Faces in the Clouds: Long-Duration, Multi-User, Cloud-Assisted Video Conferencing

Richard G. Clegg‡, Raul Landa¶, *Member, IEEE,* David Griffin†, Miguel Rio†, *Member, IEEE,* Peter Hughes∗, Ian Kegel∗, Tim Stevens∗, Peter Pietzuch‡, *Member, IEEE,* and Doug Williams∗

***Abstract*—Multi-user video conferencing is a ubiquitous technology. Increasingly end-hosts in a conference are assisted by cloud-based servers that improve the quality of experience for end users. This paper evaluates the impact of strategies for placement of such servers on user experience and deployment cost. We consider scenarios based upon the Amazon EC2 infrastructure as well as future scenarios in which cloud instances can be located at a larger number of possible sites across the planet. We compare a number of possible strategies for choosing which cloud locations should host services and how traffic should route through them. Our study is driven by real data to create demand scenarios with realistic geographical user distributions and diurnal behaviour. We conclude that on the EC2 infrastructure a well chosen static selection of servers performs well but as more cloud locations are available a dynamic choice of servers becomes important.**

***Index Terms*—Streaming media, overlay networks, video conferences**

I. Introduction

Multi-user video conferencing is now extremely common. These systems are often enhanced by software within the cloud. Cloud routers instantiated at appropriate locations can improve the user experience by taking advantage of overlay networks and application-layer multicast (ALM) [1], [2]. A multi-server architecture for multi-user video chat enables flexibility in the choice of topology for transmitting video streams from one client to another. Multiple servers are used by popular systems such as Google Hangouts and Skype [3].

This paper deals with long-duration, multi-party video chat, that is conversations where users "hang out" in a chatroom with different participants joining and leaving throughout the duration of a session. There are several reasons to believe this will become more common: firstly, Skype, Google+ and Facebook offer the "hangout" ability in video chat; secondly, consumer-focused video conferencing apps often use capabilities such as WebRTC in the web browser, removing the need to install additional software and lowering the barriers to adoption; third, commercial providers have developed optimised routed video conferencing products, for example Bluejeans' cloud-based collaboration server and desktop video products from Vidyo and Polycom; and finally, a move toward network functions virtualisation (NFV) is expected to help service providers reduce costs.

‡Dept. of Computing, Imperial College London
†Dept. of Electronic and Electrical Engineering, University College London
¶Sky UK Network Services
∗ British Telecommunications PLC
Corresponding author E-mail: richard@richardclegg.org

There are a number of requirements for such a system. Users will join and leave in an ad hoc manner resulting in a change to the network topology. Participation will vary by time of day and hence, on a global scale, active users will migrate across different global locations "following the sun". As a chat progresses, the optimal locations for video servers may change. On the other hand, well-chosen initial locations for such servers may minimise this benefit.

The system used by Google Hangouts and Skype for video chat is that users connect to their nearest server [3]. Traffic from user A to user B is sent from user A to the server nearest to user A, then to the server nearest to user B and finally to user B (we refer to this routing policy as "StayOnRoute"). The amount of traffic sent to each user from the server will increase linearly with every additional user. However, if each user connects to a different server, the amount of traffic sent between servers scales as the square of the number of servers. Perhaps for this reason, both Google Hangouts and Skype limit the number of participants in a video call to ten at the time of writing. This scaling problem would be improved if the users connected to a smaller number of servers with the capability of dynamically migrating between sites. We invistigate this and other routing policies for video traffic routed via cloud servers.

We perform a data-driven investigation into long-duration video chat with many users when the location of the video routers can vary throughout the chat session. Assuming a fixed (possibly large) number of potential cloud locations, the problem arises of where to instantiate video routers. We consider the trade-offs between a *static* choice of router locations and a *dynamic* choice with routers instantiated and destroyed over the duration of the chat session. The problem addressed is thus:
Given a corpus of models that characterise a video chat system based on a multi-server, cloud-hosted architecture, choose for each chat session those cloud sites that should deploy and execute video routers to create an acceptable trade-off between quality of experience for the user and cost to the service provider.

We consider a number of hypotheses:

- A dynamic choice of video routers will improve QoE when compared with an intelligent static choice.
- Choosing from a larger set of potential locations will improve QoE.
- The cost per user and the QoE will not worsen dramatically as the number of users in the system grows.

- Difference in demand and usage patterns will have significant impact on QoE and cost, even when the underlying infrastructure is the same.
- The routing policy used can have a large effect on delays within the system.

In order to make our investigations realistic we draw on a large number of data sources to create plausible demand models for long-duration video chat scenarios. The two scenarios chosen are from gaming, multi-party video poker, and education, a chat room for a massive open online course (MOOC). We gather data about participation in online poker and MOOCs both in terms of geographic distribution and in terms of time of day. To model the underlying cloud servers, we consider a scenario in which the locations and charges are based on Amazon EC2 and a future scenario in which 2,507 locations are taken from a data set of current data centre locations. The modelling of delay is based on a global-scale delay measurement study [4].

Scenarios are investigated using demand data for the poker and MOOC scenarios with the cloud server locations using a simulation model. We model millions of users per day joining and leaving chat rooms across the globe according to a stochastic demand model derived from analysis of user data. We combine different strategies for server selection and routing. The modelling produces estimates for the cost and delay for each user of the system over the course of many simulated days.

To investigate these scenarios it is necessary to consider how traffic is placed on the overlay network between end users and cloud-hosts. The route that traffic takes between users in a given chat session is a product of two related decisions: first, a set of cloud locations where video routers will be instantiated is chosen; second, a decision must be taken on how to route the traffic via the instantiated video routers. For the choice of cloud locations, we use strategies based on clustering. For the dynamic strategy, the current set of users are clustered into N clusters based on their geographic location and the closest unused cloud location to each cluster centroid is chosen. For the static strategy, the same clustering algorithm is used but based upon weighted probable demand in locations rather than the actual observed demand.

Our investigation shows that, when only a few sites are available to choose from, a well-chosen static selection of routers performs nearly as well as dynamic router selection. When the number of cloud host locations increases a dynamic choice of routers gives more benefit. In most studied cases, the system scales well in terms of delay (as a proxy for QoE) and cost per user but in cases in which the mean size of a session increases, the cost per user also increases. If, in the near future, cloud hosts offer a larger choice of hosting locations, solutions allowing dynamic migration of video servers will provide considerable benefits for the QoE.

The article is structured as follows: §II puts the work in context with related research; §III describes a model for video routing software that can dynamically join or leave a chat session while minimising the impact on user QoE; §IV presents the mathematical modelling framework used to investigate the trade-offs in routing multi-client video via video routers; §V describes the usage scenarios selected; the results and discussion of their implications is given in §VI; and finally §VII gives conclusions and further work. Appendix §A gives details of the exact demand modelling.

To enable the reproducibility of the results, including all graphs in the article, our code and data is publicly available on Github.[1]

II. Related work

Trends in video conferencing Historically, the business market has driven the evolution of multi-party conferencing systems, with manufacturers such as Polycom and Cisco developing combined hardware and software solutions for high-quality conferencing. These systems typically used managed private networks and the conference terminals were connected through a central multi-point control unit (MCU), which mixed incoming audio and video streams to create a combined output received by all parties.

More recently, vendors have responded to the ubiquity of high-performance tablets and smartphones by offering desktop and mobile conferencing applications as a more flexible alternative, extending the reach of their platforms beyond traditional "room systems". The increasing capabilities of these clients, combined with improved broadband Internet connectivity, has led manufacturers to implement "routed" video conferencing solutions using application-level multicast (ALM) and overlay networks [1], [2]. In the cloud context, Amazon has developed CloudFront, which offers video services (including streaming) backed by a content distribution network. Multi-layer video streaming and layered video codecs such as H.264 SVC [5] are common within such products, and algorithms have been proposed for the adaptive selection of video layers based on user preferences [6]. However, so far dynamic adaptation techniques such as these are performed within existing default network paths and deployed in relatively static configurations.

An analysis of popular multi-party video chat services including iChat, Google Hangouts and Skype suggests that a multiple servers are preferable for conferences on the Internet but different architectures are used [3]. The study considers their performance in terms of loss and delay and, in particular, the relationship between these (i.e. increased loss leads to increased delay). Google Hangouts attempts to select a static configuration from a relatively small number of servers based on the locations of participants; Skype takes a hybrid approach by routing video and voice on different paths, video (for multi-user chat) travelling via servers and voice going directly between users.

Network function virtualisation Whilst video conferencing technology has been developing, the infrastructure upon which networks are implemented also appears to be ready to undergo significant change through the adoption of network function virtualisation (NFV). NFV enables the deployment of standard network functions on commodity hardware without the need for the installation of dedicated equipment. Industrial bodies such as ETSI have been central to its promotion and major

[1] https://github.com/richardclegg/multiuservideostream

network and service providers have announced both proof-of-concept demonstrators and their intention to procure architectures that depend on NFV [7], [8].

To date, NFV trials and implementations have focused on the virtualisation of network control mechanisms and content delivery networks, reflecting today's increasing use of network bandwidth for streaming video. In modern video conferencing systems where the majority of processing is carried out at the client, we believe that there is an opportunity for greater efficiency through the virtualisation of these "video routing" functions, allowing them to be deployed in the cloud and strategically positioned to improve the QoE of the users and reduce operational costs.

