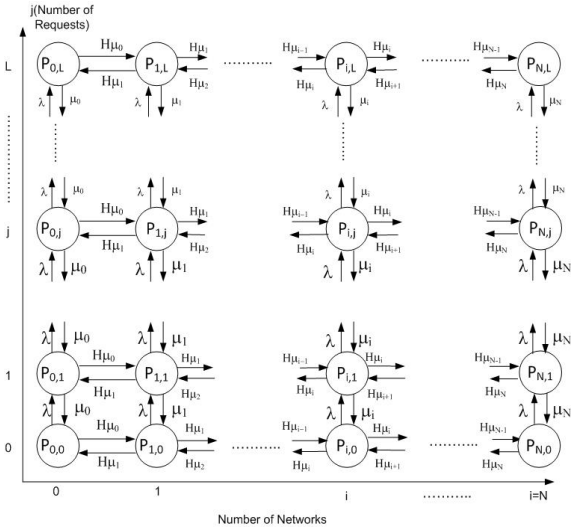


Fig. 19. State diagram of the proposed model.

Thus, $P_{i,j}$s are steady state probabilities of having i number of networks and j number of network-based requests in the system. In Figure 19 the downward transitions indicate that the requests are being served with service rate μ_i (i=0,1,. . ., N) which depends on the number of networks as well as the number of requests in the system. On the other hand, upward transitions take place because of new requests of applications with rate λ to the system. Please note that, the proposed system is an analytical model for the mobile user. The proposed model does not analyse the given network which will have many

users unlike the previous analytical models [33], [34], [35]. Hence, the service rate of the network as seen by the user will vary according to how many other users are using the network. Hence, this is based on observations by the mobile node or the service-oriented architecture which is managing the service.

The service rate is defined as the perceived rate of service responses that the mobile users are receiving. Hence, the service rates are the factor of θ multiplied by the total capacity of each network. Thus, θ is represented by the bandwidth test in Fig. 15. It is assumed that the service rate should be high enough so that it satisfies the request rate from the mobile users. However, the requests are being queued by the network and consequently, the response time increases when this condition is not satisfied. Thus, application performance degrades. The lateral transitions indicate mobility scenarios between networks. As the mobile users move between networks, each chain (column) represents the performance that the mobile users experiences at their location. The mobile user's movement can be expressed as a probability of hopping from one chain to another. H μ_i and H d_i, where (i=0,1,. . ., N), represent different probabilities for the mobile user to leave the network by moving to right-hand side and left-hand side, respectively to different network. In other words, H μ_i and H d_i are different handover rates from one network to another. Both handover rates are defined as the mobility rate and can be calculated as follows using the approach presented in previous studies [11], [35].

$$H\mu_i = Hd_i = \frac{E_i[v]P_i}{\pi A_i}, \quad (1)$$

$E_i[v]$ is the average speed of the mobile users, P_i and A_i are the length of the perimeter and the area of the corresponding networks, respectively.

IX. Markov Model of Proposed System

A. Solution of System of Linear Equations for the Steady State Probabilities

In this study, state probabilities are calculated by solving a system of linear equations. Considering the proposed model the advantage of the well-known system of balance equations is taken. A MATLAB package is used for solution with increased number of states. It is possible to represent the system of state probabilities in the form of Ax=B as shown in the matrices below, where A is a matrix of size nxn, x is a column vector of n unknowns, and B is a column vector of n values.

$$\begin{pmatrix} A_{0,0} & A_{0,1} & \cdots & A_{0,N.L} \\ A_{1,0} & A_{1,1} & \cdots & A_{1,N.L} \\ \vdots & \vdots & \vdots & \vdots \\ A_{N.L,0} & A_{N.L,1} & \cdots & A_{N.L,N.L} \end{pmatrix} \times \begin{pmatrix} x_0 \\ x_1 \\ \vdots \\ x_{N.L} \end{pmatrix} = \begin{pmatrix} B_0 \\ B_1 \\ \vdots \\ B_{N.L} \end{pmatrix}$$

