

Research Article

Architecture and Protocol to Optimize Videoconference in Wireless Networks

Jose M. Jimenez, José Luis García-Navas, Jaime Lloret , and Oscar Romero

Instituto de Investigación para la Gestión Integrada de Zonas Costeras, Universitat Politècnica de València, Spain

Correspondence should be addressed to Jaime Lloret; jlloret@dcom.upv.es

Received 23 February 2020; Revised 27 August 2020; Accepted 13 September 2020; Published 6 October 2020

Academic Editor: Tuan M. Nguyen

Copyright © 2020 Jose M. Jimenez et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In the past years, videoconferencing (VC) has become an essential means of communications. VC allows people to communicate face to face regardless of their location, and it can be used for different purposes such as business meetings, medical assistance, commercial meetings, and military operations. There are a lot of factors in real-time video transmission that can affect to the quality of service (QoS) and the quality of experience (QoE). The application that is used (Adobe Connect, Cisco Webex, and Skype), the internet connection, or the network used for the communication can affect to the QoE. Users want communication to be as good as possible in terms of QoE. In this paper, we propose an architecture for videoconferencing that provides better quality of experience than other existing applications such as Adobe Connect, Cisco Webex, and Skype. We will test how these three applications work in terms of bandwidth, packets per second, and delay using WiFi and 3G/4G connections. Finally, these applications are compared to our prototype in the same scenarios as they were tested, and also in an SDN, in order to improve the advantages of the prototype.

1. Introduction

Video conferencing is a widespread means of communication in an era where technologies are constantly evolving. It allows people to communicate all over the world using only an electronic device connected to the Internet. Video conferencing allows not only video and voice transmission but also data transmission, allowing collaborative working. Video conferencing has always been characterized by the necessity of synchronization, low delay, low jitter, low loss ratio of packets, etc., and it has to confront the user's requirements, according to the quality of the communication, that are constantly increasing.

The audio and video contents that are transmitted over the Internet are constantly growing. One of these contents that is transmitted over the Internet is television. It is known as IPTV (Internet Protocol Television), and it consists of the distribution of high-quality television content [1]. It can be real-time video or Video on Demand (VoD). IPTV provides traditional TV services to the home through the Internet Service Providers (ISP). IPTV has become a key product for

Internet Service Providers (ISP). IPTV offers benefits to both ISP and end users [2]. The Internet was not designed to transmit real-time video/audio information, so Quality of Service (QoS) is one of the most important tasks to deal with when generating IPTV services. QoS directly affects the Quality of Experience (QoE). Users demand the best QoE as possible. In order to evaluate QoE, there are both objective and subjective approaches, such as objective metrics and subjective users' evaluation [2]. Huang et al. [3] propose data-driven QoE prediction for IPTV services. They firstly evaluate user experience of IPTV in a data-driven approach, and then they analyze the user's interest and experience.

Real-time video services, such as IPTV, and especially videoconferencing, require rigorous QoS requirements in terms of bandwidth, delay, and jitter. Video transcoding is a challenging task, and meeting QoS requirements could be critical to transmit information in a reliable and secure manner [4]. QoS needs to be analyzed not only from a metric perspective but also from a customer satisfaction perspective. Quantitative metrics must be correlated with qualitative metrics from the customers [5]. QoE in video

streaming is a task that can be improved in both wired and wireless networks.

QoE prediction consists of studying the relationship between the user experience and features from the video. Some authors have investigated ways of improving QoE over wired networks by different ways. Mao et al. [6] propose an IPTV user QoE prediction algorithm based on the Long Short-Term Memory (LSTM) network. They select subjective and objective features and use the LSTM network to perform QoE modeling. Their proposal shows a higher performance in QoE prediction than other conventional neural networks. Other authors such as Jiménez et al. [7] analyze QoE from the point of view of the number of devices connected and the use of bandwidth. They study a basic topology on which a video is broadcasted and propose an algorithm to improve quality of experience when network parameters can vary.

Quality of experience can also be studied over wireless networks. Su et al. [8] conducted a survey on existing literatures about video streaming QoE. They started from the point of view of the resource allocation problem and managed to bring together separated QoE metrics into seven categories. This allowed them to analyze their importance and complexity in video source coding and wireless networks. All these measurements can be used to carry out some other investigations. QoE management systems can be developed to guarantee enough QoE in IPTV services [9]. Delay, jitter, bandwidth, and zapping time measurements are used to calculate QoE over wireless networks, using a formula.

Video streaming, including videoconferencing, consumes a substantial portion of network resources. Video transmission requires high-bandwidth and strong latency requirements. Software-Defined Networking (SDN) gives the possibility of changing the network dynamically. SDN, joined to other techniques oriented to improve video streaming, can optimize video transmission through flexible controls [10]. Jimenez et al. [11] carried out a performance comparison between Mininet and a real network when multimedia streams were being delivered. Bandwidth, delay, and jitter were studied.

Taking into account these issues, this paper proposes an architecture for videoconferencing to provide better quality of experience than other existing solutions. First of all, the system used in our prototype is defined. This system consists of an E2E QoE management scheme for real-time video communication structured in different layers. The system will have three basic processes, which correspond to the basic actions to establish a videoconference: register, connection, and transmission process. Later, a finite-state machine is proposed, and also the different states are presented and defined. In addition, three different existing VC applications (Adobe Connect, Cisco Webex, and Skype) are tested in terms of bandwidth, packets per second, and delay, using WiFi and 3G/4G connections. Finally, these applications are compared to our prototype in the same scenarios as they were tested and in another scenario where SDN is applied in order to improve the advantages of the prototype.

The remainder of this paper is organized as follows. Section 2 presents some related work. The architecture pro-

posal for videoconferencing is explained in Section 3. Section 4 describes the proposal of the protocol of communication. The performance test in videoconference applications is carried out in Section 5. Section 6 presents the performance test of the developed application. And finally, Section 7 draws the main conclusions and future works.

2. Related Works

This section presents some works where video streaming, in particular videoconferencing, is studied from different points of view.

Chakraborti et al. [12] propose an audio/videoconferencing architecture based on the Virtualized Service Edge Router (VSER) platform. Their solution uses the VSER platform for notifications and the ICN framework for data exchange. This design provides good scalability and reliability, and it also allows discovering of new participants, dynamic synchronization, and multicasting for data exchange.

In [13], Hajiesmaili et al. discuss about the multiparty cloud videoconferencing architecture. They study the advantages of using cloud resources to effectively improve videoconferencing performance. The proposed architecture consists of using multiple agents that perform transcoding tasks. Each user is assigned to the best agent in terms of bandwidth and processing availabilities for each one. Their solution decreases the operational cost and reduces conferencing delays.

There are many papers that present improvements and solutions for videoconferencing using WebRTC for different purposes. Jang-Jaccard et al. [14] propose a design and implementation of a practical videoconferencing system for telehealth using WebRTC technology. Their goal is to evaluate the possibility of improving healthcare outcomes by high-bandwidth-enabled telehealth services. Their solution seeks to be standard-based, interoperable, simple, and inexpensive. They show the limitations of using WebRTC, describe various insights into the prototype implementation, and provide code snippets.

Bestak and Hlavacek [15] discuss a videoconferencing platform based on WebRTC technology. They test the impact on the multiplexing server's CPU load and RAM requirements for different numbers of users, using different hardware and software configurations at end-point devices. The results show a strong relation between the video resolution and bit rate, and the importance of dimensioning the server according to the number of users.

Pasha et al. [16] show the shortcomings and challenges faced by videoconferencing through WebRTC and propose a Multipoint Control Unit (MCU) as a solution. They propose the best centralized architecture to support WebRTC by using MCU. Their aim is to expose how WebRTC works and how it can be improved.

Many other authors propose other different solutions for carrying out videoconferencing. Gusev and Burke [17] present a discussion about the design and implementation in C++ of Real-Time Videoconferencing over Named Data Networking (NDN-RTC) on an NDN testbed. They build the solution in C++ using the WebRTC library due to the

necessity of reasonable CPU and bandwidth efficiency. They generate a functional low latency streaming tool that can be used as a platform for studying design challenges in real-time media over NDN.

Sambath et al. [18] face the task of improving the QoS scheme in an IP multimedia subsystem (IMS) for videoconferencing. They implement IntServ and DiffServ with MPLS and study parameters such as end-to-end delay, packet loss, and jitter. Their investigation shows that proper adaptation of QoS and appropriate resource allocation provide qualitative transmission of videoconferencing in a wireline.

Hossain and Khan [19] investigate a novel Multipoint Videoconferencing (MVC) architecture potentially suitable for a Peer-to-Peer (P2P) platform, such as Gnutella. Their proposal is based on the idea that autonomous peer nodes can dynamically assume the role of the MCU. This idea improves the architecture by minimizing total traffic, individual node hotness, and video composition delay.

Video conferencing is widely used for medical purposes, and many papers are focused on how technologies can improve videoconferencing for health. Khalifeh et al. [20] describe an e-health videoconferencing platform to facilitate patients' follow-up and communication with their healthcare professionals from a distance and at low cost. This system is developed for its potential usage in the Jordanian healthcare system and, in particular, medical centers and hospitals located in the rural areas. The main challenge is its high cost, so the proposed platform seeks to provide similar service at a lower cost.

In [21], Taylor et al. study which technical factors influence the quality of videoconferencing in the home setting and evaluate the impact of these factors on the clinical perceptions and acceptance of videoconferencing for health care. They conclude that the quality of videoconferencing when using 3G instead of broadband fiber-based services was less due to failed calls, jitter, and video pixilation.

Mat Kiah et al. [22] propose a secure framework for health videoconferencing systems and a complete management solution for secure videoconferencing groups. They use Real-Time Transport Protocol over UDP to transmit information, and they also use RSA and AES algorithms to provide security services. Their study shows that an encryption algorithm insignificantly increases the videoconferencing computation time.

Furthermore, the correct operation of videoconferencing depends on secure and reliable communication such as good QoS and QoE. Many papers are focused on these aspects. Mishra et al. [23] study how cryptographic techniques are used to achieve security protection to videoconferencing. The authors propose a novel, computationally efficient and secure video encryption algorithm. Security and performance analysis are carried out over their algorithm and show that it is well secured, computation efficient, and applicable for real-life operations.

Pattaranantakul et al. [24] present an achievable secure videoconferencing system based on quantum key encryption. They propose a secure key management methodology to ensure a trusted quantum network and a secure videoconferencing system. Their proposal includes secure communica-

tion channels to exchange secret keys and management. The authors point out that encryption can produce some initial delay.

Zhao et al. [25] present an overview of selected issues about QoE and its application in video transmission. The authors study QoE modeling, assessment, and management of video transmission over different types of networks.

Gunkel et al. [26] study different video stream configurations and layouts for multiparty conferencing in respect to individual network limitations. This study explores the relationship between QoE and three different factors: layout, video quality (resolution), and network limitations (packet loss).

In [27], García et al. show the procedure to set up a server to support the MPEG DASH protocol in the Polimedia e-learning System. They use this server to present a subjective QoE study to evaluate the performance of MPEG DASH. The authors determine the aspects that are most annoying to users of Polimedia. They carry out this study in order to improve the QoE of the user. They conclude that an 8-second video is the most stable segment size for videos of Polimedia.

Finally, all the solutions, ideas, and improvements presented before can be improved by using SDN, due to the possibility of changing the network dynamically and adapting it to the necessities. Henni et al. [28] focus on an improvement of the traditional OpenFlow Controllers. They propose a dynamical QoS routing implemented by a new controller. This new way of routing supports video conferencing flow delivery over OpenFlow networks. Dynamical routing focuses on protecting such traffic over nonconstrained flows. Their proposal simulated under Mininet shows the effectiveness of the proposed approach.

Yang et al. [29] propose a videoconferencing architecture based on SDN-enabled Scalable Video Coding (SVC) multicasting. The architecture discards the traditional Internet Group Management Protocol (IGMP) and MCU to obtain a better performance. Their results show that their system can provide flexible and controllable video delivery, can reduce the network bandwidth usage, and can guarantee the quality of a videoconference.

Al Hasrouty et al. [30] investigate the impact of using SVC and SDN techniques on videoconferencing. Their aim is to reduce the bandwidth consumed by videoconferencing (using SDN) and take advantage of SVC by sacrificing video quality for usability purposes. Their algorithm defines where and how many video layers should be dropped in order to adapt the streams to the bandwidth capacities of the network.