QoE and delay in video chat This article focuses on delay as a major component of QoE for conversational and interactive group video communications. In recent years, a considerable amount of research has been carried out to develop ways of defining and measuring QoE. Frameworks such as [9] and [10] propose models which account for factors relating to the system, the user and their context. However, when considering architectures for group-based video calling, end-to-end delay stands out as an important factor. Various studies [11], [12] have explored how delay impacts video communications and other interactive group sessions such as games [13], and ITU standards have discussed the impact of delay on multi-party communication [14]. We therefore believe that end-to-end delay is a credible proxy for QoE in the context of a comparison between multiple-server, cloud-hosted architectures. We also recognise the ability of a managed network to improve QoE by increasing throughput or reducing other system factors such as jitter and packet loss.

Early work analysing online video conferencing comes from [15] who considered a design goal of absolutely bounding delay end-to-end at 100 ms maximum and considered the design constraints this places on the system. In [16], the authors use a measurement study to compare immersive collaborative environments with video conferencing. They claim that 150 ms is the maximum tolerable delay in a multi-user conferencing system and that their measurements show that video conferencing falls within this delay bound in their measurements. However, other studies, such as [17] state that "longer delays are considered acceptable—up to 350 ms". In another context (that of cloud gaming) [18] looks at the effects of delay and jitter on QoE stating that in this context 80ms is an important delay threshold.

Critical to this article is the consideration of delays across the global network infrastructure. Our datasets are obtained from two measurement studies [4], [19] which look at a large scale dataset gathered using the domain name service (DNS) infrastructure, geolocation of DNS servers and measurement of the delays between them. This work is used in two different ways in this article: (1) to inform us on the likely density of the placement of Internet users within a country; and (2) to translate geographical locations to round-trip times.

The server selection problem The problem of selecting a subset of cloud sites to host video routers has several related problems in the area of operations research. Classical problems in service placement are the uncapacitated k-median (UKM) problem (where to place k facilities to best meet demand), or the uncapacitated facility location (UFL) problem (if the number of locations is also to be optimised). But these general settings tend to abstract away the details of the applications or services being considered and simply consider the distance between a static set of users and a set of potential sites for services.

Several studies investigate the relevancy and value of service components and information in distributed systems: [20] optimises the relevancy of coverage of sensors in composite services; while [21] models the dependency of the value of information on the quality of that information. Relevancy and value when mapped to our model of video conferencing relate to the suitability of locations to host cloud-based video routers to maximise the QoE of the delivered video sessions: modelled technically as the latency experienced between users and the proportion of the session's total network path routed over managed versus unmanaged network segments.

In [22], the authors model service placement as an optimisation problem. The paper considers placing new network elements trading off improved user experience with cost assuming various use cases including video. In [23], the authors investigate cloud-based video conferencing and have a real-world deployment over Amazon EC2. Their implementation (Airlift) has similarities with our video router architecture (see §III) and their routing strategy is analogous to our "NearestOn" strategy (see §IV-E) but without the possibility of migrating routers during a call. Airlift seeks to optimise throughput for users within a given delay bound. In [24], the authors model service placement between geographically-distributed clouds. They focus on dynamic pricing, moving instance placements according to price fluctuations. In [25], the authors look at the slightly different problem of live streaming with multiple sources again focusing on dynamic pricing.

In [26], the UKM and UFL problems are considered in a decentralised manner, allowing approximate solutions to be found scalably. Finally, [27] introduces CloudOpt that uses multi-criteria optimisation to consider diverse goals including service level agreements, memory requirements and availability. Our service placement formulation differs from many discussed in the literature because we consider the idea of services moving during their lifetime. Instead of looking for the best possible optimisation, we consider a "good enough" heuristic because repeatedly dynamically solving a complex optimisation problem is unlikely to be feasible in practice.

III. Implementing dynamic video routing

A basic component of our assumed model is a software component that we call a "video router", which has the following properties:

1) It can act both as a *sink*, allowing cameras or other video routers to send video streams to it, and as a *source*, allowing destination endpoints or other video routers to receive streams from it.
2) It can replicate streams, i.e. taking a single input and producing several output copies.
3) It does not add any perceptible delay or degradation to the video streams passing through it.

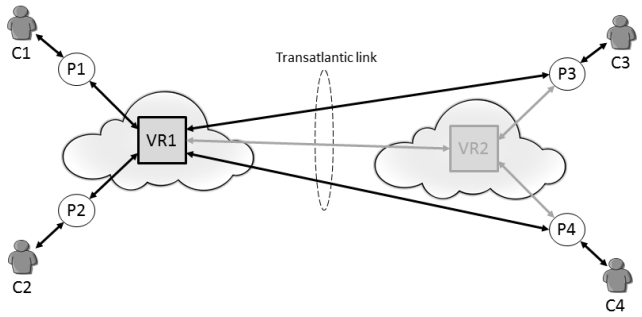


Fig. 1. Dynamic instantiation of a new video router in a live video conference

4) It can switch sources and destinations under external software control, and report the "connection map".[2]

The implementation of a multi-server cloud-based video conferencing system that can dynamically switch streams between cloud server locations with minimal impact on QoE is discussed in the next section. We base our model on the Vconect project [28], which has developed orchestration [29] and composition [17] functions for real-time high-quality audio-video communications between ad-hoc groups of people, and implements all four of the above features. The lightweight, dynamic nature of this video router goes beyond what is currently offered by conventional commercial video conferencing systems, and provides a basis for the modelling and simulation of multi-server architectures, as described here. Although our results are described in relation to an arbitrary routed video conferencing architecture, they could equally be applied to other popular multi-user video services such as Google Hangouts.

A. An architecture for dynamic migration of video routers

Figure 1 shows the QoE preserving dynamic migration strategy implemented by Vconect. The solution uses proxy routing components co-located with the client endpoints to ensure uninterrupted communications between clients. The figure demonstrates how dynamic configuration might be applied where a transatlantic link incurs significant costs. Initially, the clients C_1 and C_2 are on the same side of the link, with C_3 at the far end. In this situation, a single server VR_1 is best hosted in a cloud location close to C_1 and C_2.

At some point, however, C_4 joins the chat session. This creates an incentive to add a second video router, VR_2, in a cloud location on the other side of the link, to avoid sending traffic from C_3 and C_4 over the transatlantic link twice. First of all, the external control logic instructs proxies P_3 and P_4 to send duplicate outgoing streams to VR_2. It then connects VR_1 to VR_2, thus establishing a duplicate set of paths between VR_1 and clients C_3 and C_4. Next, VR_1, P_3 and P_4 switch their incoming streams to those from VR_2. Finally, the original connections between VR_1 and P_3 and P_4 are disconnected. If, at a later stage, client C_2 decides to leave the chat session, server VR_1 would no longer be required. Using a similar procedure, the external control logic could establish a duplicate stream path between C_1 and VR_2 and subsequently disconnect and shut down VR_1.

In the event that a new network path has a significantly different end-to-end delay, the act of performing the switch in VR_1, P_3 and P_4 might still result in a visible glitch in communication for the users. A "time-stretching" technique [30] can be deployed in each video router to mask the effect of switching between paths of different delays. In the event that a switch is being made from a path with a long delay to a path with a significantly shorter delay, the time-stretching technique introduces a delay buffer in the target stream, performing the switch and then gradually reducing the size of the buffer to zero. During the process of reducing the buffer, the client will receive a stream that is slightly faster than normal and once the buffer delay reaches zero the stream will return to normal speed. For small delay differences, this difference can be imperceptible.

By inspecting packets to determine I-frame boundaries within H.264-encoded video streams and using a time-stretching technique where appropriate, it is possible to achieve a virtually-unnoticeable switch between two cloud-based video router topologies. A small (18 participant) subjective test was performed to explore user preferences to different switching techniques, using video samples recorded from the Vconect platform. This test showed with a high confidence ($< 0.1\%$) that switching on I-frame boundaries was preferred to tearing down and re-establishing the connection [31].

In a real-world deployment, a video conferencing system would have to tolerate failures of one or more video servers or associated network links without affecting the provided service. The topic of fault-tolerance mechanisms to achieve this is orthogonal to the focus of this study—in particular, existing techniques for passive and active replication of servers could be employed, thus masking individual server failures. For such techniques, the dynamic switching capability described in this work could help make the system more resilient to failures by offering automatic redundancy to clients.

With the knowledge that it is technically possible to implement dynamic video routing between cloud locations, our study seeks to identify realistic scenarios in which this dynamic strategy could deliver significant benefits when compared with a conventional approach in which cloud locations for video routers are statically defined.

IV. Mathematical modelling framework

This section describes the mathematical model used to investigate multi-server architectures for videoconferencing. The model has several components, and its overall aim is to create a simulation with the following elements:

1) A realistic distribution of users across the earth.
2) A diurnal component so that user habits vary with their local day/night cycle and the "busy" period moves around the earth.
3) A measure of the reduced QoE that users perceive if they chat to people geographically separated.

In this model, detailed link and node level network metrics are abstracted by wide-area measurement studies characterising end-to-end performance between latitude/longitude pairs.

[2]One of the strategies investigated in this article dynamically switches video routers during a video chat session, and this requires the fourth property.

The modelling framework involves several components:

1) Demand model – When and where do users join the system and when do they leave?
2) Session model – How do individual users assign themselves to chat sessions?
3) Network model – Assuming that video routers are hosted in data centres around world, where are these data centres located and what prices are paid to use them?
4) Host location model – Choose cloud sites at which to host servers given sessions with users.
5) Routing model – Given a number of communicating users and allowed data centres, how will traffic be routed between the users?
6) Quality of Experience (QoE) model – Given a set of users and a set of routes for the traffic between them, what QoE will the users experience?

These components are obviously inter-related. For example, the session model fundamentally changes the cost model (if more people are in each session, costs become higher because each video stream is sent to more people). For the case of the demand and session models, we use two separate scenarios:

- Video poker scenario – in this scenario, contestants from around the world play poker using a video conferencing application.
- Massive open online course (MOOC) chat scenario – in this scenario, users of an online course collaborate to talk about course related topics.