In the proposed system, A is of size NxL. x is a column vector of unknown state probabilities (P_i) where i=0,1,. . .,NxL. *B* consists of the scalars in the balance equations. Redundancy is a problem amongst the global balance equations. Thus the additional information of the normalisation condition is needed. This problem is resolved by replacing one of the balance equations by the normalisation condition. Similarly *B* is a column vector with all zeros except from the last element which is 1. B_i vector is denoted, where B={0,0,. . .,0,1}. Hence the resulting equation can be expressed as follow:

$$A_i.P_i = B_i \text{ where } i=0,1,\ldots,NxL.$$

In steady state, $\pi_{i,j}$ is the proportion of time that the process spends in a state $x_{i,j}$. Recall that the transition rates are the instantaneous rates that the model makes a transition from a state $x_{i,j}$ to a state $x_i+1_{,j}$ or from a state $x_i+1_{,j}$ to a state $x_{i,j}$. When the model is in steady state, transition rates used to obtain the matrix A considering the balance equations are in an equilibrium. When all the steady state probabilities, $P_{i,j}$s, obtained, a number of steady-state performance measures can be easily computed. The mean queue length (MQL_i), throughput (γ_i) and mean response time (MRT_i) of all networks are computed respectively which can be obtained as follows:

$$MQLi = \sum_{i=0}^{N} i \sum_{j=0}^{L} Pi,j \quad (5)$$

$$\gamma i = \sum_{i=0}^{Ni} \mu i \sum_{j=0}^{L} Pi,j \quad (3)$$

$$MRT_i = \frac{MQL_i}{\gamma_i} \quad (4)$$

The proposed service management framework is used to evaluate much network slicing do mobile users need in a given location to satisfy application requirements.

X. A Case Study Using the Model to Analyse VOD Service

In this section, results are presented for the performance evaluation and optimization for VoD considering end-to-end network slicing in heterogeneous environments. The numerical study focuses on MRT of the proposed model. The mean service rates are mainly application dependent for each network.

A. Scenario-Based Study

The given scenarios are based on the mobile user movements. The mobile user is connected to core network, *C N* and goes from location A to destination B as shown in Figure 18. As it can be clearly seen from the figure, in the proposed scenario there are several radio access networks (hotspots) are assumed as RAN_1, RAN_2, and RAN_3. Hence, three wireless hotspots are considered in this paper in the city centre and they will be busy during rush hour. The user mobility pattern information can be provided based on a user's past mobility patterns which can be measured using the Wireless Footprint

method [11]. In this paper, streaming a video is primary task while a user is mobile. Hence, the requirements and modelling issues are considered for video applications. In addition, in terms of the QoS and the time the user spends in these networks may vary because of different cell sizes due to the heterogeneous environment considered. However, the proposed model can be adopted easily to different types of application as well as the traffic based on the network specifications. Hence, in order to stream a video without interruption in a mobile environment, the analysis done in [36] is followed to obtain optimal latencies for the service decision making process. Table V shows the latencies for video servicing what is needed to satisfy QoS demands by taking real-time measurements from networks.

TABLE V: MOBILE NETWORK LATENCIES RESULTS IN [36] FOR VIDEO STREAMING

T: Time (latencies)	Action
T < 40ms	Carry on using the same network
40ms ≤ T ≤ 80ms	Handover to better network if available
80ms < T	Switch the service closer to the user.

The user is moving in one direction and therefore there is a very small possibility of immediately returning to previous network for the considered scenario. A mobile node is connected to *C N*. It moves on a fixed-path through the destination B as shown in Fig. 18 while streaming a video. *C N* is assumed as LTE networks with larger coverage areas. On the other hand, there are three smaller overlapping networks between inside the core network which are represented RAN_1, RAN_2, and RAN_3, respectively. The mobile user will enter an overlapping area that is covered by smaller Wi-Fi networks. Then, the mobile user will exit the overlapping coverage area and will reach to destination. The system parameters used are mainly taken from [11], [35], based on the relevant literature [33]-[35]. In all cases, the mobile user's velocity is 5km/h for Wi-Fi and LTE networks, respectively. In addition, the LTE radius (R) is 1000 meters and the Wi-Fi radius (r) is 60 meters. To demonstrate how this model behaves in considered scenario, we study a case where the Wi-Fi networks have different service rates which keep changing according to traffic intensity of mobile users in the system for historical traffic over a day. In this study, the mobile user will get a share of the maximum capacity of the network to which the mobile user is currently attached. We denote that factor or slice of the network by the symbol, θ. Hence $\mu_i\theta$ gives the service rate actually experienced by the user in a given network, i. The considered scenario expresses a case where a mobile user may be handed-over to any available network or request that the service is moved closer to itself in order to stream a video without interruption during the day. It is assumed that the service rate should be high enough so that it satisfies the request rate from the mobile users. The service rates considered in all networks are taken from the real life scenarios in order to give realistic QoS measurements.

Table VI shows the starting service rates, departure rates and perceived probabilities, θ during the day for all networks considered in this scenario.