Our proposal improves the methods previously described. We ensure better End-to-End (E2E) QoE in videoconferencing by using the Network-adaptive Control Protocol, adjusting the transmission to the optimal values based on the characteristics of the devices and network. In addition, our proposal includes the use of SDN to get an optimal network transmission.

3. Architecture Proposal for Video Conference

3.1. System Definition. We must define an E2E QoE management scheme for real-time video communication systems, including those operating in resource varying environments.

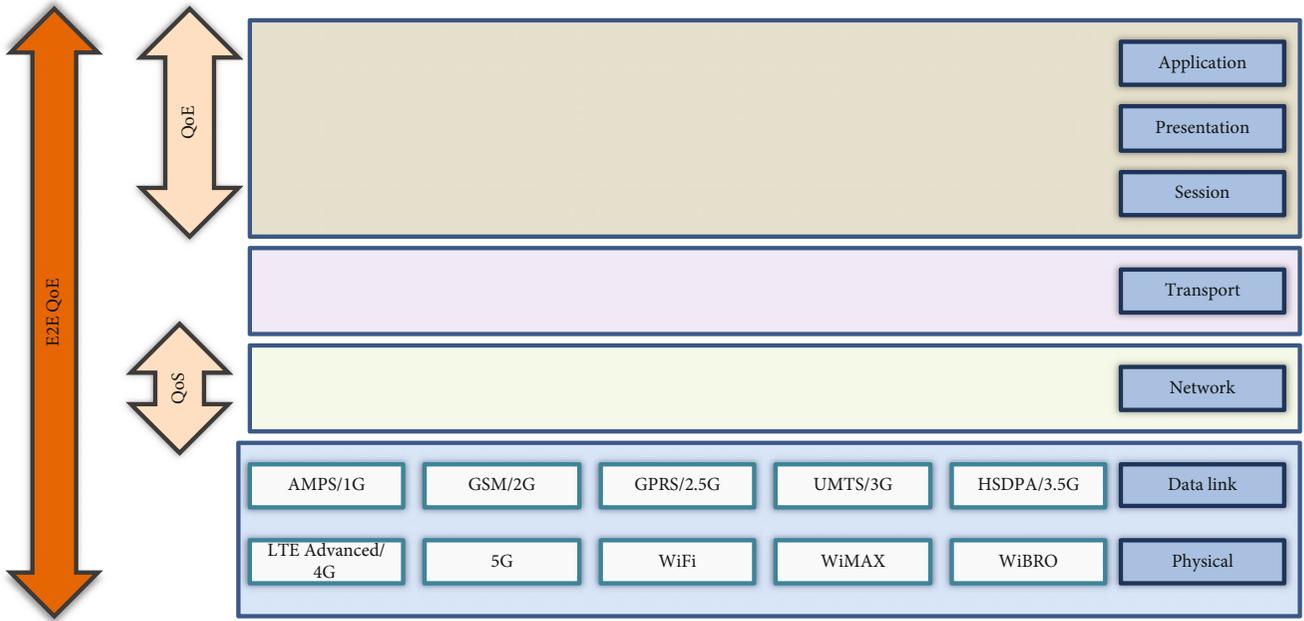


FIGURE 1: Proposed layer architecture, according to the type of QoS and QoE parameters.

In our proposal, we define a human visual perception-based E2E QoE metric and the methodology for correlating this metric to real-time video data, application/network-level QoS measurements, the capabilities of user devices, and subjective user factors.

Initially, to use it in our work, different groups of users observed the transmission of multiple videoconferences. The subjective quality of each videoconference was defined by the user's perception. This is measured in mean opinion score (MOS), from 1 to 5, where 1 is perceived as very bad quality, and 5 is considered very good quality. In this way, we obtained a subjective QoE classification to apply to the different transmissions. In addition, we vary the network parameters and the characteristics of the streams used in the communication equipment in each one of them, so that in the proposal of our system, what we do is adjust the maximum QoE selected by our users, based on to the parameters that are obtained from the computers that are communicating and the network.

We also define network-adaptive video-encoding and decoding algorithms utilizing device-based E2E QoE-driven feedback and, where available, network-based E2E QoE-driven feedback to achieve real-time adaptation according to the available device and/or network resources.

Besides, we define real-time device-based and network-based feedback control mechanisms that can be used to regulate E2E QoE by one or more of the following methods: application-level objective measurement and reporting of the actual received real-time video signal quality; network-level objective measurement and reporting of the in-transit real-time video signal quality; application-level measurement and reporting of device and/or network resources and QoS performance; and network-level measurement and reporting of device and/or network resources and QoS performance.

To carry out these objectives, we will consider which parameters affect the QoE, which algorithm is the most appropriate for the network, which algorithms are the most appropriate to provide the best QoE for end-user devices, how to make the network "adaptive" for this case, and what is the best system decision procedure to provide an "adaptive" network.

The proposed layer architecture, according to the type of QoS and QoE parameters that are considered, is shown in Figure 1.

An architecture that includes all the previously established objectives is shown in Figure 2.

As can be seen in Figure 2, our architecture is based on a Network-adaptive Control Protocol. Through this protocol, we manage to adapt the transmission between the end users to the maximum possible QoE. To achieve the goal, we must take into account how to handle a large amount of information, at least during the initial process of the connection.

Information can be classified depending on where the information is obtained from: obtained from the source devices, obtained from the network between end users, or obtained from the destination device. The information obtained from the source devices can be about available features of the devices (type of camera, CPU, RAM, and iOS), characteristics of device-based network analysis (bandwidth, delay, jitter, and packet loss), characteristics relative to video compression that can be achieved (codecs supported by software), and data calculated from monitoring video-encoded targets (bits/frame, frames/sec, and achievable QoE). Information that can be obtained from the network between end users includes available features of the devices in all the networks where the communication between end users pass through (bandwidth, delay, jitter, packet loss, and achievable QoE). Information that can be obtained from the destination device includes available features of the devices (type of

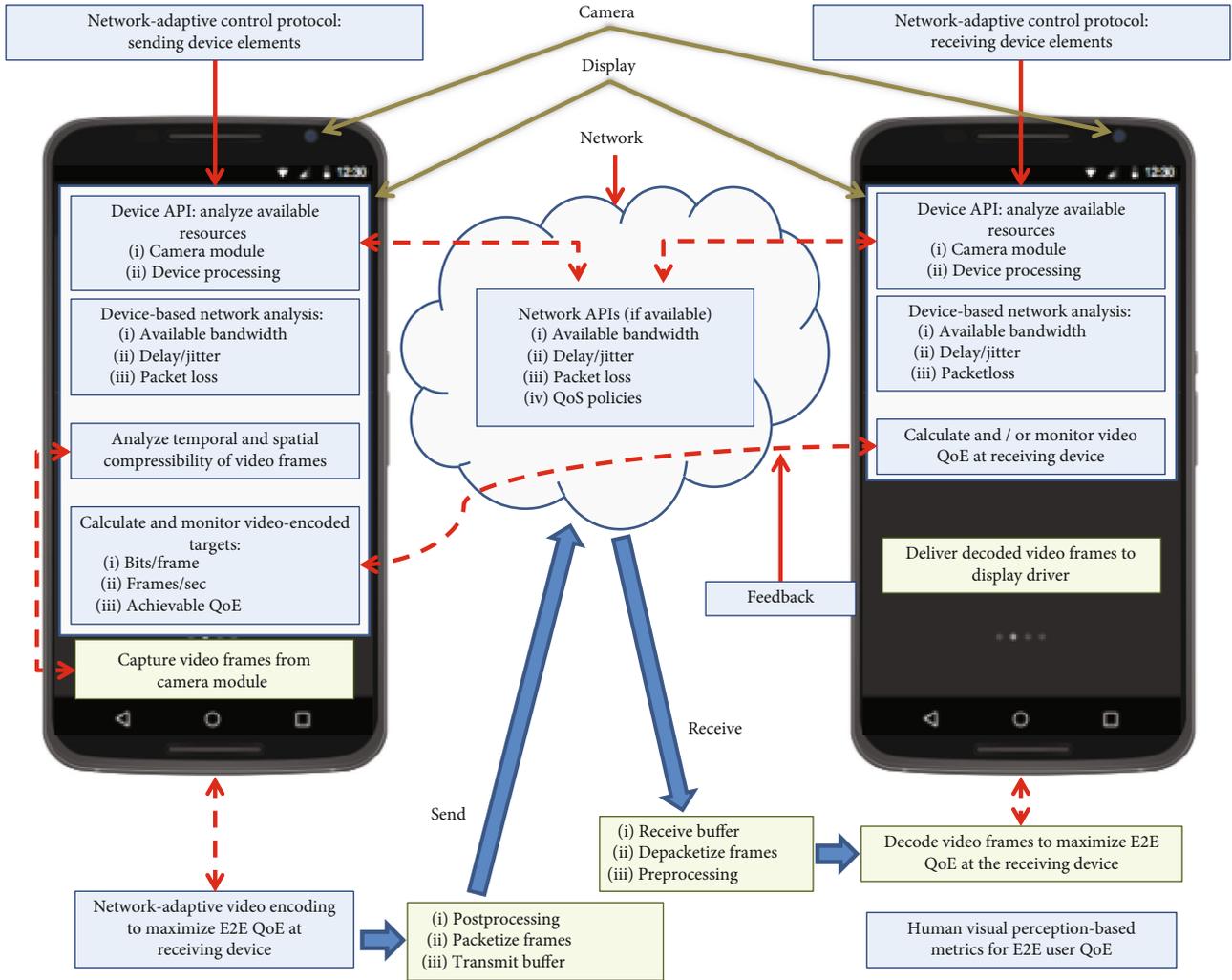


FIGURE 2: Architecture that includes all the established objectives.

camera, CPU, RAM, and iOS), characteristics of device-based network analysis (bandwidth, delay, jitter, and packet loss), and data calculated from monitoring video-encoded targets (bits/frame, frames/sec, and achievable QoE).

Figure 3 shows a generic communication protocol between two users connected to two different service providers, for the establishment of the call. Later, in Section 4, we will detail the proposed protocol for our architecture.

3.2. System Process. In order to design the architecture, we propose three basic processes. They correspond to the basic actions to establish a video communication. Each process is associated to a set of states and transitions that will be detailed later when the system state machine is explained. Figure 4 shows the relationship between the processes of the system. The register process is the start and end process of the system. It is the only process that requires the user’s intervention for executing it.

System processes, with the states of each process, are next described in detail.

3.2.1. Register Process. This process includes three states: Idle state, Registered state, and Failed state. The user, when start-

ing or ending the videoconference, is in the Idle state. From the Idle state, the user enters the Registered state where the final user with whom it will establish the communication will be identified and selected. The Failed status will be reached whenever video communication is interrupted for any reason. From the Failed state, the user passes to the Idle state where they can try again to start a new videoconference.

3.2.2. Connection Process. This process includes two states: the Active state and the Established state. The Active state is accessed after the registration phase, that is, when the connection is requested through the application used to connect to the end user. In this state, the initial information exchange of the adjustment parameters occurs, which are used by the connected users. The Active state is also reached, from the Forwarding state, in the case of a small failure during the transmission, trying to recover the transmission again before reaching the Failed state. From the Active state, users can arrive to the Failed state when it is impossible establish the connection with the final user. The Established state is accessed only from the Active state. In this state, videoconference begins. From the Established state, only the Forwarding state can be reached.

