Dynamic Cloud Service Placement for Live Video Streaming with a Remote-Controlled Drone

Florian Wamser, Frank Loh, Michael Seufert, Phuoc Tran-Gia
University of Würzburg, Institute of Computer Science, Germany
{wamser,frank.loh,seufert,trangia}@informatik.uni-wuerzburg.de
Roberto Bruschi, Paolo Lago
CNIT - Genoa Research Unit, DITEN University of Genoa, Genoa, Italy
roberto.bruschi@cnit.it, paolo@tnt-lab.unige.it

***Abstract*—In this demonstration we will show the prospects of dynamic cloud service placement. A cloud service is implemented that manages real-time video streaming and real-time control commands for a remote controlled drone. The requirements for this cloud service are groundbreaking because the image transfer and the control commands should be transmitted in real time that the drone can be controlled smoothly by a user. The goal is to improve user QoE for streaming services and real-time control by cloud service migrations, respecting the concept of dynamic Edge Computing by means of moving computing and monitoring applications on demand close to the user and the end device.**

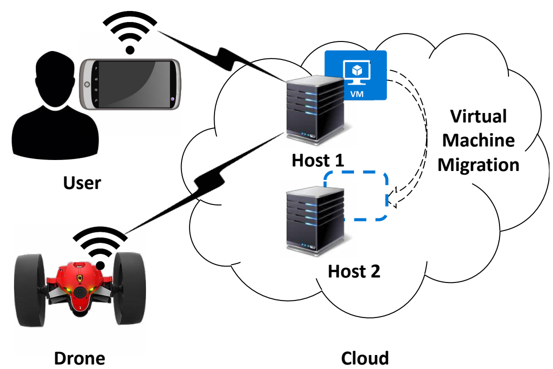

Fig. 1. Demonstration scenario and drone controlling setup

I. Introduction and Problem Description

The continuous growth of Internet content, applications and services has led to ever more demanding requirements in the networks. To meet this, there have been significant developments in the migration of applications and services towards the cloud [1], [2].

The cloud not only provides applications, but also offers dynamic adaptability, scalability, and optimization with respect to the location of the application by migrating storage and application containers. In particular, cloud services can typically be placed and orchestrated at a variety of locations in the cloud. A new trend is the implementation of edge server resources to place services near the user, extending the cloud to the edge of the access network [3].

This demonstration shows a dynamic placement of a cloud service with video streaming and real-time control commands for a remote-controlled drone. We control a drone with the help of a cloud service which, in addition to the control commands, also returns a camera image to the operator. The requirements of this cloud application are groundbreaking. The camera requires high data rates from the drone to the cloud and from the cloud to the operator, so that the image can be transmitted in acceptable quality. For the control of the drone, a short delay must also be given in the direction of the drone so that the control can be carried out in real time.

From a technical point of view, the cloud service is consistently placed and moved according to the requirements of the application. This allows the audience to recognize the necessity and usefulness of service migrations in the cloud. The scenario is additionally motivated by the trend that many services include and expect adaptivity because of their needs and high expectations in user experience - both on the application side, as well as in the network or in the cloud. We trigger a migration of the cloud service according to the user experience based on the control information of the drone and the playback quality of the camera image. This is achieved by the targeted monitoring of objective and measurable properties of the application in the network, which have a high correlation to the user experience (Quality of Experience, QoE) such as video resolution, framerate and latency of the control information [4].

The provision of high quality video content with little delay poses several challenges for the cloud. Next to the given scalability and simple data distribution, a key challenge is the service optimization within the cloud. Especially for live video streaming, easy and fast migration is essential to guarantee the satisfaction of the streaming user. With the increased mobility of each user together with the demand of high quality streaming on tablets and smartphones, the way to migrate is different for each service. For this reason, a cloud framework was developed in the course of the H2020 INPUT project [5], which implements the concept of QoE monitoring and service migration. Furthermore, the concept includes the consideration of edge resources, so that a user-oriented migration is possible.

The course of demonstration is as follows. The drone is steered by an application at the user's smartphone. The smartphone is connected to a virtual maschine in the cloud infrastructure that forwards steering information to the drone.

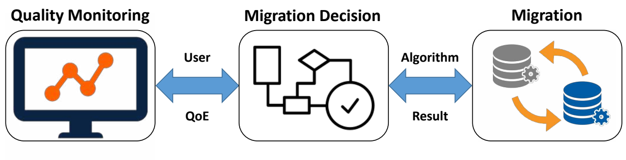

Fig. 2. Quality monitoring and service migration

By a cloud based monitoring of the streaming properties, the quality is observed. If a quality degradation based on the defined QoE parameters is detected, a migration is started to improve the streaming and controlling. In Figure 1 the scenario is presented.