We use these scenarios not because of interest in the scenarios themselves but rather to generate realistic demand for video streams with multiple chat participants. The scenarios are described in more detail in §V.

The work loop for the simulation is described by the pseudocode in Listing 1. This shows how all the six sub-models previously described operate together. The getNewUser(time) function uses the properties of the Poisson processes within the demand model to calculate when and where the next arrival and departure will be. The main work loop of the simulation is simply adding or removing users and updating the sessions and statistics accordingly.

A. Demand model

The demand model describes how users enter and leave the system, and it has two components: the location and time of an arrival and the subsequent departure time. It makes some simplifying assumptions:

1) Arrivals can be modelled as a series of processes that are Poisson but with a rate that varies diurnally (see [32], Table 3). Many such processes are placed at fixed geographical locations.
2) Departures are modelled by assuming a lognormal distribution of session times. That is, an arrival stays for a time with a lognormal distribution. These session times are assumed independent of the arrival characteristics. Lognormal distributions are extremely common in call durations [32].

```
users= [] events= [] sessions= [] time= 0;
#Use the demand model to get a new user;
user= getNewUser(time);
events.add(userArriveEvent(user));
events.add(userDepartEvent(user));
while true do
    firstEvent= events.popFirst();
    time= thisEvent.getTime();
    #Choose hosts for sessions with host selection model;
    hosts= calculateHosts(sessions);
    #Choose routes given sessions and hosts;
    routes= getRoutes(sessions,hosts);
    #Use network and QoE models – costs and delays;
    updateQoECostStatistics(routes,time);
    if time > SIMULATION_END then
        break;
    if firstEvent.getType() == ARRIVAL then
        users.add(event.getUser());
        user= getNewUser(time);
        events.add(userArriveEvent(user));
        events.add(userDepartEvent(user));
    else
        users.del(event.getUser());
    end
    #Use session model to group users;
    sessions= assignUsersToSessions(users);
end
printQoEStatistics();
```

Algorithm 1: Pseudocode for simulation loop

3) The arrival process is further broken down into a geographical and a temporal component, and these are independent.

As an output, this model produces arrival and departure times with associated latitudes and longitudes.

We assume that N locations on the Earth's surface are specified in terms of latitude and longitude and also an associated rate multiplier so that each location has a triple (x_i, y_i, l_i) where x_i is the latitude y_i is the longitude and l_i is the rate multiplier for that location. Further, we assume a daily periodicity where a day is split into M equal time units and each time unit has an associated rate multiplier h_i (this will be referred to as "hour" although any time period can be used). Let $h(t)$ be the rate multiplier associated with time t. Finally, let λ be the unmodified rate of arrival (without multipliers). So the rate of arrival at the ith location during the jth time period is given by $l_i h_j \lambda$. For the jth time period, the total arrival rate is thus $\sum_{i=1}^{N} l_i h_j \lambda$.

The demand model for a scenario is completely specified by the following information:

- A set of N triples (x_i, y_i, l_i) that specify a location and the rate of arrivals at that location (relative to other locations).
- A set of M values h_i that specify the rate of arrival for each hour i relative to the other hours.

- A multiplier λ that tunes the overall arrival rate.[3]
- A pair (μ, σ^2) that specifies the mean and standard deviation of the lognormal duration for a user's stay.

B. Session model

This is a scenario specific model that assigns users to *sessions*. These sessions represent users which are considered to be communicating with each other. Only users in the same session exchange data. In the session model, an arriving user has three options:

- join an existing session;
- start a new session; and
- join a waiting room before joining an existing or new session later.

Users stay in the same session until it ends. The poker and MOOC scenarios use different session models (see §V).

C. Network model

The network model defines the number and location of the video routers that users connect to and the price paid by the users for access to the servers that host them. To communicate with another user, the video must stream via at least one router (termed *server*). The main network model used is based upon the Amazon EC2 cloud offering as of late 2016.

The following table shows the locations and traffic and instance costs. Traffic costs are for traffic leaving the network (EC2 does not charge for traffic entering the network). The other cost is an instance cost which is shown for a c3.8xlarge instance, the cheapest instance with 10Gb networking available. Amazon also charges $0.01 per GB for traffic going between EC2 instance locations.

Name	Traffic ($/GB)	Cost ($/hr)
Ashburn (US East 1)	0.05	1.68
Palo Alto (US West 1)	0.05	1.68
Oregon (US West 2)	0.05	1.68
Dublin (EU West 1)	0.05	1.912
Frankfurt (EU Central 1)	0.05	2.064
Singapore (AP SE 1)	0.08	2.117
Sydney (AP SE 2)	0.12	2.117
Tokyo (AP NE 1)	0.12	2.043
Seoul (AP NE 2)	0.108	1.91
São Paolo (SA 1)	0.19	2.6

Our usage scenarios assume that each user can send video streams in one of two formats: a full stream at 2 Mb/s (the stream currently in focus) and a low-quality (thumbnail) stream at 0.125 Mb/s. During each session, each user sends in both full and thumbnail format. The video router forwards a single full stream from the user with "focus", modelling the screen layout typically used by existing services such as Google Hangouts. We make the simplifying approximation that each user has an equal share of "focus" during their time in a session. Compute costs are calculated by assuming that an Amazon c3.8xlarge instance can host video routers until its bandwidth of 10Gb/s is exhausted. In reality, of course, some overhead would be necessary. However, the compute costs are a tiny part of the total cost and the extra cost would be minimal.

An alternative scenario is also considered, using a hypothetical cloud service which offers a much larger number of data centres. In this model, 2,507 data centres with their latitude and longitude are used. These data centres are taken from http://www.datacentermap.com. No charging information was available for this scenario. Charges are a complex function of the policy of the organisation setting the charges and local conditions and regulations. In order to reflect the geographic nature of these charges we have created artificial prices for the 2,507 data centre model by taking the prices of the three geographically nearest EC2 data centres and weighting by the inverse of the distance to them. So, for example, if a data centre is almost exactly in the same location as an EC2 data centre it will have almost the same prices but if one is equidistance between three it will have the average of those.

D. Host location model

The host location model selects which cloud locations will be used by a session to host servers. Sessions are free to choose different sites from each other. A maximum number of cloud hosting sites S is set and no session may have services in more than S sites. Three methods are considered:

- random – cloud-host locations are static and randomly allocated;
- dynamic – the users in the session are clustered into S clusters. Each cluster (from largest to smallest) chooses an available hosting location closest to the cluster centroid; and
- static – The 1,000 locations (x_i, y_i, l_i) with the largest demand (l_i) are clustered into S clusters weighted by the demand l_i. Each cluster chooses the server nearest its centroid.

The random algorithm is a straw man showing what a badly performing algorithm would look like. The static algorithm is an intelligent algorithm but it always makes the same choice, based upon a one-off clustering. The dynamic algorithm is rerun every time the users in a session change and takes account of the minute-by-minute changes in a session.

By way of example, Figure 2 shows the clustering as would be used by the dynamic server selection model run for a session with twenty five users and three servers to be selected. The users are selected randomly but using the Poker scenario (see §V-A). This scenario has a lot of demand in North America and Europe. The coloured dots represent users in a given cluster. In this case, the algorithm generates three clusters, one centred around north America (green), one centred around Europe (black) and one in east Asia (blue) containing only a single user. The triangles represent the cluster centroid for that cluster and the diamonds the selected server. As a result of the clustering the algorithm has then selected the US west coast, Dublin and Tokyo servers—this appears to be the correct clustering and choice of servers for the user set chosen.

[3]This is technically redundant as multiplying all of the l_i or all of the h_i would serve the same purpose, but it is useful to have a single parameter to tune the arrival rate.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCC.2017.2680440, IEEE Transactions on Cloud Computing

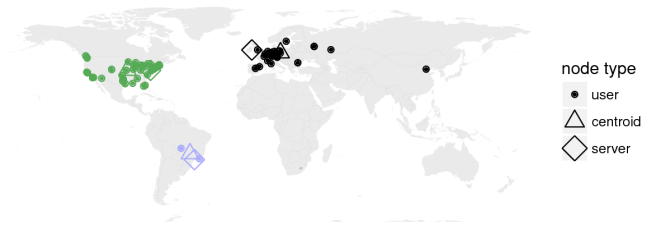


Fig. 2. Typical clustering to get 3 servers for 25 users

Fig. 3. Routing of traffic for the StayOnRoute model with a session of 12 users

E. Routing model

The routing model for each session takes the users in the session (from the demand and session models) and the allowed servers (from the server and network model) and chooses the route which traffic will take between each pair of users in a session. Three models are used, considering traffic from user A to user B:

1) StayOnRoute – the traffic goes from A to the server nearest to A then to the server nearest to B and then to B (sometimes this will only involve one server if the nearest server to A and B is the same server).
2) HotPotato – the traffic goes to the server that minimises the delay along the route (A, server, B).
3) NearestOn – the traffic goes to the server nearest (in terms of delay) to A and then directly to B.

Each strategy has advantages and disadvantages: StayOnRoute is built with the idea that bandwidth between data centres is more likely to be well-provisioned than the general Internet and therefore users will experience less loss of quality if the traffic spends as much time as possible on this managed network (see §IV-F); HotPotato clearly provides the minimum delay for the session considering the users in the session and the servers available. However, HotPotato is not a realistic model except where there are very few sites in use since the user has to send many copies of their high-quality video stream; and NearestOn is an attempt to make HotPotato more realistic while still keeping delay low. Since users always upload to the same server the use of their upload bandwidth is reduced.

Figure 3 shows the routes taken for a session with twelve users (some are in similar locations); the routes are curves indicating the great circle distance. The thickness of line indicates the data rate required. Black lines are between servers and red lines between users and servers.