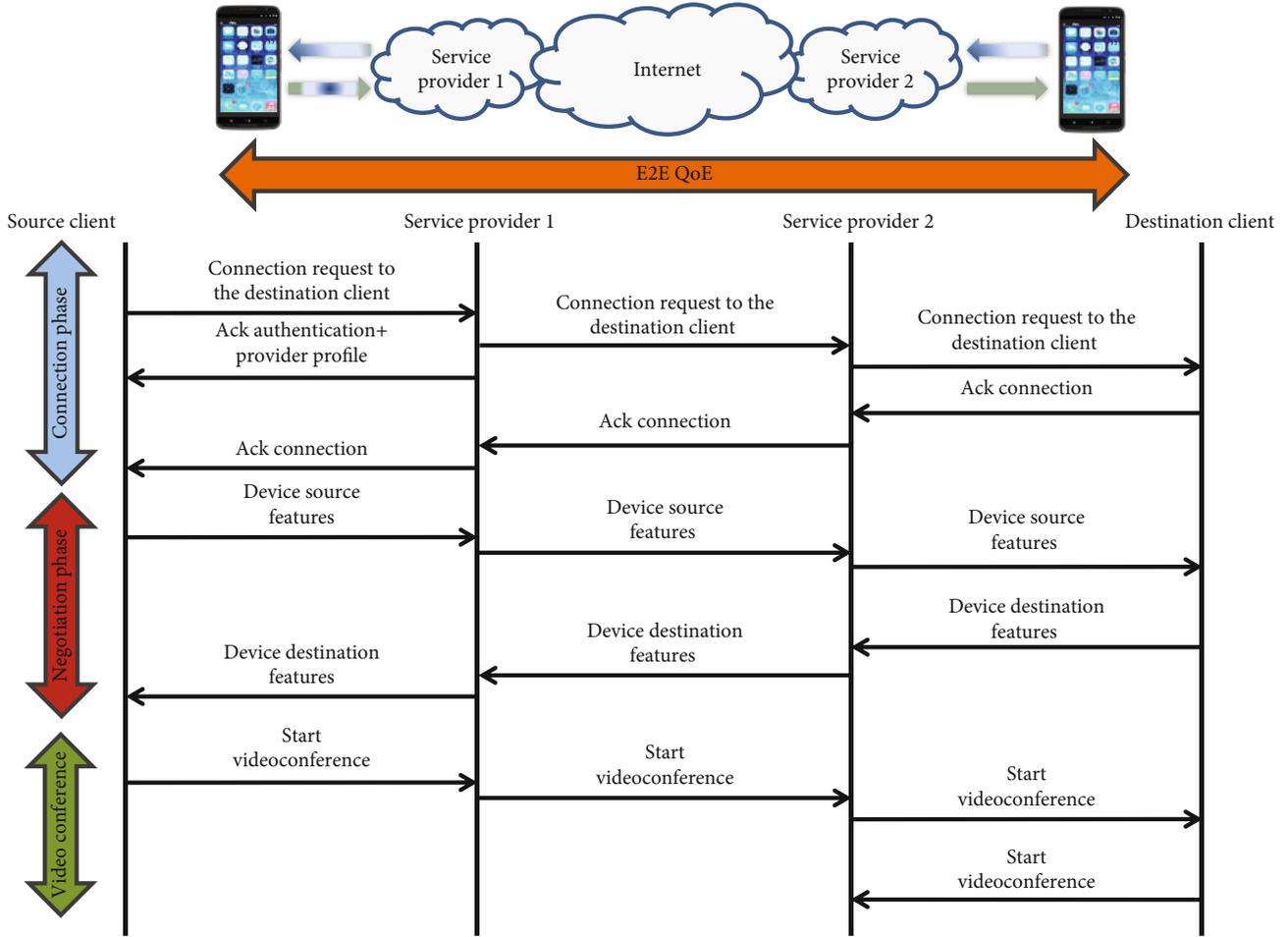


FIGURE 3: Communication protocol between two users connected to two different service providers.

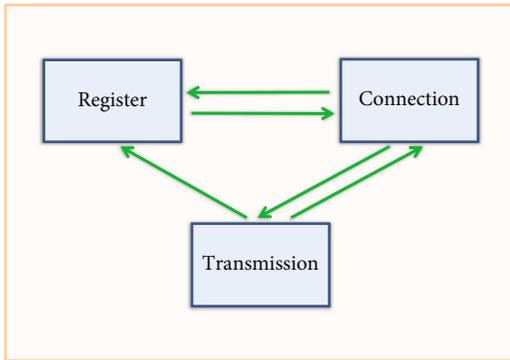


FIGURE 4: System processes of videoconference architecture.

3.2.3. *Transmission Process.* This process includes only one state: Transmission state. Users arrive at the Forwarding state from the Established state, when users have already begun the video communication. In this state, the instantaneous parameters of the devices and the network are controlled periodically. In case of need, the characteristics of the video communication are varied. In the event of a small communication failure, we can try to reenter at the Active state, and if the transmission is terminated or it is impossible to

establish communication with the end user, it is passed to the Failed state.

3.3. *Finite-State Machine.* Figure 5 shows the System Finite-State machine. We can see its different states and the transitions between states. In this section, we describe each state of the system and the conditions and events that will make the node change from one state to another inside a process.

The processes included in Figure 5 are as follows.

3.3.1. *Idle State.* At first, this is the state where the user is, before initiating access to the application to establish the videoconference, or once the videoconference is finished. Then, after the application is selected to make a videoconference, the user will go from this state to the Registered state.

3.3.2. *Registered States.* This state is accessed only from the Idle state. The user initiating the videoconference, depending on the employed software, must initiate the authentication process in the server. Once authenticated, it will search for the remote user that it wants to connect to, in its own database or in the server database. Once the end user is found, it will demand the connection with the selected end user to the server. The server tries to make contact between the users to

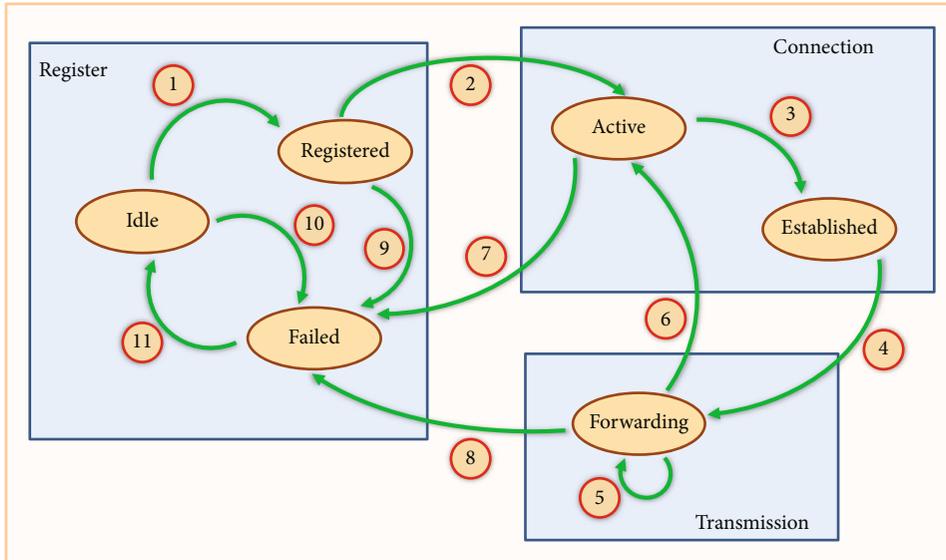


FIGURE 5: Finite-State machine.

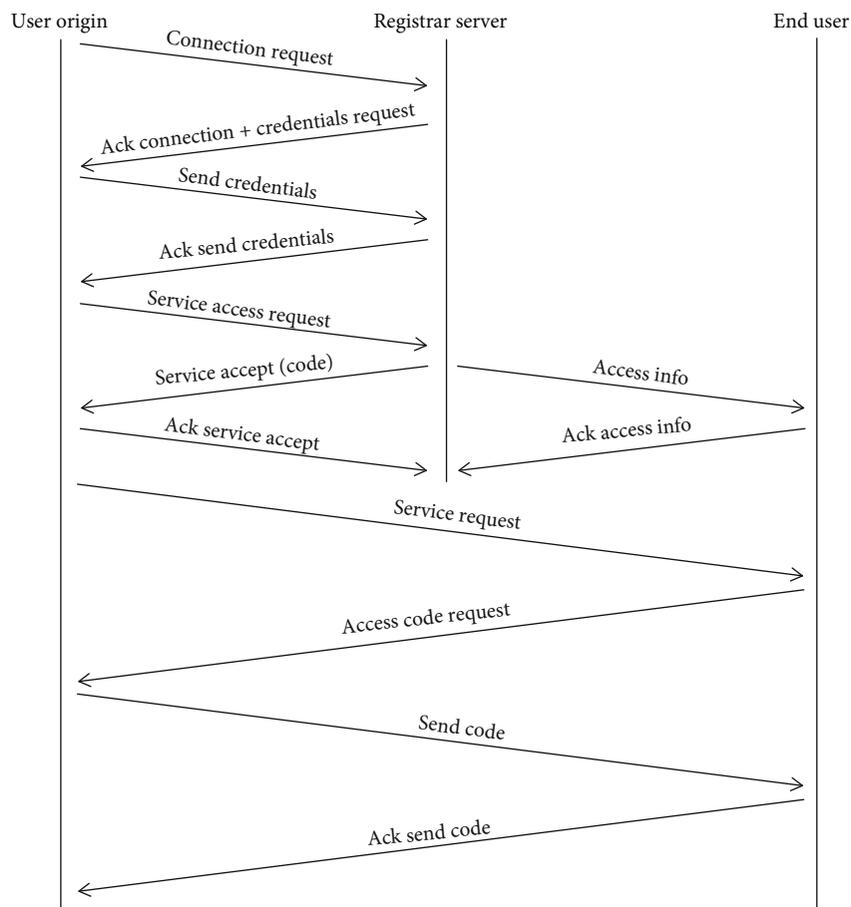


FIGURE 6: Protocol proposed for the Idle and Registered states.

establish an initial connection, and they will go to the Active state. In case the user that is initiating the call does not want to connect with any of the available users or cannot establish a connection with the end user, it will go to the Failed state.

3.3.3. *Active State.* This state can be reached from the Registered or Forwarding states. From the Active state, we can move to the Established or Failed states. Once the initial contact between the users participating in the videoconference

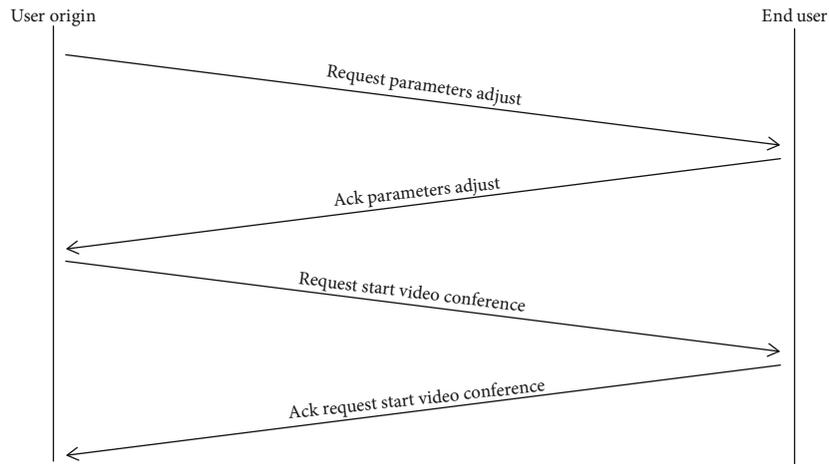


FIGURE 7: Protocol proposed for the Active state.

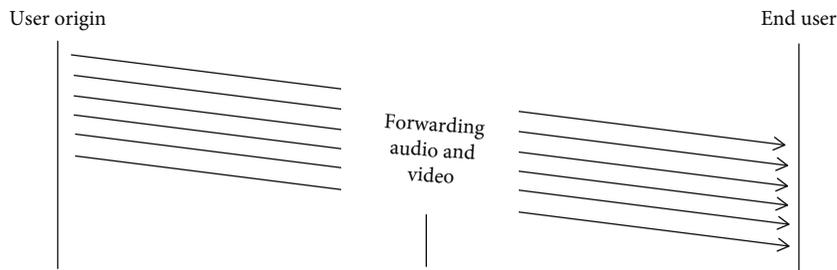


FIGURE 8: Protocol proposed for the Established state.