TABLE VI: SERVICE RATES, HANDOVER RATES AND Θ CONSIDERED FOR HISTORICAL TRAFFIC OVER A DAY

	μ(i)	Mμ(i)	Md(i)	Light,(θ)	Moderate, (θ)	Heavy,(θ)
CN	50	0.0088	0.0088	0.9	0.4	0.2
RAN_1	150	0.0147	0.0147	0.2	0.16	0.08
RAN_2	150	0.0147	0.0147	0.18	0.12	0.06
RAN_3	150	0.0147	0.0147	0.14	0.08	0.04

Assuming the average data rate of wireless hotspots networks is greater than that of the LTE network. Thus, it is assumed that RAN_1, RAN_2, and RAN_3 provide quicker service than *C N*. Thus, the *C N* has service rates as μ_1=50 tasks/sec. On the other hand, service rates of RAN_1, RAN_2, RAN_3 are taken as 150 tasks/sec. The outstanding network-based request rate is λ=40 tasks/sec. Our main focus is the inner area which is formed by the smaller networks. The queuing capacity is limited with L=50 which represent the maximum number of requests waiting for each network. The results will allow us to predict service decision-making process across multiple networks along a user's path.

TABLEVII: MEAN RESPONSE TIME RESULTS FOR HISTORICAL TRAFFIC OVER A DAY

Mean Response Time (ms)			
	LIGHT	MODERATE	HEAVY
CN	22.2	50.0	100
RAN_1	33.3	41.6	83.1
RAN_2	37.0	55.5	110.9
RAN_3	47.6	83.2	166.3

The mean response time results obtained from the proposed model is given in Table VII in milliseconds. From Table VII, we can derive Table VIII which is a decision table based on the requirements as detailed in Table V. From Table VIII, for the light context, we see that most of the options in that column, point to staying in the current network (C) because the traffic is light and hence the mobile user is given a largest slice in each network.

TABLE VIII: DECISION TABLE OF A USER FOR HISTORICAL TRAFFIC OVER A DAY

	LIGHT	MODERATE	HEAVY
CN	C	H	M
RAN_1	C	H	M
RAN_2	C	H	M
RAN_3	H	M	M

However, in the moderate context, the networks are busier and hence the recommendation from the decision table is to do a handover (H) to the next network as soon as possible. In the case of RAN_3, it is recommended that service be migrated (M) to the final destination, B. However, in the heavy traffic context, all the results are to migrate the service closer to the user for all networks. This shows that in heavy traffic the user's slice is smaller and this results in substantial service degradation. So, this means that small variations in network slicing can lead to very different options in order to maximize the network.

XI. Conclusion and Future Work

This paper has looked at a new framework for intelligent service migration in the Future Internet using SDN, NFV and SLTP, a prototype of a low latency transport protocol. This new framework will allow us to build mobile QoS-aware applications and services. This work is necessary to accommodate new networks, such as VANET and Haptic networks that require low latency and high bandwidth. The results suggest that this work can be used as a reference model to develop and build a real testbed to investigate the development of new applications and services.

References

[1] G. Mapp, F. Shaikh, J. Crowcroft, D. Cottingham, and J. Baliosian, "YComm: A global architecture for heterogeneous networking (Invited Paper)," in *Proc. 3rd Annual International Wireless Internet Conference (WICON)*, October 2007.

[2] Open Networking Foundation. Software-Defined Networking: The New Norm for Networks, April 2012.

[3] R. Stewart. Stream control transmission protocol, 2007.

[4] A. Ghosh, V. Vardhan, G. Mapp, O. Gemikonakli, and J. Loo, "Providing ubiquitous communication using road-side units in vanet systems: Unveiling the challenges," in *Proc. 13th International Conference on ITS Telecommunications*, Nov. 2013, pp. 74–79.

[5] Sun Microsystems, *NFS: Network File System Protocol Specification*, IETF, March 1989.

[6] D. R. McAuley, "Protocol design for high speed networks," PhD thesis. In Technical Report 186. Computer Laboratory, University of Cambridge, January 1990.

[7] T. W. Strayer, B. J. Dempsey, and A. C. Weaver, *The Xpress Transfer Protocol*, August 1992.

[8] G. E. Mapp, S. Pope, and A. Hopper, "The design and implementation of a high-speed user-space transport," in *Proc. Globecom 97*, Phoenix, Arizona, November 1997.

[9] T. Braun, C. Diot, A. Hoglander, and V. Roca, "An experimental user level implementation of TCP," in Technical Report 2650, INRIA Sophia Antiplois, France, 1995.