F. Quality of Experience (QoE) model

The QoE model measures the users' experience of video chat. QoE is a complex function of many variables as discussed in §II. One of the most critical components of QoE for interactive video sessions is the latency between users: while throughput and loss have an impact on the quality of video sessions a fundamental metric which can impair interactive sessions between groups of people is latency [13]. Hence we focus on the delays between users. We use two methods to calculate the latency between two users specified by latitude and longitude both of which are based upon a large measurement study of actual experienced delays:

1) As a linear modification of the Haversine (great circle) distance around the surface of the Earth—that is the shortest distance between two points around the planet's surface. The linear model is parameterised by comparison against real data, as described in [4]
2) As a modification to the Haversine distance formula that accounts for the fact that the relationship between distance and delay differs by region [4].

These models give realistic data-driven estimates of the delay between points on the Earth's surface. We also break down the delay into components for the "unmanaged" network (i.e. the network that connects the user to the first server) and the "managed" network (i.e. the network between servers). It may be the case that the network between servers is of higher average quality than the general Internet. In particular, for example, when considering Amazon EC2 servers, the bandwidth between data centres is controlled by Amazon. Hence a user's QoE is also improved if the major proportion of the path between users and video routers is over managed rather than unmanaged network (for example, by having lower jitter, packet loss or increased throughput). In [33] the authors measure intra cloud bandwidth for several cloud providers and find TCP throughput varies from mean levels of 70Mb/s up to 300Mb/s. The lowest level available is considerably more than the requirements for the system described here and, since the paper is from 2010, the provided bandwidth has likely increased since then.

V. Evaluation Scenarios

As previously stated, the model is investigated in the context of two scenarios, one based on online gaming and one based on MOOC. The scenarios are chosen as motivating studies from which a reasonable demand profile can be derived. The aim in selecting the scenarios is:

- To investigate realistic demand profiles that may arise and how demand may change in distribution through time.
- To ensure that different scenarios provide some coverage of different traffic patterns that could arise.
- To stimulate investigation into how realistic demand distributions affect provisioning.

The details of how the provided input data becomes the model parameters used is given in Appendix A.

A. The video poker scenario

The poker scenario simulates a hypothetical online poker game in which people from across the world play poker with an accompanying video stream. Note again that the

main interest here is not the application itself; it is just a vehicle for obtaining an assumed distribution of users in sessions distributed across the world in a manner that changes realistically in time.

For the demand model, [34] provides an exhaustive survey and gives the proportion of online poker players in each country. This was used, as described in the previous section, to give the triples (x_i, y_i, l_i).

The duration of users staying in the system was drawn from a study of online poker players [35], which gives the average session length as 50.27 minutes and the standard deviation as 37.76 minutes. These figures are used with a lognormal distribution and player session lengths are drawn from this. The time of day behaviour was taken from a plot of the number of users on the US based poker site PokerScout—the daily number of users from 17th December 2013 was used.

The base case for the poker model was an average of 10,000 players per day arriving as described earlier, with session model parameters of a minimum of 4 and maximum of 10 players per table. Arriving players are assigned randomly between active sessions with fewer than 10 players. If all tables are occupied, arriving users join a waiting list until a slot becomes available at an existing table or until there are 4 players in the waiting room, when a new session will start. If a session has fewer than 4 players, the remaining users will join the waiting list.

B. The massive open online course (MOOC) scenario

The MOOC scenario models students discussing an online course over a video chat. MOOCs typically put up lecture videos in blocks for a week and then allow discussion on forums (where video chats are sometimes informally organised). This scenario models the forum chat extending to include a video chat room. The session model used in the simulations was a single global session with an average of 1,000 student arrivals per day. A lower base rate of arrivals than the Poker scenario was used to avoid unrealistically large session sizes at peak hours.

MOOC users per country are established from existing surveys [36]–[43]. The information from each survey gives the proportion of MOOC users by country for the top N countries and the rest of the users as "other". The surveys are combined with their information weighted by the number of users surveyed. The "other" is distributed over the remaining countries proportionally to their Internet population in such a way as to ensure they would not have made the top N list.

For session duration, the shape of the distribution was taken from British Telecom (BT) call duration data that was made available for this research. The data was an excellent fit to a lognormal distribution with mean 1.0 minutes and standard deviation 0.8 minutes. Many telephone calls are short and therefore the distribution was scaled to a mean of 27 minutes which has been reported [44] as the mean duration of Skype video calls. Due to uncertainties about that mean duration, the MOOC scenario was tested with a variety of session durations.

C. Scenario comparisons

Figure 4 shows the locations and rates for the Poker and MOOC demand models. The red and blue shapes represents demand for MOOC (red square) or Poker (blue circle). The size of the shape represents the relative size of the demand. So, large demand can be seen in the east and west coast of the US and some cities in Europe for both Poker and MOOC. Large demand for MOOC but not Poker can be seen in some east and south Asian cities.

The static server selection policy selects n servers by creating n clusters over all locations with demand weighted by the amount of demand for that scenario. This gives the servers that will always be selected for the static policy if n servers are required.

VI. Experimental results

To simplify presentation, results are considered against a base case scenario:

- Ten possible cloud host locations, the locations of Amazon EC2 servers charging Amazon EC2 pricing for data.
- The StayOnRoute routing model – data from user A to user B goes from A to A's nearest selected server, to B's nearest selected server and then to B.
- Delays modelled using the Haversine distance as a basis.

Each experimental setting is run for 100 (simulated) days. The results for quantities of interest are taken at the end of each day. The mean and standard deviation of daily results are then calculated. Graphs are plotted to show the mean with error bars at plus and minus 1.96 times the standard deviation. So for a quantity X of interest, the plotted data point is the mean of $X_1, X_2, \ldots, X_{100}$ for the observed days. Each of the X_i are themselves typically means measured over the day. As the number of samples is large, the central limit theorem applies, and X should have an approximately normal distribution. Let σ_X^2 be the standard deviation of the X_i. If X is normally distributed then $\pm 1.96\sigma_X^2$ forms a 95% confidence interval.

Sometimes the error bars are too close together to be easily seen, but for consistency they have always been plotted. For error bars not to overlap, points are shifted slightly right and left on the x-axis by different amounts for each line. The two measures used in most plots are the mean delay per user pair (measured over all pairs of users in all sessions, weighted by the amount of time that pair spends in the system) and the mean cost per user hour (the mean cost a user would incur in charges for one hour in the system). Delay is broken down into total delay and delay on unmanaged network (the network between the user and the cloud server instance). Cost is broken down into the total cost and the cost on managed network (between server instances). The cost spent on purchasing instances is so small as to be negligible in all cases and hence is not shown.

A. Scalability with respect to number of host locations used

This section considers how the system changes as the number of cloud host sites used per session increases from just one to using all ten (in the case of Amazon EC2, ten

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCC.2017.2680440, IEEE Transactions on Cloud Computing

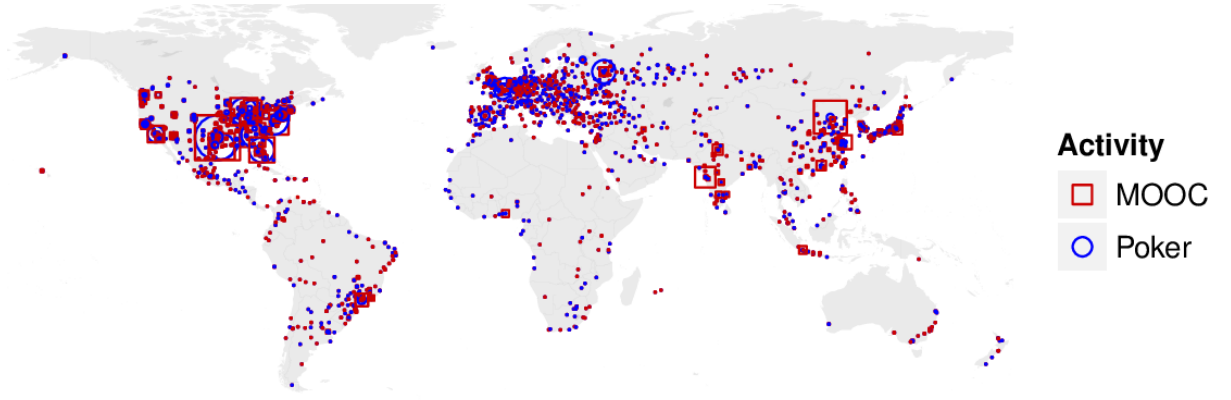
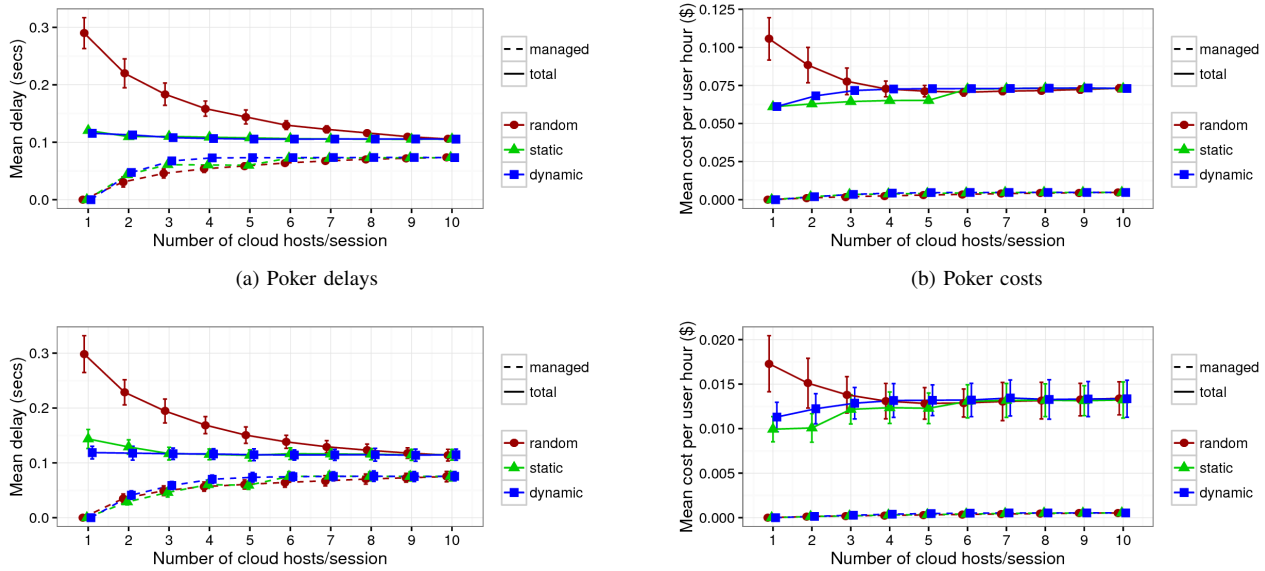