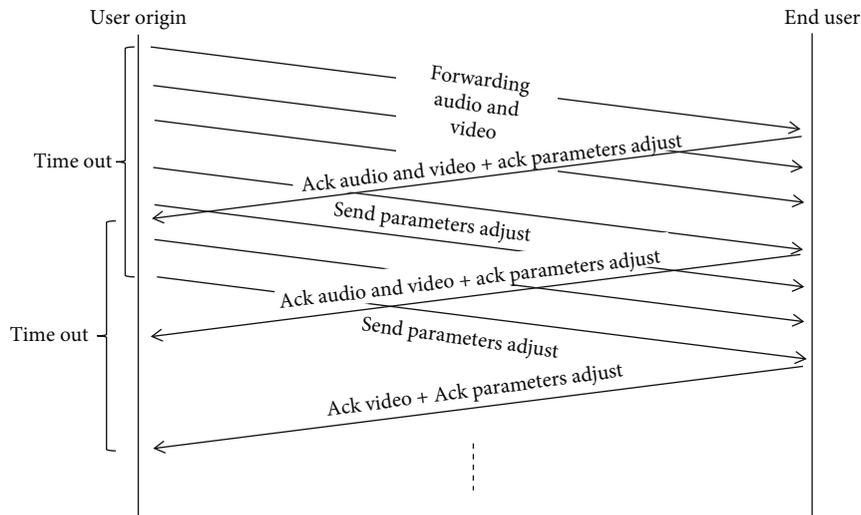


FIGURE 9: Protocol proposed for the Forwarding state when the videoconference is running correctly.

has been established, in the Active state, an exchange of parameters between the end devices of the users will be initiated, at the same time that the information of the network parameters is obtained. Using the information obtained, an algorithm to get an agreement to reach the maximum E2E QoE among the users at that moment will be applied. From this moment on, it will go to Established status. In case that one of the two users rejects or terminates the connection, or a connection agreement cannot be reached due to the

parameters of any of the user devices or of the network, it will go to the Failed state.

3.3.4. Established State. Established status can only be reached from the Active state. From the Established state, you can only move to the Forwarding state. Once the Established state is reached, the video starts from the devices of the connected users, moving to the Forwarding state.

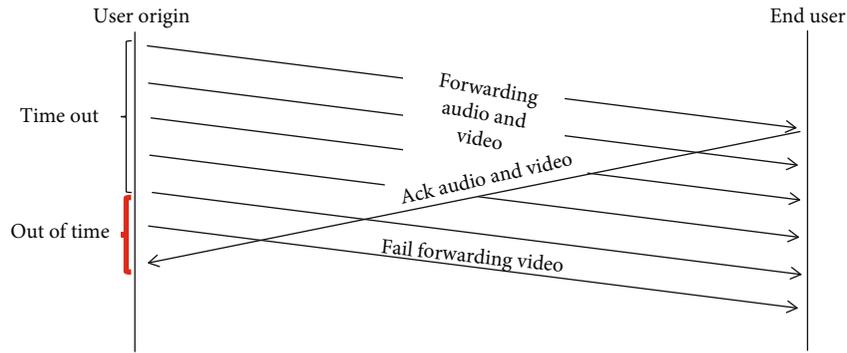


FIGURE 10: Protocol proposed for the Forwarding state when there is a problem.

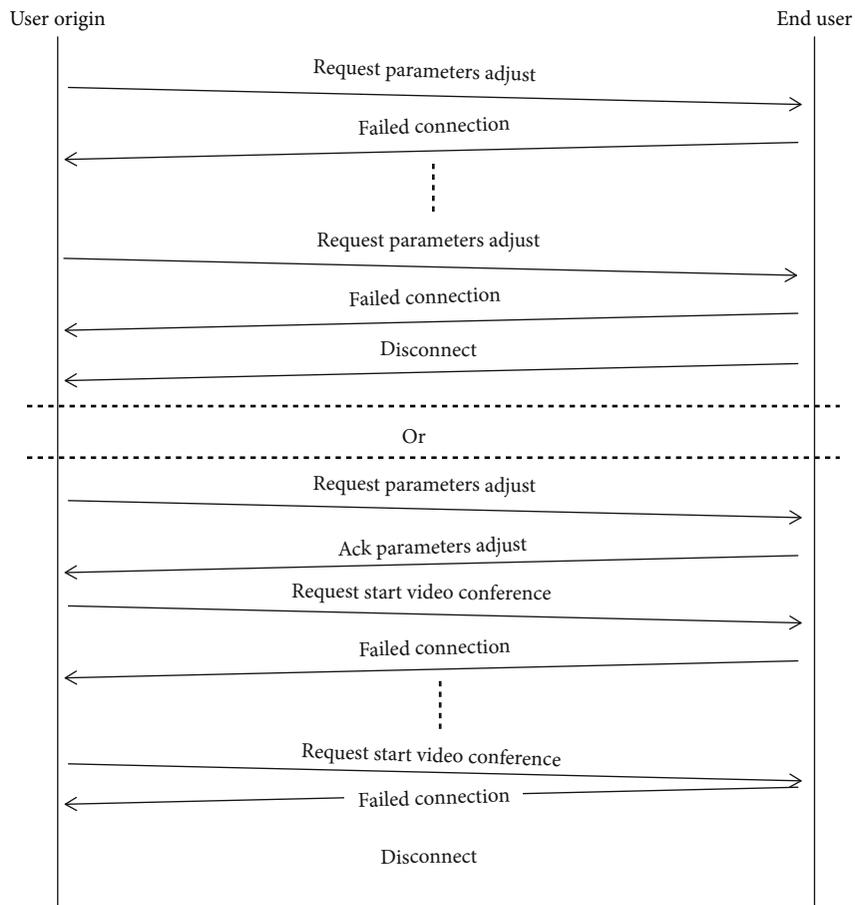


FIGURE 11: Protocol proposed for the transition from the Active state to the Failed state.

3.3.5. *Forwarding State.* The Forwarding state can be reached from the Established state and from the Forwarding state itself. Once the videoconference starts, we will remain in the Forwarding state while everything is working correctly. In this state, the final devices will continuously control the characteristics of the devices themselves and the network, so that when any variation appears, the appropriate measures are taken and the maximum E2E QoE is still maintained. We can vary the codec that was used until then, in case of a need for more compression. Periodically, an acknowledgment

message (Ack) will be exchanged (both for an appropriate videoconference reception and for the adjustment of parameters) between the devices of the users participating in the videoconference. It will establish a maximum period of time (time out) that, if exceeded, the corresponding Ack will not be received. Thus, it will be considered that the videoconference is failing. If a failure occurs, it will go back to the Active state to try to renegotiate the parameters of the devices and the network parameters and go on to relaunch the transmission. In case of not being able to establish the connection

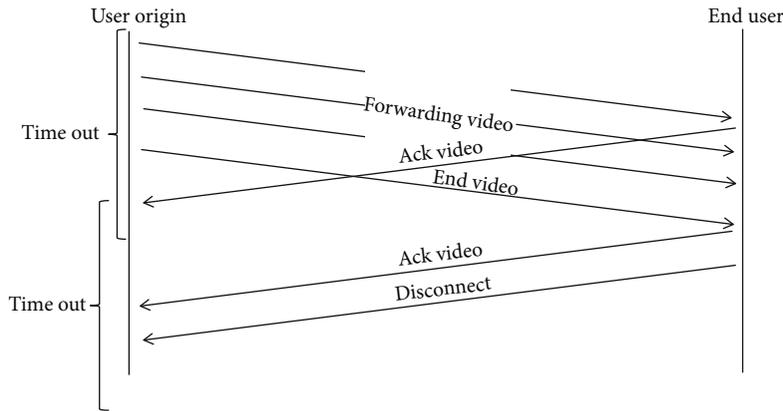


FIGURE 12: Protocol proposed for the transition from the Forwarding state to the Failed state.

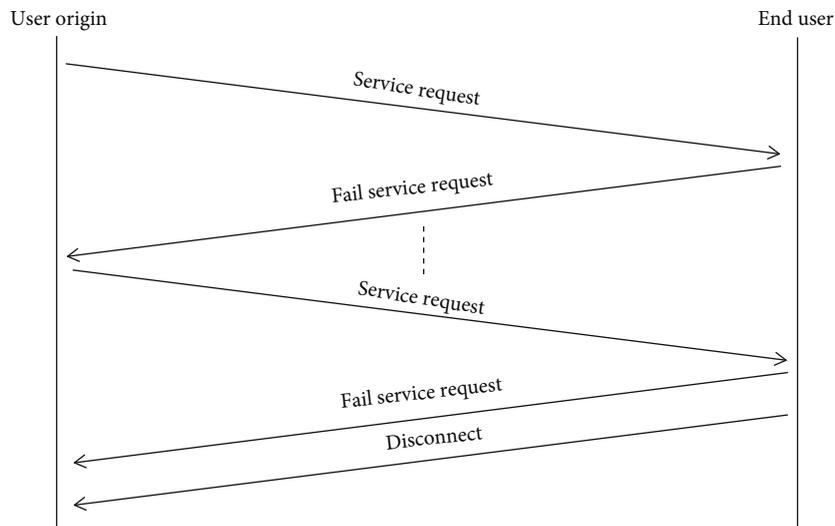


FIGURE 13: Protocol proposed for the transition from the Registered state to the Failed state.

again, it will go to the Failed state. In the event that any of the users ends the videoconference, it will go to the Failed state.

3.3.6. *Failed State.* The Failed status is reached if the videoconference did not work correctly or if one of the users decided to disconnect. The Failed state is reached from the Idle, Registered, Active, and Forwarding states. From the Failed state, it passes to the Idle state to start the whole process again.

4. Protocol Proposal

Figure 6 shows the protocol proposed for the start of the establishment of the connection. It includes the generic actions that will be carried out during the Idle and Registered states.

Figure 7 shows the proposed protocol for the Active state. In this state, users exchange characteristic parameters of their devices and also of the network, in order to achieve the transmission of the videoconference with the maximum E2E QoE.

Figure 8 shows the proposed protocol for the Established state. In this state, the video and audio transmission of the videoconference between the interlocutors begins. The transmission of audio and video will be done in both directions simultaneously, although in Figure 8, the delivery can be observed in only one direction.

Figure 9 shows the proposed protocol for the Forwarding state, but only during the correct operation of the videoconference. It can be seen that the transmission initiated during the Established state continues. Its correct operation is being controlled by the exchange of Acks. They are received before the time out expires. The transmission of audio, video, and Acks will be done in both directions simultaneously, although Figure 9 shows the transmission in only one direction.

Figure 10 shows the proposed protocol for the Forwarding state, when the videoconference stops working correctly, since the Ack is not received from the remote user within the time out. When this situation occurs, it will go back to the Active state to try to recover the transmission. The transmission of audio and video will be done in both directions

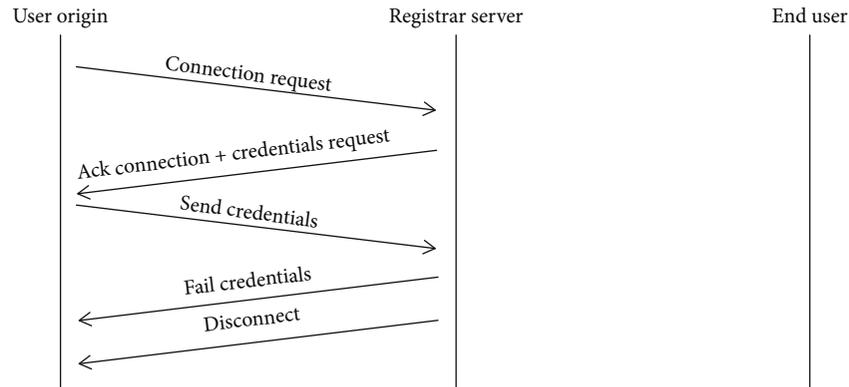


FIGURE 14: Protocol proposed for the transition from the Idle state to the Failed state.

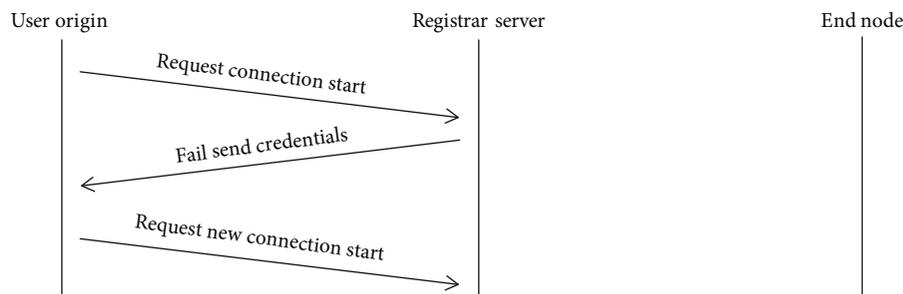


FIGURE 15: Protocol proposed for the transition from the Failed state to the Idle state.

simultaneously, although in Figure 10, the transmission can be observed in only one direction.