[10] K. Mansley, "User-Level TCP in High-Performance Server-Cluster Networks," PhD thesis. In Laboratory for Communication Engineering. Department for Engineering, University of Cambridge, December 2004.

[11] F. Sardis, "Exploring traffic and QoS management mechanisms to support mobile cloud computing using service localisation in heterogeneous environments," PhD thesis, 2014.

[12] S. S. M. Almeida and D. Corujo, "An end-to-end qos framework for 4g mobile heterogeneous environments," 2007.

[13] R. O. M. Kuroda and M. Yoshida, "Secure service and network framework for mobile Ethernet," 2004.

[14] ITU. Global Information Infrastructure, Internet Protocol Aspects and Next Generation Networks, y.140.1., Technical report, International Telecommunication Union, 2004.

[15] HoKey Working Group. Hokey: Security Support for IEEE 802.21: Media Independent Handover Services, 2010.

[16] N. Niebert, A. Schieder, A. Abramowicz, H. Malmgen, *et al.*, "Ambient networks: An architecture for communication networks beyond 3G," *IEEE Wireless Communications*, vol. 11, 2004.

[17] M. Aiash, G. Mapp, A. Lasebae, J. Loo, F. Sardis, R. Phan, M. Augusto, R. Vanni, and E. Moreira, "A survey of potential architectures for communication in heterogeneous networks," in *Proc. Wireless Telecommunications Symposium*, April 2012.

[18] M. Aiash, G. Mapp, A. Lasebae, J. Loo, and R. Phan, "A formally verified AKA protocol for vertical handover in heterogeneous environments using Casper/FDR," *EURASIP Journal on Wireless Communincations and Networking*, April 2012.

[19] M. Aiash, G. Mapp, A. Lasebae, R. Phan, and J. Loo, "Integrating mobility, quality-of-service and security in future mobile networks," *Electrical Engineering and Intelligent Systems: Lecture Notes in Electical Engineering*, vol. 130, pp. 195–206, 2013.

[20] N. Kamiyama, Y. Nakano, K. Shiomoto, G. Hasegawa, M. Murata, and H. Miyahara, "Analyzing effect of edge computing on reduction of web response time," in *Proc. IEEE Global Communications Conference*, Dec. 2016, pp. 1–6.

[21] K. Imagane, K. Kanai, J. Katto, and T. Tsuda, "Evaluation and analysis of system latency of edge computing for multimedia data processing," in *Proc. IEEE 5th Global Conference on Consumer Electronics*, Oct. 2016, pp. 1–2.

[22] K. Zhang, Y. Mao, S. Leng, A. Vinel, and Y. Zhang, "Delay constrained offloading for mobile edge computing in cloud-enabled vehicular networks," in *Proc. 8th International Workshop on Resilient Networks Design and Modeling*, Sept. 2016, pp. 288–294.

[23] B. P. Rimal, D. P. Van, and M. Maier, "Mobile edge computing empowered fiber-wireless access networks in the 5g era," *IEEE Communications Magazine*, vol. 55, no. 2, pp. 192–200, February 2017.

[24] Y. Yu, "Mobile edge computing towards 5g: Vision, recent progress, and open challenges," *China Communications*, vol. 13, no. 2, pp. 89–99, Nov. 2016.

[25] M. Femminella, M. Pergolesi, and G. Reali, "Performance evaluation of edge cloud computing system for big data applications," in *Proc. 5th IEEE International Conference on Cloud Networking*, Oct. 2016, pp. 170–175.

[26] F. Sardis, G. Mapp, J. Loo, M. Aiash, and A. Vinel, "On the investigation of cloud-based mobile media environments with service-populating and qos-aware