II. Implementation and Required Components

This section highlights the implementation, required components and the software used in the demonstration. The monitoring setup and server migration is introduced.

For controlling the drone, an application based on the open-source SDK framework provided by Parrot is running on a VM exposing a web GUI to the end-user. With it, the user can drive, record videos and make pictures with the drone. Additionally, a route plan can be created. After starting this plan the drone is driving this route without any further interaction with the user required.

Each steering information from the smartphone app is forwarded to a host in the cloud within a virtual machine running the cloud service application as depicted in Figure 1. When the smartphone application is started the drone is starting a UDP video stream in the motion JPEG (MJPEG) format. The stream is sent to the host in the cloud infrastructure operating as a video proxy and a web server. The user's smartphone connects to the web server to receive the live stream. At the web server frames per second (FPS), video quality and dropped frames are monitored by Deep Packet Inspection (DPI) at the application layer, having a high correlation to the user's QoE. With the monitoring result, the QoE for the user is estimated. If any monitored parameter is falling below a predefined threshold for a given time period a migration algorithm is called. An overview of the quality monitoring and migration is presented in Figure 2.

By driving the drone away from the host or behind obstacles single frames of the stream are dropped. By a lower frame rate, the video quality is declining and drone controlling is delayed. This involves a degradation of user's QoE. If it drops below a given threshold, a migration algorithm starts a service migration inside the cloud. When migrating from one host to another, the goal is to minimize the number of hops to the client and the drone. This leads to an increased frame rate and a smooth control after migration. An exemplary service migration is presented in Figure 3. While having several dropped frames at the left side of the figure, after server migration at the right side of the figure all frames are received at the user's device.

The approach of monitoring and dynamically migrating in the edge cloud is developed in the INPUT project going

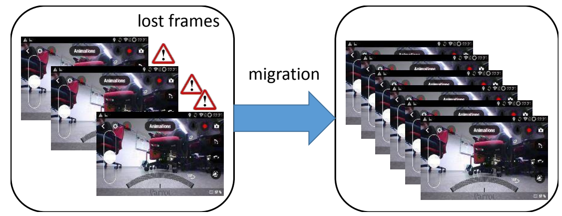

Fig. 3. Increased frame rate after migration

beyond typical IaaS-based service models. The goal to improve user QoE for streaming services by Software Defined Network (SDN) and Network Funcitons Virtualization (NFV) is compatible with the concept of Edge Computing by moving computing and monitoring applications from datacenters closer to the end user.

III. Demonstration Scenario and Presentation

For the demonstration scenario the setup presented in Figure 1 is created. The drone, a smartphone with the running controlling application and two servers with running virtual machines are installed. The first server has a poor connection to the drone causing high packet loss. The second server has a good connection without losing packets serving a high user's QoE.

At the demonstration users can steer the drone with our smartphone. By switching from the well connected server to the other, differences in streaming behavior can be detected. Especially the difficulty of controlling the drone with a bad server connection, high packet loss and bad video quality is presented.

The differences in quality are visualized at a separate monitor including FPS, packet loss and video quality. By changing hosts in the simulated cloud environment improvements and and degradation are displayed and quantified. A comparison between amount of packet loss and video quality degradation and difficulty in drone steering is presented. Additionally, the result of the delivered video stream after packet loss is displayed at the user's smartphone.

Acknowledgment

This work was partly funded in the framework of the EU ICT project INPUT (H2020-2014-ICT-644672) and the DFG grant QoE-DZ (TR257/41). The authors alone are responsible for the content.

References

[1] S. Kaisler and W. H. Money, "Service migration in a cloud architecture," in *System Sciences (HICSS), 2011 44th Hawaii International Conference on*. IEEE, 2011, pp. 1–10.
[2] U. Mandal, M. F. Habib, S. Zhang, B. Mukherjee, and M. Tornatore, "Greening the cloud using renewable-energy-aware service migration," *IEEE network*, vol. 27, no. 6, pp. 36–43, 2013.
[3] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*. ACM, 2012, pp. 13–16.
[4] M. Seufert, S. Egger, M. Slanina, T. Zinner, T. Hoßfeld, and P. Tran-Gia, "A survey on quality of experience of http adaptive streaming," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 1, pp. 469–492, 2015.
[5] "In-network programmability for next-generation personal cloud service support." [Online]. Available: http://www.input-project.eu/