Figure 11 shows the proposed protocol for the transition from the Active state to the Failed state. As seen in Figure 11, the transition can occur for two reasons, after several requests for adjustment parameters that have not been answered or after several failures in the attempt to start the videoconference. The transmission of audio and video will be done in both directions simultaneously, although in Figure 11, the transmission can be observed in only one direction.

Figure 12 shows the proposed protocol for the transition from the Forwarding state to the Failed state. As shown in Figure 12, the transition occurs when one of the users involved in the videoconference decides to end it, without any transmission error. The transmission of audio and video will be done in both directions simultaneously, although in Figure 12, the transmission can be observed in only one direction.

Figure 13 shows the proposed protocol for the transition from the Registered state to the Failed state. As shown in Figure 13, the transition occurs when, once the user origin is authenticated, the connection to the end user cannot be established. When this connection attempt fails, it passes to the Failed state in which it will be disconnected.

Figure 14 shows the proposed protocol for the transition from the Idle state to the Failed state. As seen in the image, the transition occurs when, once the videoconferencing software application starts, the user origin cannot be authenticated in the server. When this authentication attempt fails, it passes to the Failed state in which it will be disconnected.

Finally, Figure 15 shows the proposed protocol for the transition from the Failed state to the Idle state. As shown in Figure 15, the transition occurs when the credentials sent by the user origin failed and when it cannot be authenticated in the server. It sends it to the Idle state, where it can try to start a new connection.

5. Performance Test in Videoconference Applications

We have made several performance tests using some of the best-known videoconference applications used in business, academic, and even personal areas. We have made multiple videoconferencing sessions with Adobe Connect, Webex, and Skype.

The topology used during the test is shown in Figure 16. We have used two PCs with the following features: Intel Core i7-7700 3.6 Ghz, 16 GB RAM DDR4 2400 MHz, integrated network card 10/100/1000, integrated wireless network card, and Windows 10 64-bit OS. The network devices we have used were a router for accessing the Internet with a connection of 300 Mbps and two Linksys RE6500-EJ access points that support the 802.11 a, b, g, and n standards. We have also used two JIAYU smartphones model JY-S3 with an eight-core MT6752 processor at 1.7 Ghz, with 2 GB of RAM and 16 GB of internal memory and an Android 5.1 operating system.

In all the equipment, PCs, and smartphones, we installed the software to make the videoconference, which can be downloaded from web pages and as apps provided by the

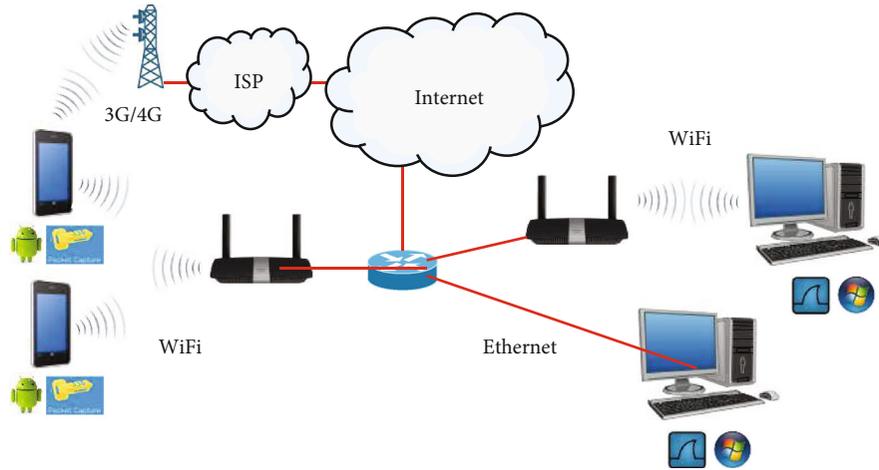


FIGURE 16: Topology used in the videoconference performance test.

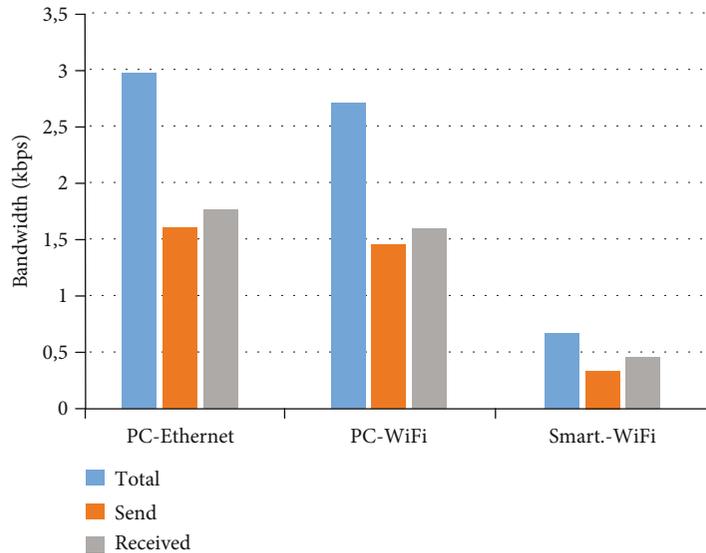


FIGURE 17: Results obtained when using Adobe Connect and smartphone destination with WiFi.

manufacturers (Adobe Connect, Cisco Webex, and Skype). We have also installed the software that allows us to capture the traffic sent. In the case of PCs, we have installed Wireshark, while in smartphones, we have used an app called tcpdump.

In our performance tests, we made different captures to observe the characteristics of the sent traffic with each application and with a duration of 3 minutes.

The data has been captured when the origin of the videoconference was made from different devices, PCs, or smartphones (connected by cable or wireless), and the destination was a smartphone that was connected via WiFi, 3G, or 4G.

5.1. Results Obtained When Using Adobe Connect (WiFi). In Figure 17, it can be seen that when the transmission is made from a PC, both through its Ethernet and wireless interfaces, the traffic increases (approximately 300%) with respect to the

transmission made from the smartphone. Due to these results, we consider that Adobe Connect takes into consideration the type of device from which the transmission is made, PC or smartphone, above the technology used, wired or wireless.

5.2. Results Obtained When Using Adobe Connect (3G/4G). In Figure 18, it is observed that the significant differences are more related to the type of device being used in the test, PC or smartphone, regardless of the connection technology (3G/4G) used in the target device as it happens in Figure 17. When 3G or 4G is used, unlike when both ends employ WiFi technology, the bandwidth consumption is very asymmetric. The bandwidth used by a PC multiplies approximately by 4 the bandwidth used by the mobile.

5.3. Results Obtained When Using Cisco Webex (WiFi). Figure 19 shows the bandwidth consumption when the target device is a smartphone connected via WiFi. In general, there

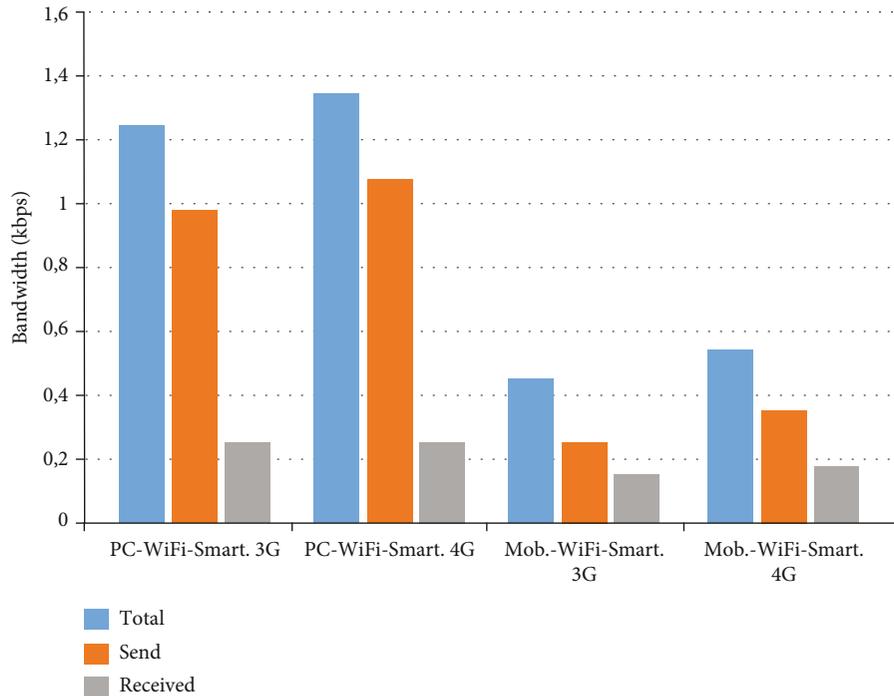


FIGURE 18: Results obtained when using Adobe Connect and smartphone destination with 3G/4G.

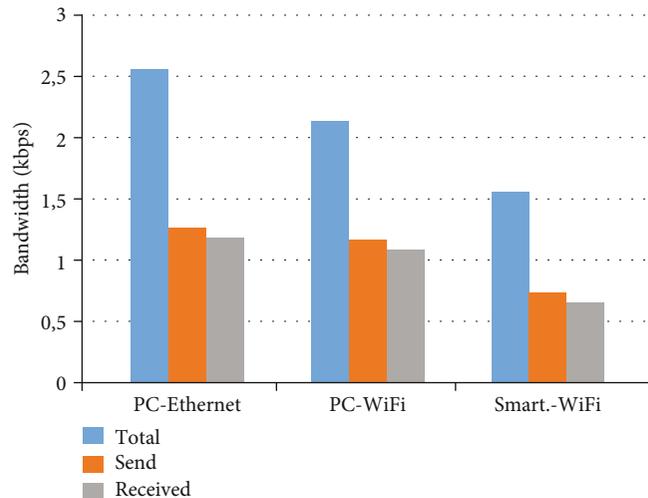


FIGURE 19: Results obtained when using Cisco WebEx and smartphone destination with WiFi.

are no significant differences when changing the device or technology. In the experimental data presented, it was observed that we obtained lower bandwidth consumption results when we use WiFi technology compared to when we use the Ethernet interface.

5.4. Results Obtained When Using Cisco Webex (3G/4G).

Figure 20 shows the results when the target device is connected using 3G/4G technologies. As can be seen in Figure 20, it does not show great differences. The consumption of bandwidth is slightly asymmetric when the transmission is made from a mobile phone connected by WiFi to a

mobile phone connected by 3G or 4G; the smartphone connected via WiFi tends to consume less.

5.5. *Results Obtained When Using Skype (WiFi)*. Figure 21 shows the results obtained when we used the Skype application. The results are very similar to those obtained when we used Adobe for the transmission. Significant differences can be observed when using a PC or a smartphone. In the case of establishing the videoconference between a PC and a smartphone or between a smartphone and another smartphone, the transmission is very asymmetric. When the transmission is made between two smartphones, there is a greater bandwidth saving, reaching a reduction of 80%.

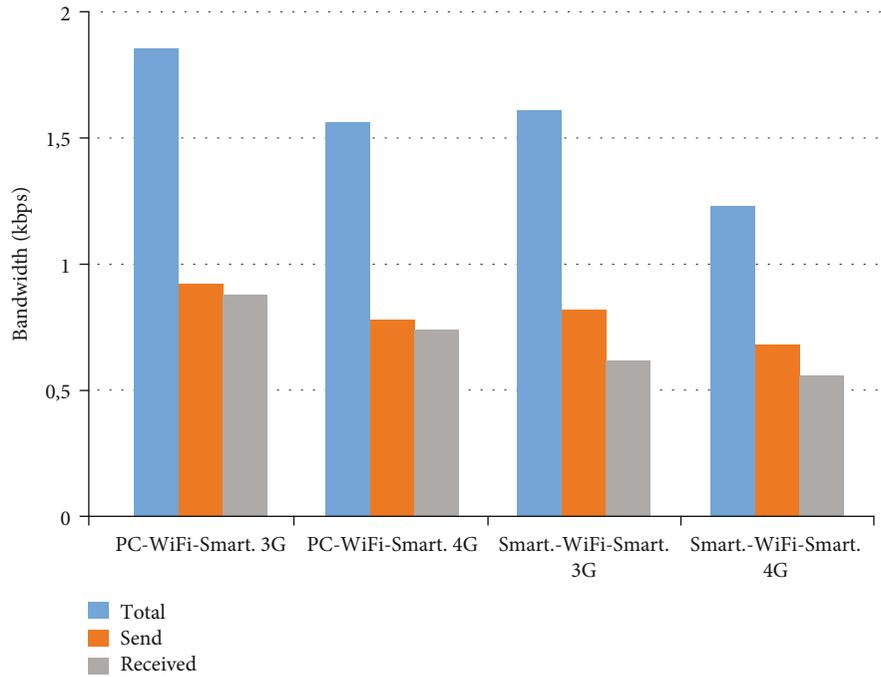


FIGURE 20: Results obtained when using Cisco WebEx and smartphone destination with 3G/4G.