Fig. 4. Demand for the poker and the MOOC scenarios



(a) Poker delays

(b) Poker costs

(c) MOOC delays

(d) MOOC costs

Fig. 5. Results for the two scenarios varying the number of cloud sites available per session

means using all ten locations in which case all systems are equivalent as all use all possible locations). Figure 5a shows the delay experienced for traffic between a typical user pair. The three host selection policies are compared: dynamic, static and random. The blue squares represent dynamic, the green triangles static and the red circles represent random. The solid line represents the total delay and the dotted line represents the delay over the managed network. In the case where only one host is selected, no part of the path is over the managed network, hence the delay on the managed network is zero.

In terms of total delay, it is surprising that in this scenario the performance of dynamic and static looks similar. This is a result of the fact that for the user profiles for Poker, most of the users are in the US and Europe. So the static policy (always selecting the US East Coast server first) is never particularly bad. It can be seen that the dynamic model has a slight but statistically significant decrease in time spent on unmanaged network when between three and five cloud sites are being used.

The cost model shows the particular distortions caused by the EC2 pricing model. For the poker model, this can be seen in Figure 5b. The cost is influenced in two ways: first by using the more expensive host locations; and second by encouraging more traffic onto the managed network. The figure shows a combination of these effects. The random scenario has costs that are high (and extremely variable) because often it chooses an expensive host location even though these are not efficient. The static model, by contrast, never picks the more expensive locations even when those would reduce delay. In the Poker scenario, the static model only makes use of the most expensive servers when more hosts locations are being used. The cost on managed network is a trivial component of the total cost (but this is because managed network is priced relatively low). Surprisingly, in this case the static model based upon a good prediction of demand could be considered to have advantages over the dynamic model (that did not greatly improve QoE but was more expensive).

Figure 5c shows delay used for MOOC. Here, compared with Poker, the demand is more globally diverse with a large demand in East Asia. The error bars are larger for the MOOC

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCC.2017.2680440, IEEE Transactions on Cloud Computing

model because the number of users is smaller (1,000 per day as opposed to 10,000). The random model is better at reducing the time on unmanaged network than was the case for the Poker scenario, but this is more than offset by the extremely large total delay on the network for the random model (due to the more geographically dispersed user base). The main difference in performance is that static model appears worse for just one server (but this difference is within the error limits). However, it is worth emphasising that the mean delay for both Poker and MOOC for all server selection models is similar. MOOC has a higher overall delay than the Poker scenario when all ten sites are used, and this is likely to be due to its higher geographical diversity.

For costs, the story is rather similar to that with Poker as seen in Figure 5d. The main change is that, as with the delay case, the variance has increased. The MOOC model has a higher average cost per user than poker (almost double) reflecting an increase in users situated near the high cost servers.

The main two lessons from these results are: first that in these scenarios the QoS benefits from dynamically choosing cloud hosts is small (a slight decrease in time spent on unmanaged network when the number of servers selected is between three and five in the poker scenario); second that for the Amazon EC2 pricing model the differing location charges can mean that dynamically shifting servers can be a more expensive policy.

A number of variations on these base scenarios were tried. The repeatability of the results was tested extensively and in all cases repeated runs produced statistically indistinguishable results. The static model was varied to see if any difference was made by having different servers chosen for different times of day. For the scenarios tried, this did not make much difference. The distribution of the delays per user pair were investigated. The cumulative distribution function here is hard to interpret as it is a function not only of the scenario and servers chosen but also of the distribution of users across the planet. For this reason, results presented here are mainly in terms of mean delay and standard deviation, since interpreting results by comparing CDFs is problematic.

B. Scalability with respect to users

Next we explore the scalability with respect to users. For the Poker scenario, with three servers being used (from the ten Amazon EC2 data centres), the number of users is varied. For the Poker scenario, we try 2,500, 5,000, 7,500 and 10,000 users; for the MOOC scenario, we try 250, 500, 750 and 1,000 users. The MOOC scenarios are smaller to avoid unrealistically large sessions as all MOOC users are in the same session whereas the poker users are split into "tables". Again the dynamic, static and random strategies are used.

For poker, the scalability with respect to users is excellent—there is no statistically significant effect on delay or mean cost per user hour of increasing numbers of users in the system (see figures 6a and 6b).

In the MOOC scenario experiments, the cost per user increases linearly as the number of users increases (figure 6d). There are three effects at play: first, as the number of users goes up, and data must be transferred between each user pair, the number of streams goes up as the square of the number of users; however, there is a second effect: only one user transmits full bandwidth at any given time so as the number of users in a session goes up then the average stream bandwidth falls off with the number of users. The main effect here is that the cost per user increases (as every user must send some data to every other user in a session). This arises because a session with n users comprises $n(n-1)$ streams (from each user to each user).

The major lesson for system designers here is that the greatest scalability problem is an increase in cost per user if the size of sessions increases.

C. Sensitivity to session length

The session lengths for all simulations are drawn from a lognormal distribution with given mean and standard deviation. We now consider the effect of varying that mean. Results for the Poker and MOOC scenarios are tried with the mean session length varied in the set 1,000, 2,000, 3,000, 4,000 and 5,000 seconds. Each run uses three servers and the usual default parameters for those scenarios. These runs are to check the sensitivity to the assumptions about session length. Obviously by increasing the session length, the cost per user will be increased and, if no other effects come into play, the cost should increase linearly with the session length: A user online for n times as long uses n times as much bandwidth.

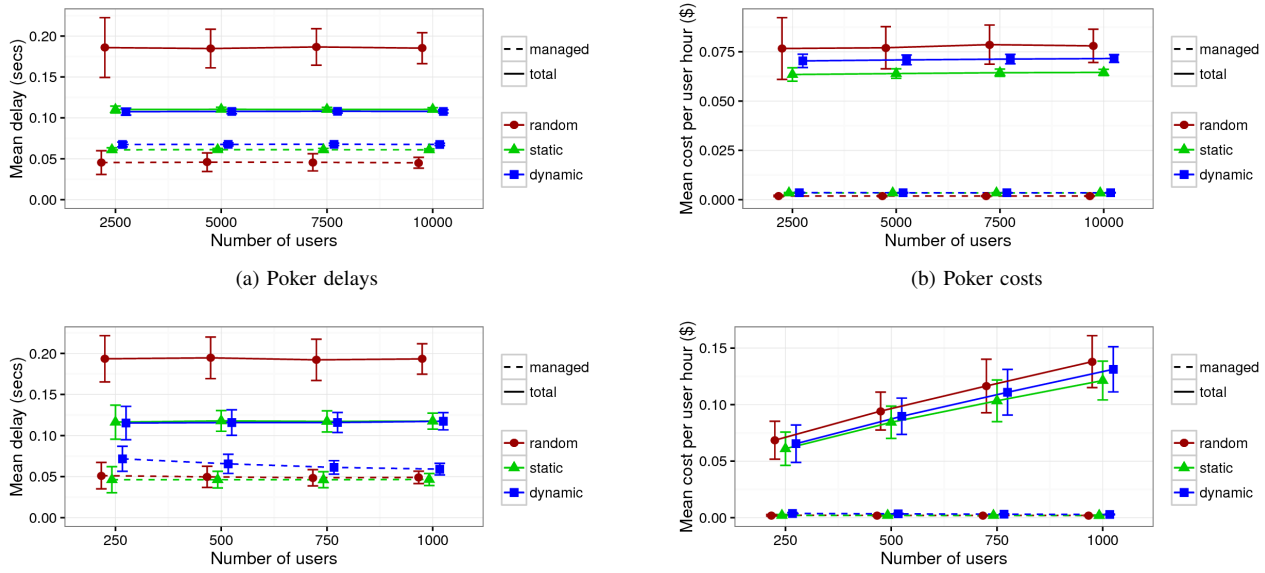
The figures for this experiment are not shown for space reasons but they reflect essentially the same story as in the previous section. The QoE results show no relationship between session length and delay for either scenario and are omitted here. The Poker model shows the expected linear relationship between session length and cost per user, that is the cost per user hour does not change—the model has a capped maximum number of users in a session, and therefore increasing the average number of users in the system does not greatly change the number of users in the average session. For the MOOC model, the results are different. As the session lengths grow, each user spends more time online but also the average size of the session grows and, hence, each user spends that time sending and receiving traffic from more other users. The graph is consistent with a linear increase in cost per user hour as would be expected in this case. This is mitigated by the fact that extremely large session sizes are simply not realistic for the MOOC scenario (since it posits that all users are in the same chat room and bases costs on all users having small scale video displayed).