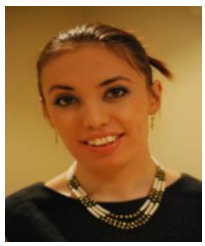
mechanisms," *IEEE Transactions on Multimedia*, vol. 15, no. 4, pp. 769–777, June 2013.
[27] H. Li, G. Shou, Y. Hu, and Z. Guo, "Mobile edge computing: Progress and challenges," in *Proc. 4th IEEE International Conference on Mobile Cloud Computing, Services, and Engineering*, March 2016, pp. 83–84.
[28] T. Wan and P. Ashwood, A Performance Study of CPRI Over Ethernet, IEEE1904, 2015.
[29] M Bjorklund, RFC 6020: YANG - A Data Modeling Language for the Network Configuration Protocol (NETCONF), IETF, October 2010.
[30] Solarlare. 10 Gbps NIC Cards for Sale.
[31] SlideShare. Understanding DPDK, February 2015.
[32] G. Mapp, D. Thakker, and D. Silcott, "The design of a storage architecture for mobile heterogeneous devices," ICNS2007, p. 41, 2007.
[33] D. Bruneo, "A stochastic model to investigate data center performance and qos in iaas cloud computing systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 3, pp. 560–569, March 2014.
[34] J. Vilaplana, F. Solsona, I. Teixid ́o, J. Mateo, F. Abella, and J. Rius, "A queuing theory model for cloud computing," *The Journal of Supercomputing*, vol. 69, no. 1, pp. 492–507, Jul. 2014.
[35] Y. Kirsal, E. Ever, A. Kocyigit, O. Gemikonakli, and G. Mapp, "Modelling and analysis of vertical handover in highly mobile environments," *The Journal of Supercomputing*, vol. 71, no. 12, pp. 4352–4380, Dec. 2015.
[36] G. Mapp, D. Thakker, and O. Gemikonakli, "Exploring gate-limited analytical models for high performance network storage servers," in *Proc. 18th International Conference on Computer Communications and Networks*, Aug. 2009, pp. 1–5.

Oneykachukwu Augustine Ezenwigbo received his B.Sc degree in industrial physics (Physics Electronics & Information Technology Applications) from Covenant University, Ogun state, Nigeria, in 2011 and his M.Sc degree in telecommunication engineering from Middlesex University, London, UK, in 2014. Currently, he is pursuing his Ph.D degree in intelligent service migration in 5G at the Department of Computer Science, Middlesex University. His research interest include quality of service, quality of experience, software defined networks, 5g networks and cloud computing. He is a member of the Y-Comm and Vehicular Ad-Hoc Network (VANET) research group at Middlesex University.

Yonal Kirsal received his B.Sc degree in electrical and electronic engineering from Eastern Mediterranean University (EMU), Fagamusta, Cyprus in 2006 and M.Sc degree in computer networks from Middlesex University, London, the UK in 2008 achieving high honours (equivalent to UK first class) and distinction, respectively. He received his Ph.D degree in computer and communication engineering from Middlesex University. The title of his Ph.D thesis is modelling and performance evaluation of wireless and mobile communication systems in heterogeneous environments. During his M.Sc and Ph.D studies, he was working in research areas related to wireless communication systems, performance evaluation and queue theory, discrete event simulation, network design and availability/reliability. He is currently a fulltime lecturer at European University of Lefke (EUL).

Vishnu Vardhan Paranthaman received his BTech degree in information technology from Rajalakshmi Engineering College, Chennai, India, in 2010 and his M.Sc degree in computer networks management from Middlesex University, London, UK, in 2013. Currently, he is pursuing his Ph.D degree in resource allocation for Next Generation Mobile Networks at the Department of Computer Science, Middlesex University. His research interest include intelligent transport systems (ITS), 5g networks and mobile cloud platforms. He is a member of the Y-Comm and Vehicular Ad-Hoc Network (VANET) research group at Middlesex University and has worked in projects funded by UK Department of Transport (DfT) between 2015 and 2016.

Glenford Mapp received his Ph.D from the Computer Laboratory, University of Cambridge in 1992. He then worked at AT&T Cambridge Laboratories for ten years before joining Middlesex University in London in 2003, where he is currently an associate professor. He was also a visiting research fellow at the Computer Laboratory between 2003 and 2010 where he worked on several projects. His primary expertise is in the development of new technologies for mobile and distributed systems. He has published over 100 papers in refereed journals and conferences. Glenford does research on Y-Comm, a new architecture for building future mobile networks and also is the head of the Intelligent Transportation Research Group at Middlesex University.



Ramona Trestian received the B.Eng. degree in telecommunications from the Electronics, Telecommunications and the Technology of Information Department, Technical University of Cluj Napoca, Romania in 2007, and the Ph.D degree from the School of Electronic Engineering, Dublin City University (DCU), Dublin, Ireland in 2012 for her research in adaptive multimedia systems and network selection mechanisms. She was with Dublin City University as an IBM/Irish research council exascale postdoctoral researcher, from December 2011 to August 2013. She is currently a senior lecturer with the Design Engineering and Mathematics Department, Faculty of Science and Technology, Middlesex University, London, UK. She has published in prestigious international conferences and journals and has three edited

books. Her research interests include mobile and wireless communications, multimedia streaming, user quality of experience, handover and network selection strategies, and software defined networks. She is a reviewer for international journals and conferences and a member of the IEEE young professionals, IEEE communications society and IEEE Broadcast technology society.