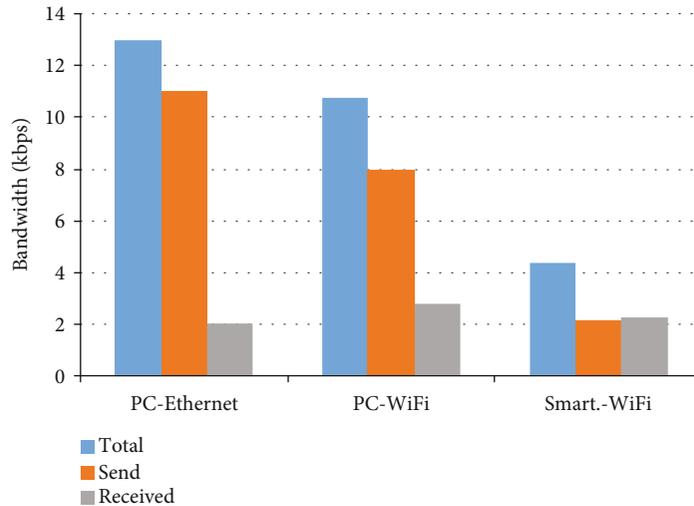


FIGURE 21: Results obtained when using Skype and smartphone destination with WiFi.

5.6. *Results Obtained When Using Skype (3G/4G).* In Figure 22, it can be seen that bandwidth consumption is greater when a PC is used than when a smartphone is used. It is also observed that bandwidth consumption when using 4G technology is slightly higher than when using 3G.

5.7. *Comparison of Cisco Webex vs. Adobe Connect in Terms of Sent Packets.* In Figure 23, we observed the number of transmitted packets in both Cisco Webex and Adobe Connect. A correlation can be seen between the bandwidth consumption and the number of transmitted packets in each of the experimental assumptions. In the case of using the fragmentation of packages, depending on the technology or type

of device, several differences have been observed. An improvement could be achieved by modifying the fragmentation of the packets depending on the technologies or devices.

6. Performance Test of the Developed Application

6.1. *Developed Application.* The operation mode of the application used for testing the prototype proposed in this paper is shown in Figure 24. When starting the application, the login page is shown first. In case the user has no account, the application will enter into a sign up login activity. In order to be

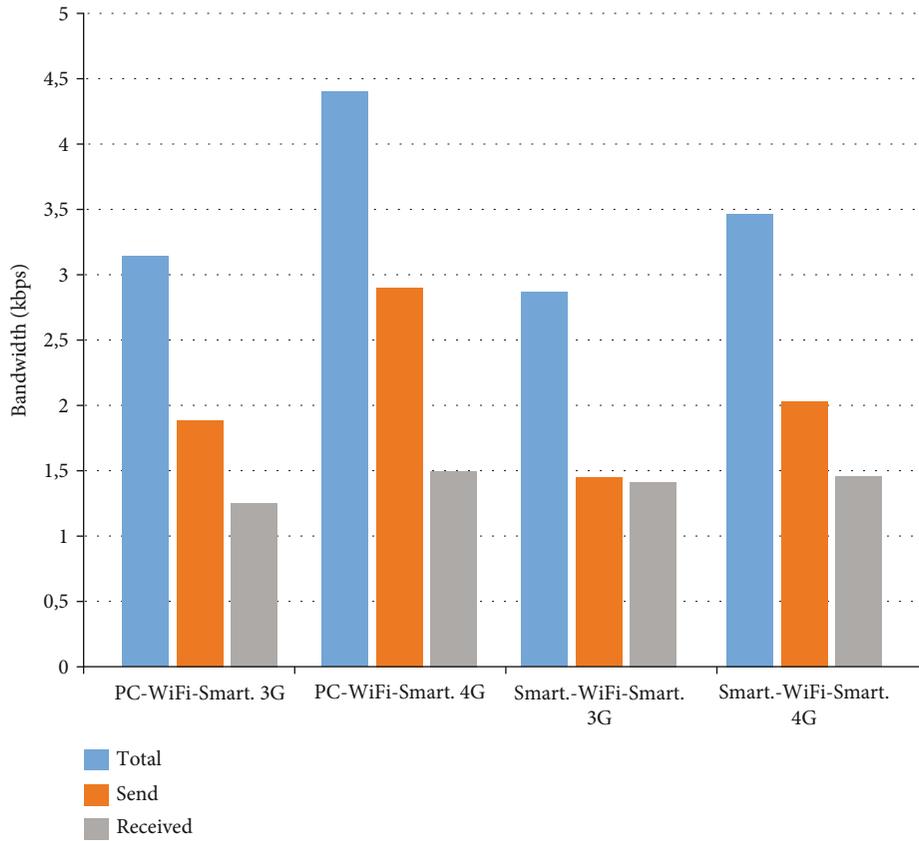


FIGURE 22: Results obtained when using Skype and smartphone destination with 3G/4G.

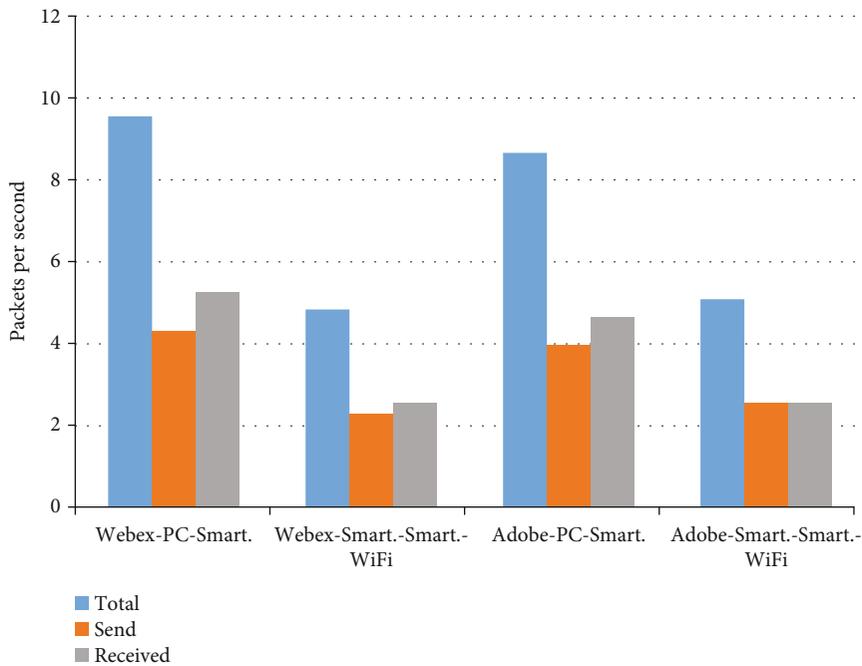


FIGURE 23: Comparison of Cisco WebEx vs. Adobe Connect in terms of sent packets.

registered on the application, it is necessary for the application to have permission to read the Phone_state. When the user already has an account, the user can login. Once the user

is logged in, the main screen is shown and it contains the About button, Contacts button, and field to enter a user and call him. The Contacts button will show the contacts of

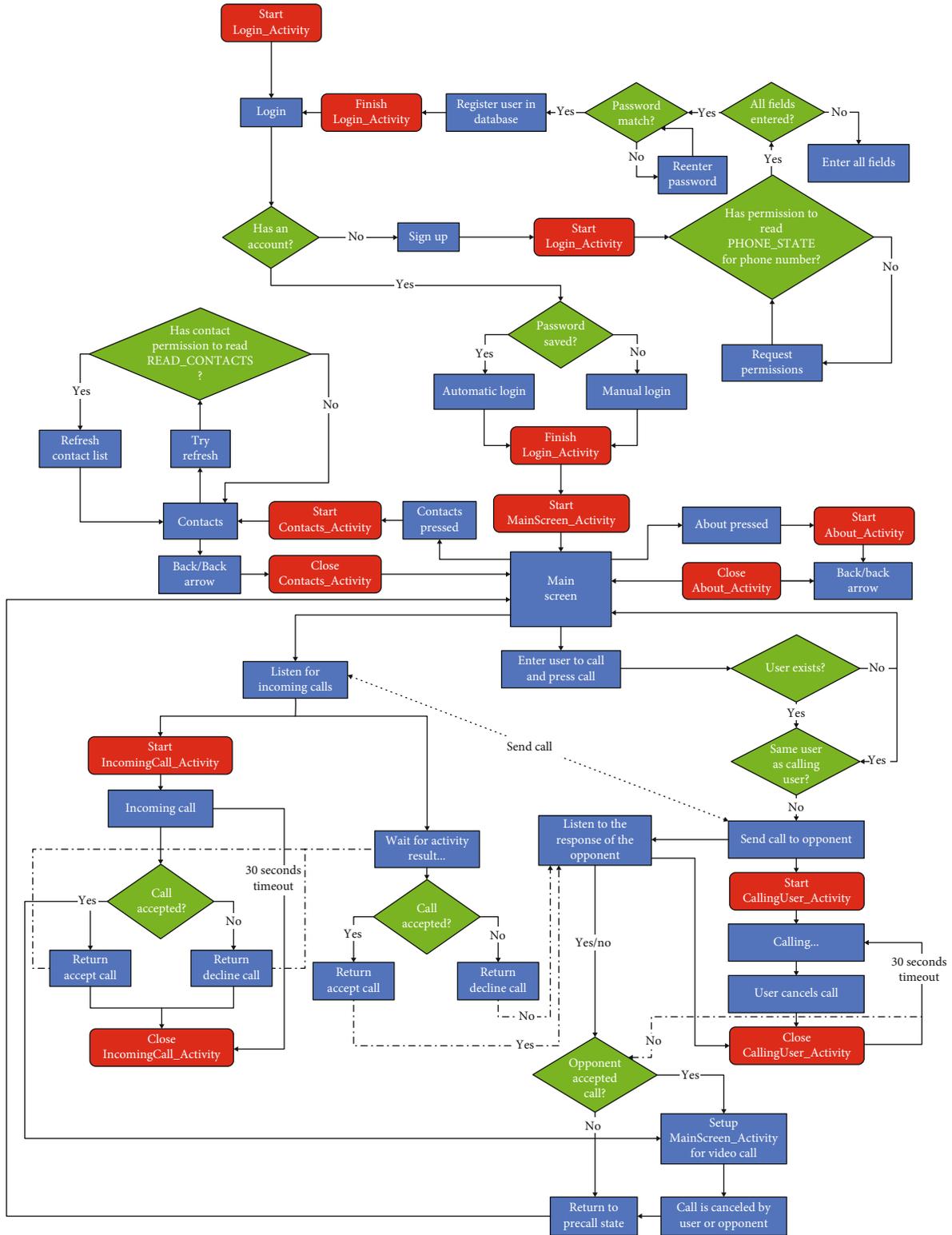


FIGURE 24: Block diagram of application operation.

the mobile phone, so the application needs permission to read them from the mobile phone. When the field of user to be called is filled up and then the call button is pressed, the application will check if the user requested is correct and available.