For system designers the main lesson here is that increasing users' time spent online can affect scalability if this in turn increases the size of sessions.

D. Results with more data centres

This section describes results with a larger number of available data centres. As described in §IV-C, a set of 2,507 servers from around the world have been used to investigate what would happen if a finer-grained choice of server was available.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCC.2017.2680440, IEEE Transactions on Cloud Computing



(a) Poker delays

(b) Poker costs

(c) MOOC delays

(d) MOOC costs

Fig. 6. Results for the two scenarios varying the number of users per day and keeping three servers.

The Poker and MOOC scenario experiments from §VI-A are repeated using the larger number of servers. The price structure used for these is derived from the EC2 prices by geographical weighting as described in §IV-C. Figure 7a shows the poker scenario with the choice between the larger number of servers. As can be seen, here the dynamic server selection strategy has a great advantage. Because of the wider choice of servers selected, the dynamic strategy picks "good" servers more often than any other. By the time ten servers (the maximum session size in the poker scenario) are chosen, the time spent on unmanaged network is nearly zero with the dynamic selection strategy.

Figure 7b shows the costs for this scenario. The dynamic method imposes higher costs. This is because it more commonly chooses the higher priced servers when that is useful to reduce delay. As the policy was chosen to reduce delay without regard for cost this is to be expected.

Figure 7c and 7d show the same large data centre set of results for the MOOC scenario. The results are broadly similar to those for the poker scenario. In some cases (due to the larger error bars) the results from dynamic and static are not statistically distinguishable but the overall pattern is as for the poker scenario.

We also considered the model where the ten "best" data centres for the Poker scenario were used by choosing those ten that were identified by the static routing model. Rerunning the scenario with only those ten sites showed that the overall delay and pattern of behaviour for the server selection strategies was very similar to the base Amazon EC2 scenario. However, the proportion of time spent on managed network increased (from 63% to 70% for the ten server data point) indicating that those servers produced reduced time on unmanaged network for the Poker scenario but, perhaps surprisingly, not very much reduced.

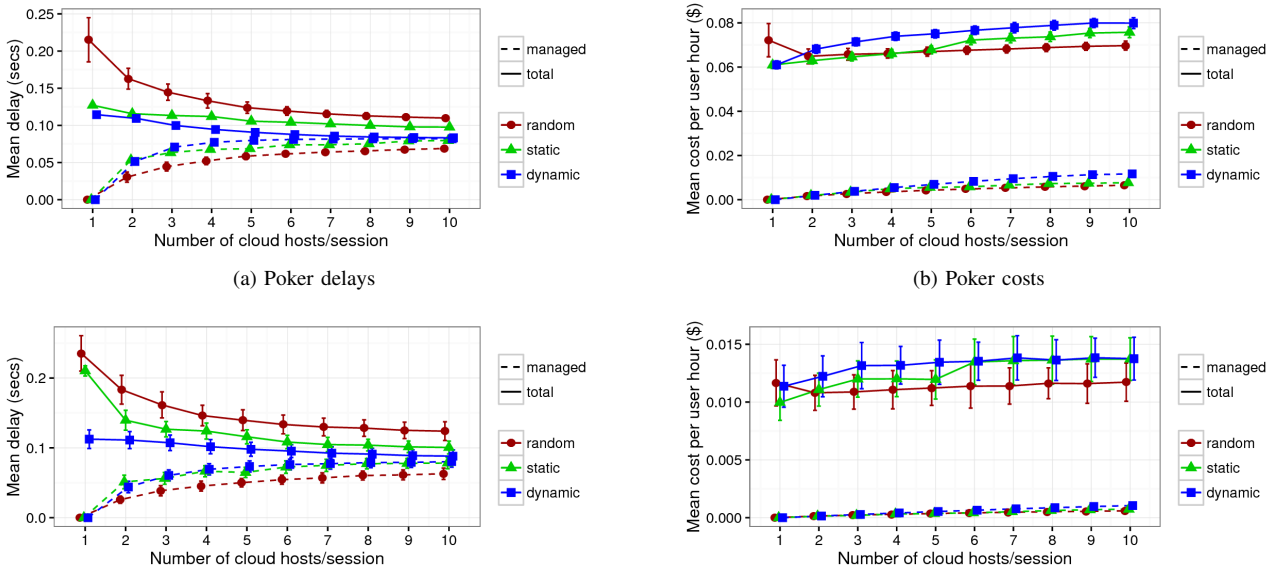
In terms of system design, the implication here is that more potential cloud location sites mean that the dynamic choice of sites becomes more effective at reducing delay and time spent on unmanaged network. However, it should also be noticed that the cost per user for internal network traffic increases linearly with the number of cloud locations used. Our experiments show that five server locations is a good trade off point as it gains almost all of the QoE advantage but almost halves the cost of internal traffic per user.

E. Different demand models

To investigate how much the results of the poker scenario were specific to the exact demand profile we created two further models representing different shifts of usage. The first represents an artificial scenario where every country has a completely equal desire to play online poker and the usage of online poker in a country is simply proportional to the internet using population of that country (not the total population). The second represents a 'half way' scenario where the proportion of demand at each site is an half that of the original poker scenario and half the 'equal usage' scenario.

These two scenarios should be compared with figures 5a for QoE and 5b for cost. The large shift in demand has produced some changes that are not at first apparent. Most importantly the overall delay has increased considerably. In the original poker model (Figure 5a) the total delay with all ten cloud hosts/session was just over 0.1 seconds. This has increased to nearly 0.2 seconds in the equal usage scenario (Figure 8a). It is somewhere between the two for the halfway scenario as might be expected (Figure 8c). This is a natural consequence of shifting the demand from being largely concentrated in western Europe and the United States to being more globally distributed. The graph shape, however, remains consistent and the policies, random, dynamic and static perform similarly well. Note also the larger amount of time spent on unmanaged network in the new scenarios. This is what might be expected when the users are further from their "nearest" cloud point. The western countries are relatively better provisioned with amazon EC2 sites (five of ten sites are in the United States or Europe).

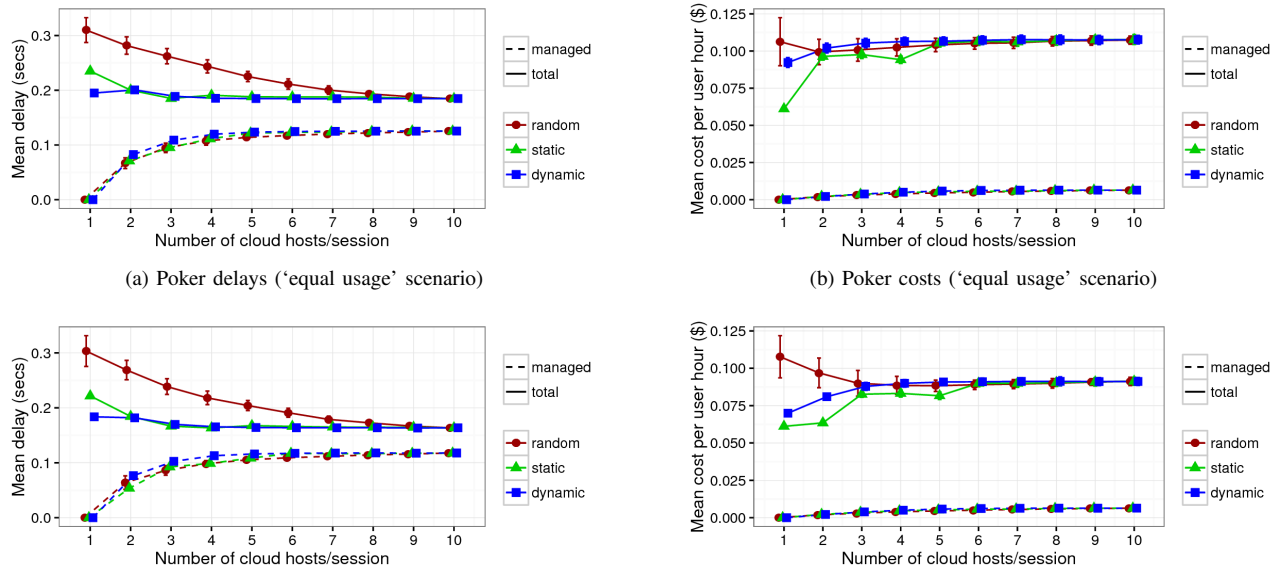
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCC.2017.2680440, IEEE Transactions on Cloud Computing



(a) Poker delays (b) Poker costs

(c) MOOC delays (d) MOOC costs

Fig. 7. Delay and cost versus number of cloud sites used per session for the many data centre large server scenario



(a) Poker delays (‘equal usage’ scenario) (b) Poker costs (‘equal usage’ scenario)

(c) Poker delays (‘halfway’ scenario) (d) Poker costs (‘halfway’ scenario)

Fig. 8. Delay and cost versus number of cloud sites used per session for the two different distribution scenarios

When considering costs the shift of demand away from the cheaper servers in Europe and the United States naturally brings about an increase in the average cost per user. The average cost with all ten servers has risen from $0.75/user hour in the original scenario (Figure 5b) to $1.1/user hour in the equal scenario (Figure 8b) and (again) part way between the two for the half way scenario (Figure 8d).