When a user tries to initiate a call, the application will initiate the activity Start_Calling_User, which will be cancelled in case the user cancels it manually or in the case of running out of time after 30 seconds. The user that receives the call will have the option to accept or reject the call. If the

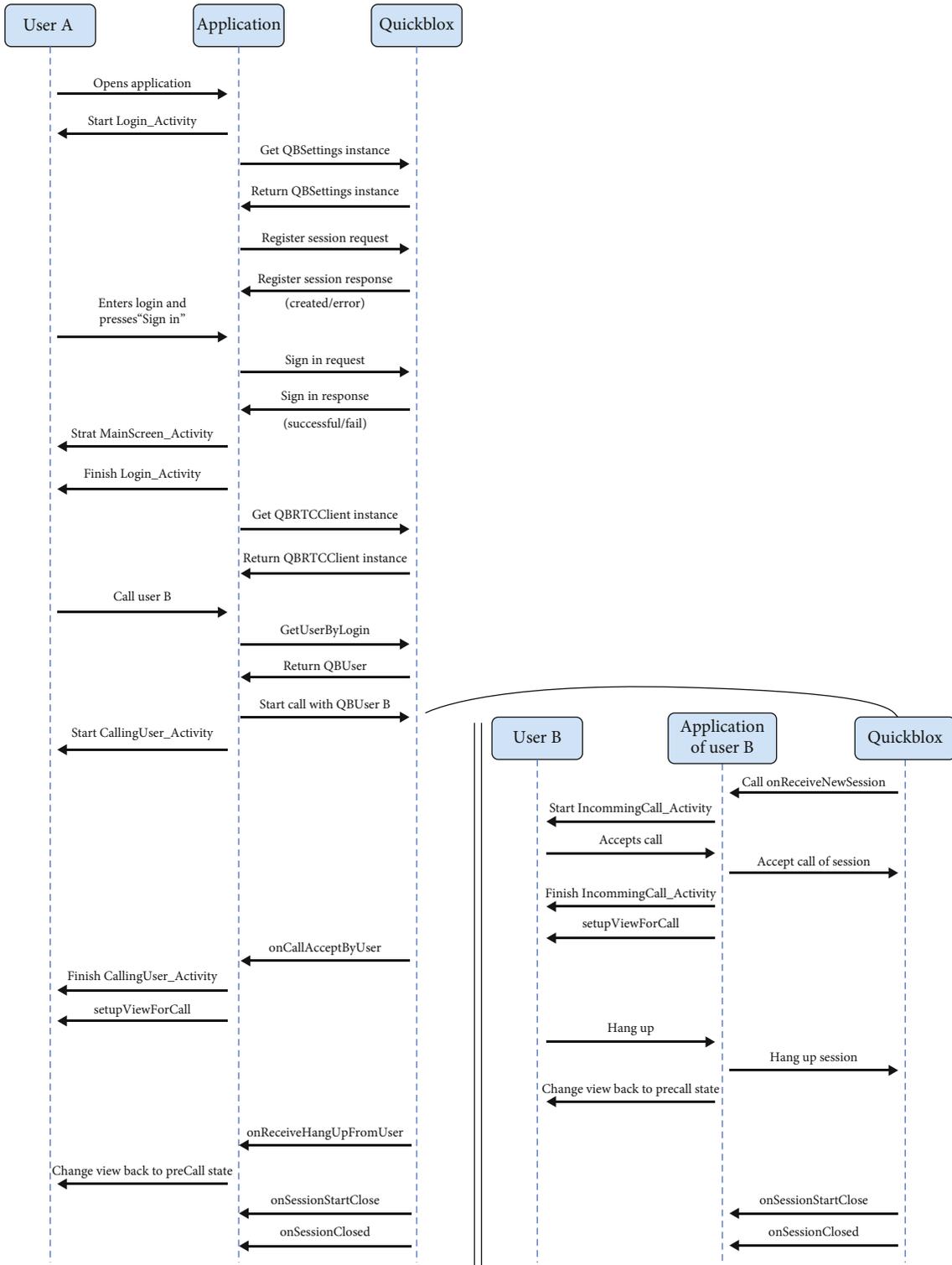


FIGURE 25: Application protocol flow diagram.

communication is established, a MainScreen_Activity will be initiated. Finally, when the call ends, the application will return to precall state.

Figure 25 shows how the application protocol works since a user (user A) opens the application until a conversation is started (with user B), maintained, and finalized. When

user A opens the application, the app will respond with the StartLogin_Activity and will communicate with the Quickblox to get a QBSettings instance and get information about whether the user is registered or not. After that, the user is able to login. When the user presses the login button, the application will send to the Quickblox a sign-in request and

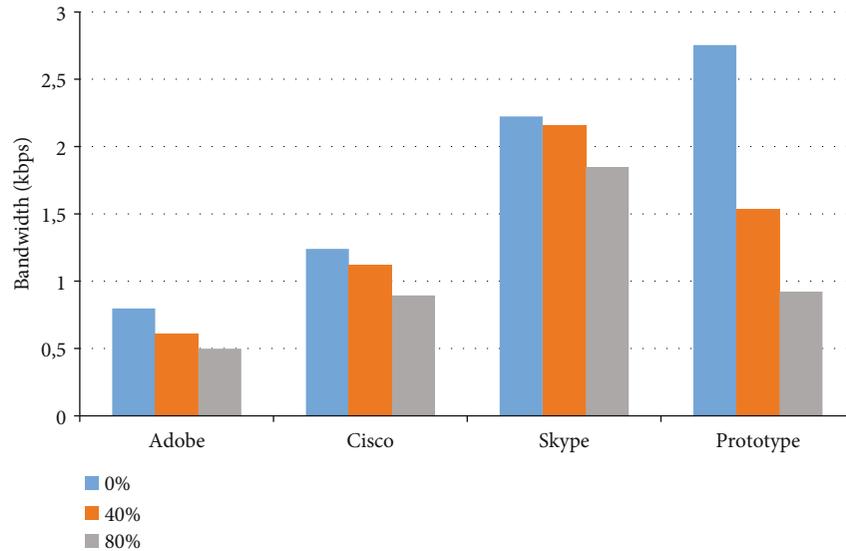


FIGURE 26: Results of scenario 1.

it will respond with a successful/failed state. If the user's login is successful, the application will show the Main Screen and will finish the login activity. The application will get a QBRTCCClient instance from the Quickblox. Now the calls are available.

If user A wants to call user B, the user will ask for the application for calling user B, and the application will ask Quickblox for the QBUser of user B. With that information, the application of user A is able to call user B, passing through the Quickblox. The Quickblox communicates to the application of user B that is receiving a call. User B accepts the call and will Setup View for call. User A will receive an onCallAcceptByUser B and will also Setup View for call.

In case one of the users hangs up the call, the application of the user that is hanging up will communicate this to the Quickblox which sends it to the other application. Both applications change view back to the precall state and close the session.

6.2. Test. In this section, the tests performed with the developed application are presented. We have developed a basic application for videoconferencing which implements the features and characteristics of the algorithm and protocol that have been defined and explained in Section 3 and 4. This application allows us to show the validity of our proposal compared to other commercial applications. The main goal is to show that this work improves the QoE of the videoconference users.

The comparison has been performed on three different scenarios:

- (1) *Scenario 1.* It focuses on the analysis of the resources of the local device, such as the CPU and the RAM combined with the smartphone characteristics (resolution available, camera features, etc.). Following this information, the implementation of our algorithm

adapts the videoconference transmission to guarantee the best conditions for the user

- (2) *Scenario 2.* It focuses on the analysis of the network status from the point of view of the local device. QoS parameters (loss packet, delay, jitter, and bandwidth availability) are observed. When changes in the network conditions happen, the algorithm acts to achieve the best possible QoE
- (3) *Scenario 3.* Finally, in this last scenario, the whole network is analyzed through the capabilities of SDN. The developed protocol links the mobile device with a network managed by SDN, in order to optimize the path of the video transmission used in the videoconference and to minimize the end-to-end delay, jitter, and packet loss

In the next subsections, the measurements from our developed application will be referenced as prototype.

6.2.1. Scenario 1. The experimental set used in this scenario is the same as the one presented in Figure 16, described in Section 5. Each test has been repeated 10 times, and the average has been calculated. The obtained values are presented in Figure 26. Figure 26, we can see how the developed prototype has been able to adapt the bandwidth when the available resources of the CPU and RAM have changed.

In order to perform the measurements in this scenario, in addition to the prototype videoconference application, another experimental app has been developed. The goal of this last application is to spend the resources of the device. The application runs an infinite loop making some random mathematical calculations and can be adjusted to manage the amount of resources of the device.

The first column of each application shows the bandwidth used for the videoconference when there are no other applications running in the same device. For the second

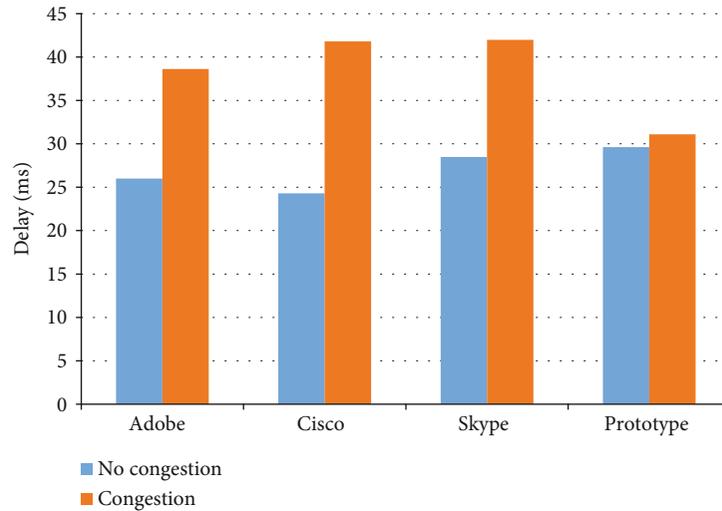


FIGURE 27: Results of scenario 2.

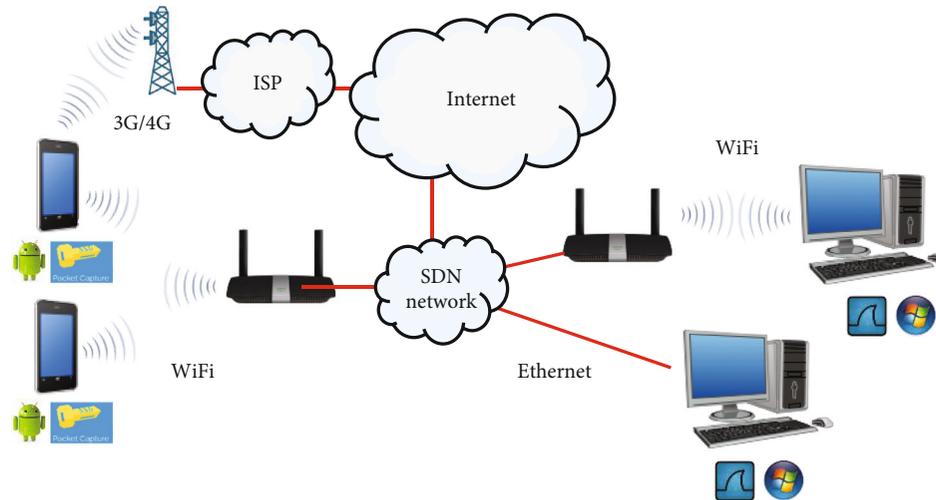


FIGURE 28: Topology used in scenario 3.

and third experimental conditions, shown at the second and third columns, the resources consumed by the application described above were 40% and 80%, respectively.

From the results, we can see how our prototype application gets worse results when the resources of the device are free (0%). But, when the resources decrease, the algorithm used for the prototype is able to adapt in order to reduce the bandwidth consumption, while the commercial solutions show similar results in the three experimental conditions.

6.2.2. Scenario 2. In this scenario, we use the same topology as in scenario 1, which was presented in Figure 16. As in the previous scenario, we have repeated the tests 10 times for this scenario. Now, our goal is to observe the behavior of our application when the local network parameters (loss packet, delay, jitter, and bandwidth availability) change. Basically, we increase the traffic that is sent to the network.

In order to increase network traffic, to achieve congestion, we have developed an application that generates traffic.

In addition, the application, which runs on both ends of the network, allows measuring the latency, based on the exchange of standard ICMP packets, between the final devices that perform the videoconference.

As can be seen in Figure 27, commercial applications have worse latency when congestion appears in the network, because they do not make any type of adaptation in the new situation generated, while our prototype adapts and maintains low levels of latency.