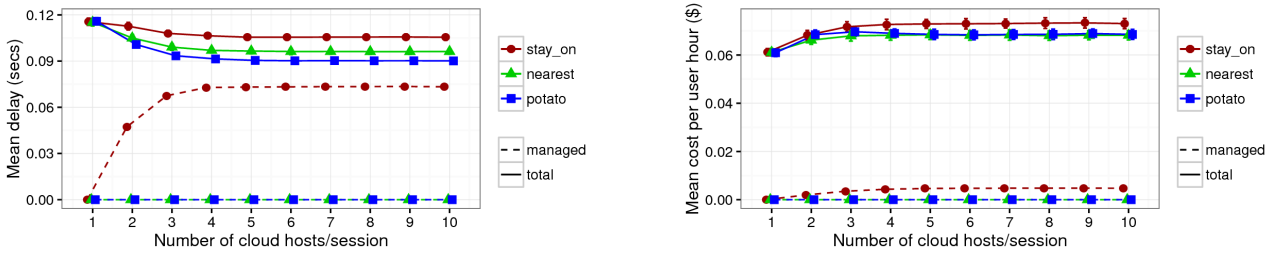
In terms of system design the lesson here is an obvious one. The more globally distributed the demand on the system the harder it will be to achieve good QoE for the users. It is also important to notice that the geographical location of demand can have a big effect on the cost of a running system due to the different prices in different locations.

F. Different routing models

In §IV-E, three routing models were described each with different aims: the *HotPotato* routing model minimises delay by routing by one server and one server only: the server (from the set of available servers) which minimises delay between the pair;the *NearestOn* model, by contrast, sends traffic from A to B by going to the server nearest to A then immediately to B—note that this model is asymmetric, i.e. a stream from A to B need not take the same path as a stream from B to A; finally the *StayOnRoute* model routes traffic from A to the server nearest to A, on to the server nearest to B and then to B. This may only be two hops if both A and B have the same nearest servers.

Figure 9a shows the delay for the three routing models with the base case scenario and three servers. The delays are in the order expected that is the HotPotato model has

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TCC.2017.2680440, IEEE Transactions on Cloud Computing



(a) Mean delay per user pair

(b) Mean cost per user hour

Fig. 9. Delay and cost for different routing models in the poker scenario

the lowest delay although this is only fractionally lower than the NearestOn model. The StayOnRoute model has a higher delay but this is more than offset in all cases by the shift onto managed network. In other words, the amount of time spent on unmanaged (poorer quality) network decreases in the StayOnRoute model. Whether this is sufficient to compensate for the slight (but statistically significant) increase in delay depends on user preferences and the available throughput on the managed and unmanaged network segments.

Figure 9b shows the costs for the modelling using the dynamic server selection policy Poker base case for the three routing policies. HotPotato and NearestOn use no managed network and the costs are purely on unmanaged. The two appear to have the same costs within the range of the error bars. The cost grows as more servers are added because this makes it more likely that high price servers will be added. Remembering that it is normal for demand to be centred in North America and Europe, once the session size is moderately large, one of these servers is by far most likely to be selected as a cluster centre. However, when two or three servers are to be selected, it is possible that one or two users in a distant location will have a server allocated near them. The shape of the cost function then is a combination of these distortions and the additional factor of the StayOnRoute model adding cost for the managed network used but this is of little significance by comparison.

The lesson for system designer here is that the two policies that do not use managed network can slightly reduce delay and cost. However, if the managed network is significantly more reliable, even three cloud host locations can ensure that a large proportion of the traffic's path is spent on managed network.

VII. Conclusions and Further Work

This article considers simulations of a multi-user, cloud-assisted global video conferencing system. We assume a video router (e.g. as developed by the Vconect project) that can migrate between the cloud hosted locations in a user-transparent manner. Using realistic demand from real-world data, we investigated two different demand scenarios (one based on gaming the other based on education). We compared scenarios where video router locations were selected statically and dynamically. Our experiments were run to determine the delay experienced by users as a result of the choices of cloud host location and routes for their video session.

In summary, our conclusions from this study are:

- When there is not a large number of host locations to choose from, dynamic migration does not offer much (if any) advantage over static choice of servers.
- Increasing the number of host locations to choose from increases the complexity of the problem. It helps delay and in particular reduces the amount of time users spend on unmanaged network.
- Scaling problems occur when the number of users per chat session increases (as cost from traffic leaving the network increases as the square of the number of users) and when the number of server locations in use by the session increases (as the amount of traffic between servers increases as the square of the number of locations).
- Varying the scenario considered greatly changes the QoE and cost per user. In particular, the cost is sensitive to the average number of users communicating in a single chat and (when charges vary by location) where those users are typically based.
- If the network between cloud hosts is significantly more reliable than the general internet, moving traffic onto that network as soon as possible has only minor impacts for delay and cost.

We conclude that, in current cloud architectures, there are benefits to dynamic migration of video server only in the case where cloud providers diversify their number of global locations.

The authors are exploring opportunities to exploit the results of this work, anticipating how globally-distributed ad hoc video conferences could create challenges for telecommunications providers. In particular, there is scope to extend the existing capabilities of cloud-hosted IP Multimedia Subsystem (IMS) components—and to investigate the potential advantages of dynamic video routing within and between IMS clouds connected using IP Exchange (IPX) services. Another aspect of future work is to use our model to evaluate suitable locations for new data centres to host video routers to improve QoE beyond that achievable with current deployments. To aid this further investigation, the code and data to replicate our results are publicly available.

VIII. Acknowledgements

The research leading to these results has received funding from the European Commission's Seventh Framework Programme (FP7/2007-2013) under grant agreements ICT-2011-287760 and ICT-2011-318205.

REFERENCES

[1] Y. Chu, S. Rao, S. Seshan, and H. Zhang, "Enabling conferencing applications on the Internet using an overlay multicast architecture," in *Proc. ACM/SIGCOMM*, 2001, pp. 55–67.

[2] N. Egi, N. Blundell, and L. Mathy, "Efficient overlay audio conferencing," in *Proc. 5th Int. IFIP-TC6 Networking Conference*, 2006, pp. 666–677.

[3] Y. Xu, C. Yu, J. Li, and Y. Liu, "Video telephony for end-consumers: Measurement study of google+, ichat, and skype," in *Proc. of ACM Internet Measurement Conference*, 2012.

[4] R. Landa, J. Araujo, R. Clegg, E. Mykoniati, D. Griffin, and M. Rio, "The large-scale geography of internet round trip times," in *IFIP Networking Conference, 2013*, 2013, pp. 1–9.

[5] Wainhouse Research, "A ready market: Introducing H.264-SVC," Sponsored white paper http://bicky.com.np/nac/wp-content/uploads/2012/04/wainhouse.pdf, 2006.

[6] M. Chen, G. Su, and M. Wu, "Dynamic resource allocation for robust distributed multi-point video conferencing," *IEEE Trans. on Multimedia*, vol. 10, no. 5, August 2008.

[7] ETSI, "Network functions virtualisation – introductory white paper," https://portal.etsi.org/nfv/nfv_white_paper.pdf, 2012.

[8] ETSI GS NFV, "Network function virtualisation (NFV): Use cases," http://www.etsi.org/deliver/etsi_gs/NFV/001_099/001/01.01.01_60/gs_NFV001v010101p.pdf, 2013.

[9] D. Geerts, K. D. Moor, I. Ketyko, A. Jacobs, J. V. den Bergh, W. Joseph, L. Martens, and L. D. Marez, "Linking an integrated framework with appropriate methods for measuring QoE," in *Proc. of QoMEX*, 2010, pp. 158–163.

[10] W. Wu, A. Arefin, R. Rivas, K. Nahrstedt, R. Sheppard, and Z. Yang, "Quality of experience in distributed interactive multimedia environments: toward a theoretical framework," in *Proc. of ACM Int. Conf. on Multimedia*, 2009, pp. 481–490.

[11] E. Geelhoed, A. Parker, D. J. Williams, and M. Groen, "Effects of latency on telepresence, HP labs technical report HPL-2009-120," http://www.hpl.hp.com/techreports/2009/HPL-2009-120, 2009.

[12] J. Tam, E. Carter, S. Kiesler, and J. Hodgins, "Video increases the perception of naturalness during remote interactions with latency," in *Proc. of CHI*, 2012, pp. 2045–2050.

[13] S. Agarwal and J. R. Lorch, "Matchmaking for online games and other latency-sensitive p2p systems," in *Proc. of ACM SIGCOMM*, 2009, pp. 315–326.

[14] ITU-T, "Recommendation p.1301, subjective quality evaluation of audio and audiovisual multiparty telemeetings," http://www.itu.int/ITU-T/recommendations/rec.aspx?rec=11687, 2013.

[15] Y. Chu, S. Rao, S. Seshan, and H. Zhang, "End-to-end delay analysis of videoconferencing over packet-switched networks," *IEEE/ACM Transactions on Networking*, vol. 8, no. 4, pp. 479–492, 2000.

[16] D. Roberts, T. Duckworth, C. Moore, R. Wolff, , and J. O'Hare, "Comparing the end to end latency of an immersive collaborative environment and a video conference," in *Proceedings of the IEEE/ACM international Symposium on Distributed Simulation and Real Time Applications*, 2009, pp. 89–94.

[17] I. Kegel, P. Cesar, J. Jansen, D. Bulterman, T. Stevens, J. Kort, and N. Frber, "Enabling togetherness in high-quality domestic video conferencing," in *Proc. 20th ACM int. conf. on Multimedia*, 2012, pp. 159–168.

[18] S. Choy, B. Wong, G. Simon, and C. Rosenberg, "The brewing storm in cloud gaming: A measurement study on cloud to end-user latency," in *Proc. Workshop on Network and Systems Support for Games*, 2012, pp. 1–6.

[19] R. Landa, R. Clegg, J. Araujo, E. Mykoniati, D. Griffin, and M. Rio, "Measuring the relationships between internet geography and rtt," in *International Conference on Computer Communications and Networks (ICCCN)*, 2013, pp. 1–7.

[20] S. Y. Shah, B. Szymanski, P. Zerfos, and C. Gibson, "Towards relevancy aware service oriented systems in wsns," *IEEE Transactions on Service Computing*, vol. 8, 2015.

[21] C. Bisdikian, L. M. Kaplan, and M. B. Srivastava, "On the quality and value of information in sensor networks," *ACM Trans. Sen. Netw.*, vol. 9, no. 4, pp. 48:1–48:26, 2013.

[22] I. Papapanagiotou, M. Falkner, and M. Devetsikiotis, "Optimal functionality placement for multiplay service provider architectures," *IEEE Trans. Network and Service Management,*, vol. 9, no. 3, pp. 359–372, 2012.

[23] Y. Feng, B. Li, and B. Li, "Airlift: Video conferencing as a cloud service using inter-datacenter networks." in *Proceedings of ICNP conference*, 2012, pp. 1–11.

[24] Q. Zhang, Q. Zhu, M. Zhani, and R. Boutaba, "Dynamic service placement in geographically distributed clouds," in *Proc. Dist. Computing Systems (ICDCS)*, 2012, pp. 526–535.

[25] F. Chen, C. Zhang, F. Wang, and J. Liu, "Crowdsourced live streaming over cloud," in *Proc. of INFCOMM*, 2015.

[26] N. Laoutaris, G. Smaragdakis, K. Oikonomou, I. Stavrakakis, and A. Bestavros, "Distributed placement of service facilities in large-scale networks," in *IEEE INFOCOM*, 2007, pp. 2144–2152.

[27] J. Li, M. Woodside, J. Chinneck, and M. Litoiu, "CloudOpt: Multi-goal optimization of application deployments across a cloud," in *Network and Service Management (CNSM)*, 2011, pp. 1–9.

[28] T. Stevens, I. Kegel, D. Williams, P. Cesar, R. Kaiser, N. Färber, P. Torres, P. Stenton, M. Ursu, and M. Falelakis, "Video communication for networked communities: Challenges and opportunities," in *Int. Conf. on Intelligence in Next Generation Networks*, 2012.

[29] M. Ursu, P. Torres, M. Frantzis, V. Zsombori, and R. Kaiser, "Socialising through orchestrated video communication," in *Proc. of the Int. Conf. on Multimedia*, 2011, pp. 1526–1530.

[30] T. Stevens, I. Kegel, and A. Wills, "Synchronisation of multi-party switched video streams," Patent (Pending) UK 14 250 109.7, 09 30, 2014.

[31] I. Kegel, D. Williams, T. Stevens, P. Hughes, R. Clegg, R. Landa, D. Griffin, and M. Rio, "Exploring the benefits of NFV and SDN for smart video conferencing services," in *NFV and SDN Summit*, 2015.

[32] R. G. Clegg, C. D. Cairano-Gilfedder, and S. Zhou, "A critical look at power law modelling of the internet," *Computer Communications*, vol. 33, no. 3, pp. 259 – 268, 2010.

[33] A. Li, X. Yang, S. Kandula, and M. Zhang, "Cloudcmp: Comparing public cloud providers," in *Proc. Internet Measurement Conference*, 2010, pp. 1–14.

[34] I. Fiedler and A.-C. Wilcke, "The market for online poker," *UNLV Gaming Research and Review Journal*, pp. 7–9, 2012.

[35] I. Fiedler, "The gambling habits of online poker players," *The Journal of Gambling Business and Economics*, vol. 6, pp. 1–23, 2012.

[36] Coursera, "Functional programming principles in Scala," http://docs.scala-lang.org/news/functional-programming-principles-in-scala-impressions-and-statistics.html, 2012.

[37] University of Edinburgh, "Report on MOOC usage," https://www.era.lib.ed.ac.uk/bitstream/1842/6683/1/Edinburgh%20MOOCs%20Report%202013%20%231.pdf, 2013.

[38] Coursera, "Computational investing, part I," http://augmentedtrader.wordpress.com/2013/05/25/mooc-student-demographics-fall-2012/, 2012.

[39] ——, "Computational investing, part I," http://augmentedtrader.wordpress.com/2013/01/27/mooc-student-demographics/, 2012.

[40] The Knight Center for Journalism in the Americas, "Introduction to data journalism," https://knightcenter.utexas.edu/00-14152-nearly-4000-students-62-countries-participate-knight-center%E2%80%99s-mooc-introduction-data-journa, 2013.

[41] Harvard Uinversity, edX, "6.002x: Circuits and electronics," http://www.rpajournal.com/dev/wp-content/uploads/2013/05/SF2.pdf, 2012.

[42] Coursera, "Internet, history, technology and security," 2012.

[43] Harvard Uinversity, edX, "CS50x: Introduction to computer science," https://blog.cs50.net/2013/05/01/0/, 2012.

[44] Statistics Brain, "Skype statistics," http://www.statisticbrain.com/skype-statistics/, 2012.

[45] CIA, "CIA world factbook internet users," https://www.cia.gov/library/publications/the-world-factbook/rankorder/2153rank.html, 2013.

[46] Geonames, "GeoNames database," http://www.geonames.org, 2013.

APPENDIX A
DETAILS OF DEMAND MODELLING

The scenarios consist of a demand model and a session model. The demand model has an arrival component. The location of demand is expressed as a set of triples (x_i, y_i, l_i), latitude, longitude and rate. First the problem was broken down into two sub-problems:

- Estimate the *relative* Internet-using population L_i at (x_i, y_i)—the figure is proportional to the number of internet users at that location.
- Estimate the *relative* proportion $P_{s,i}$ of the Internet using population at location i using scenario s.

It is then the case that $l_i = P_{s,i}L_i$. Note that because everything is going to be multiplied by a constant λ, only relative numbers are needed, and it is not necessary to know the exact population at a site, only its size in relation to other sites.

The relative Internet-using population is calculated from three data sets:

1) the number of Internet users in each country;
2) a list of all population centres with more than 1,000 inhabitants; and
3) a list of the locations of DNS servers in the world.

First, [45] gives the Internet-using population in each country. Second, from [46], a list of all centres of population with more than 1,000 inhabitants was obtained and these were used as the latitude and longitude (x_i, y_i). The next step was to split the Internet-using population of a country between the centres of population. Instead of using a population-related split, the number of DNS servers in the location was used as a proxy for Internet penetration in that area. A dataset of DNS servers was collected by the authors in [19] and this is used for the normalisation. Hence, demand is given in triples (x_i, y_i, l_i) with the following properties:

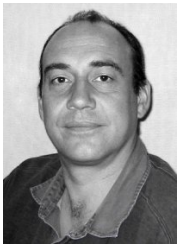
- Each location corresponds to a population centre with more than 1,000 inhabitants.
- The total demand over all locations in a country sums to the correct proportion of demand within that country.
- The demand in a location is proportional to the proportion of DNS servers within that country at that location.

This method gives a reasonable assignment of Internet demand to location from the available data sources.

For neither scenario could we obtain data about exact locations of users and hence proportions split down to a country level were used in the calculation of $P_{s,i}$. In the absence of better data, we therefore assumed that users in a country were split across the locations in that country according to the proportion of the demand in that location within the country.

Richard G. Clegg is a Research Fellow at the Department of Computing in Imperial College London. His PhD in mathematics and statistics from the University of York was gained in 2005. His research interests include investigations of the dynamic behaviour of networks and measurement of network traffic statistics.

Raul Landa is currently a Data Scientist in the Network Planning and Operations group within Sky UK Network Services. His research interests involve the measurement-based modelling of Internet video quality of experience, the large-scale behaviour of content delivery networks and the strategic aspects in peer-to-peer overlays. He received a PhD in Electronic and Electrical Engineering from University College London, UK.



David Griffin is a Principal Research Associate at University College London. He has a PhD in Electronic and Electrical Engineering from UCL. His research interests are in planning, management and dynamic control for providing QoS in multi-service networks, p2p networking and novel routing paradigms in the Internet.

Miguel Rio is a Reader (Associate Professor) in Computer Networks at the Department of Electronic and Electrical Engineering at University College London. He has published extensively in top ranked Conferences and journals in the areas of Network measurement, congestion control, new network architectures and, more recently, in the interaction between cloud and network services. He holds a PhD from the University of Kent at Canterbury and MSc and MEng from the Department of Informatics, University of Minho, Portugal.

Peter Hughes is a Senior Researcher in BT Research & Innovation where he has worked for over 40 years. During this time he has worked on a wide variety of projects including speech band data transmission, voice and audio technologies, accessible technologies and voice quality. His current research interests include voice quality assessment in multi-party communications and real-world performance of voice technologies.

Ian Kegel leads the Future Content Group within BT Research & Innovation, where his research includes understanding, measuring and improving the quality of audio and video communication experiences. Ian has worked in both the defence and telecommunications industries on projects ranging from radar signal processing to multimedia delivery, and has spent more than 12 years leading content-related research within BT.

Tim Stevens is with BT Research & Innovation near Ipswich, U.K. and has written all kinds of software from embedded systems to real-time video tools and gained several patents. He is currently contributing to Standards and developing techniques for distributing broadcast video over IP. Tim is a Chartered Engineer and member of the U.K.'s Institution of Engineering and Technology.

Peter Pietzuch is a Reader (Associate Professor) in the Department of Computing at Imperial College London where he leads the Large-scale Distributed Systems (LSDS) group that does research on scalable distributed software systems of any kind. He received his PhD degree in computer science from the University of Cambridge.

Doug Williams is a principal researcher at BT Research & Innovation. He has a PhD in the design and fabrication of special optical fibres. His current and recent interests include the likely viability of new services and their impact on overall demand for bandwidth.