6.2.3. Scenario 3. Figure 28 shows the topology used to perform the tests in scenario 3. In this scenario, we have replaced the router used in previous scenarios by an SDN network.

The SDN network is made up of different devices, including an SDN controller and several layer 3 switches (model HP ProCurve 3500yl-24G-PWR Intelligent Edge). These switches support the OpenFlow protocol and allow us to work with SDN.

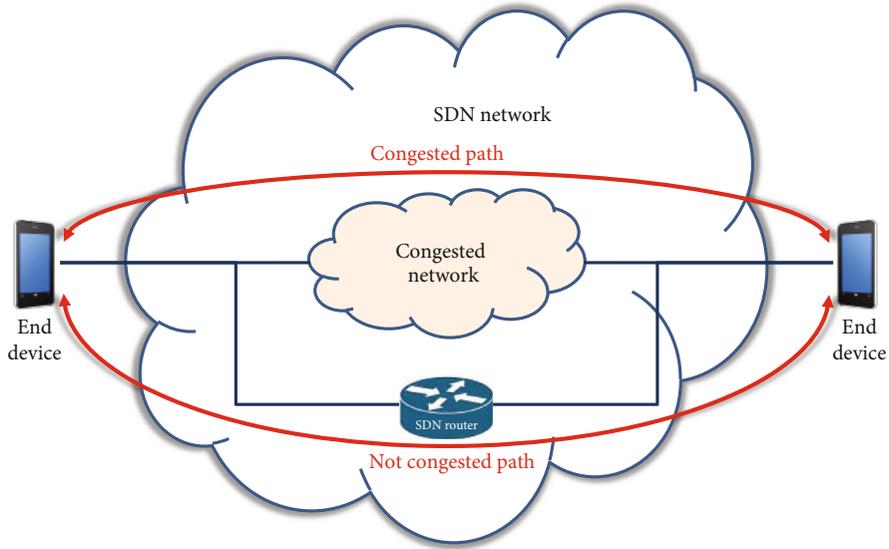


FIGURE 29: SDN network.

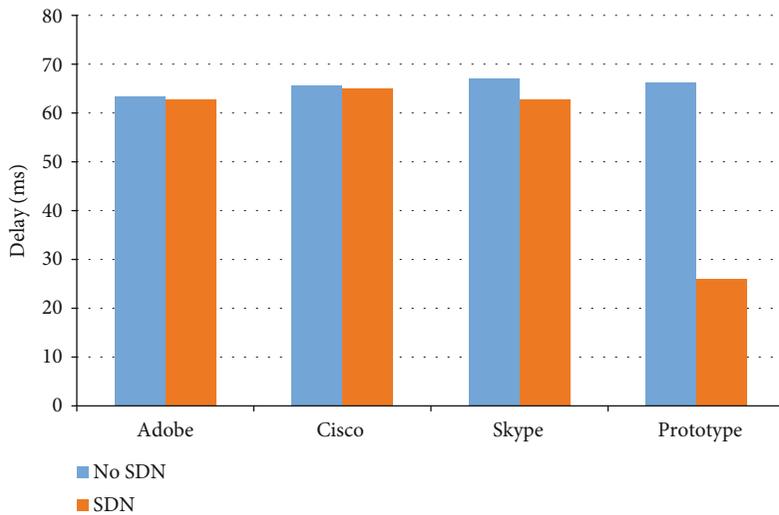


FIGURE 30: Results of scenario 3.

Our target in this scenario is to observe the correct functioning of our proposal when there is congestion in the transport network. As can be seen in Figure 29, we have created an SDN network using two basic routes. One of the routes uses a path that crosses a congested network, and the path of the other route avoids the congestion (Congested path and Not congested path).

The SDN controller has been programmed to communicate to mobile devices with the SDN network. For this purpose, an extension of the OpenFlow protocol has been developed. From our prototype, which we have installed on mobile devices, we launched an SDN activation request to the controller. From that moment, the SDN controller manages the traffic by transmitting the packets through the noncongested links.

As can be seen in Figure 30, commercial solutions, since they do not support SDN technology, do not change their behavior in an SDN network.

However, when using our prototype together with SDN, it can be clearly seen how the latency decreases in a very significant way. The SDN controller sends the traffic through an alternative path, completely free of congestion.

All the tests that are presented in our work have been carried out on a network that meets special requirements. Only the videoconference traffic stream was being sent. By using SDN, we added a new stream, which was sent at the same time as the videoconference stream. This new stream was always controlled for us. The newly added stream was sent to saturate the available bandwidth in the network and to automatically readjust the transmission parameters of the videoconference to the optimal ones to achieve the highest QoE under those conditions. If instead of controlling the traffic sent, the network is transmitting multiple streams, you can use FluidRAN [31] or LayBack [32] architectures, which have been shown to present substantial gains in handling streams by statistical multiplexing.

7. Conclusion

In this paper, we have presented a new architecture and a new protocol to optimize videoconferencing. First, we have defined an E2E QoE Management Scheme. This scheme utilizes correlation of both subjective and objective E2E QoE with received real-time video data (stream header and/or video signal), application-level QoS measurements, and network-level QoS measurements. We define real-time device-based and network-based feedback control mechanisms that can be used to adjust E2E QoE, and we present our proposal of architecture for videoconference. We propose three basic processes, which correspond to the basic actions to establish a videoconference (register, connection, and transmission). Later, we propose a Finite-State Machine, and we present and define the different states. After defining the system, we present our new protocol for videoconferencing.

Various videoconferencing applications, such as Adobe Connect, Cisco Webex, and Skype, have been tested. Data about bandwidth, packets/s, and delay have been collected and compared with the results of our prototype. Results show that when the resources of the device to be used for the application decrease, the algorithm used for the prototype is able to adapt in order to reduce the bandwidth consumption, while the commercial solutions are not able to do this. Regarding delay, commercial applications have worse latency when congestion appears in the network, while our prototype adapts and maintains low levels of latency.

Finally, commercial solutions do not change their behavior regardless of whether or not they use SDN technology. However, our prototype with SDN shows that latency decreases in a very significant way.

This paper is part of the dissertation of Jose M. Jimenez [33]. In future work, we will add more functionality in terms of codec selection, video conversion formats, and the selection of type of device among others.

Data Availability

There is no database of measurements in this research work.

Conflicts of Interest

The authors declare that they have no conflict of interest.

Acknowledgments

This work has been supported by the “Ministerio de Economía y Competitividad” in the “Programa Estatal de Fomento de la Investigación Científica y Técnica de Excelencia, Subprograma Estatal de Generación de Conocimiento” within the project under Grant TIN2017-84802-C2-1-P.

References

- [1] J. Lloret, M. García, and F. Boronat, *IPTV: la televisión por Internet*, vol. 230, Editorial Vértice, Málaga, España, 2008.
- [2] J. Frnda, J. Nedoma, J. Vanus, and R. Martinek, “A hybrid QoS-QoE estimation system for IPTV service,” *Electronics*, vol. 8, no. 5, p. 585, 2019.
- [3] R. Huang, X. Wei, Y. Gao, C. Lv, J. Mao, and Q. Bao, “Data-driven QoE prediction for IPTV service,” *Computer Communications*, vol. 118, pp. 195–204, 2018.
- [4] A. D. Tesfamicael, V. Liu, E. Foo, and B. Caelli, “Modeling for a QoS-efficient real-time video conferencing system in the cloud,” in *2018 26th International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*, pp. 1–6, Split, Croatia, 2018.
- [5] N. Rao, A. Maleki, F. Chen et al., “Analysis of the effect of QoS on video conferencing QoE,” in *2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC)*, pp. 1267–1272, Tangier, Morocco, 2019.
- [6] J. Mao, R. Huang, X. Wei, Q. Bao, Z. Dong, and Y. Qian, “IPTV user QoE prediction based on the LSTM network,” in *2017 9th International Conference on Wireless Communications and Signal Processing (WCSP)*, pp. 1–6, Nanjing, China, 2017.
- [7] J. M. Jiménez, A. Canovas, A. López, and J. Lloret, “A new algorithm to improve the QoE of IPTV service customers,” in *2015 IEEE International Conference on Communications (ICC)*, pp. 6990–6995, London, UK, 2015.
- [8] G. M. Su, X. Su, Y. Bai, M. Wang, A. V. Vasilakos, and H. Wang, “QoE in video streaming over wireless networks: perspectives and research challenges,” *Wireless Networks*, vol. 22, no. 5, pp. 1571–1593, 2016.
- [9] M. Garcia, A. Canovas, M. Edo, and J. Lloret, “A QoE management system for ubiquitous IPTV devices,” in *2009 Third International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies*, pp. 147–152, Sliema, Malta, 2009.
- [10] S. Shen, “Efficient SVC multicast streaming for video conferencing with SDN control,” *IEEE Transactions on Network and Service Management*, vol. 16, no. 2, pp. 403–416, 2019.
- [11] J. M. Jimenez, O. Romero, A. Rego, A. Dilendra, and J. Lloret, “Study of multimedia delivery over software defined networks,” *Network Protocols and Algorithms*, vol. 7, no. 4, 2016.
- [12] A. Chakraborti, S. O. Amin, B. Zhao, A. Azgin, R. Ravindran, and G. Wang, “ICN based scalable audio-video conferencing on virtualized service edge router (VSER) platform,” *Proceedings of the 2nd International Conference on Information-Centric Networking - ICN '15*, 2015, pp. 217–218, New York, NY, USA, 2015.
- [13] M. H. Hajiesmaili, L. T. Mak, Z. Wang, C. Wu, M. Chen, and A. Khonsari, “Cost-effective low-delay design for multiparty cloud video conferencing,” *IEEE Transactions on Multimedia*, vol. 19, no. 12, pp. 2760–2774, 2017.
- [14] J. Jang-Jaccard, S. Nepal, B. Celler, and B. Yan, “WebRTC-based video conferencing service for telehealth,” *Computing*, vol. 98, no. 1-2, pp. 169–193, 2016.
- [15] R. Bestak and J. Hlavacek, “A Videoconferencing System Based on WebRTC Technology,” in *Computer Networks*, P. Gaj, A. Kwiecień, and M. Sawicki, Eds., vol. 718 of CN 2017. Communications in Computer and Information Science, Springer, 2017.
- [16] M. Pasha, F. Shahzad, and A. Ahmad, “Analysis of challenges faced by WebRTC videoconferencing and a remedial architecture,” in *2016 International Journal of Computer Science and Information Security*, vol. 14, 2016no. 10.

- [17] P. Gusev and J. Burke, "NDN-RTC: real-time videoconferencing over named data networking," in *Proceedings of the 2nd International Conference on Information-Centric Networking - ICN '15*, pp. 117–126, New York, NY, USA, 2015.
- [18] School of Computing, Telkom University, Bandung, Indonesia, K. Sambath, M. Abdurahman, and V. Suryani, "High quality of service video conferencing over IMS," *International Journal of Information and Education Technology*, vol. 6, no. 6, pp. 470–476, 2016.
- [19] M. A. Hossain and J. I. Khan, "Dynamic MCU placement for video conferencing on peer-to-peer network," in *2015 IEEE International Symposium on Multimedia (ISM)*, pp. 144–147, Miami, FL, USA, 2015.
- [20] A. Khalifeh, M. A. Al-Tae, F. AlAbsi, S. Alrawi, and A. Murshed, "A videoconferencing platform for eHealth services in Jordan," in *2016 3rd Middle East Conference on Biomedical Engineering (MECBME)*, pp. 98–101, Beirut, Lebanon, 2016.
- [21] A. Taylor, G. Morris, J. Pech, S. Rechter, C. Carati, and M. R. Kidd, "Home telehealth video conferencing: perceptions and performance," *JMIR mHealth uHealth*, vol. 3, no. 3, article e90, 2015.
- [22] M. L. Mat Kiah, S. H. al-Bakri, A. A. Zaidan, B. B. Zaidan, and M. Hussain, "Design and develop a video conferencing framework for real-time telemedicine applications using secure group-based communication architecture," *Journal of Medical Systems*, vol. 38, no. 10, p. 133, 2014.
- [23] M. K. Mishra, S. Mukhopadhyay, and G. P. Biswas, "Architecture and secure implementation for video conferencing technique," in *Proceedings of the 2015 Third International Conference on Computer, Communication, Control and Information Technology (C3IT)*, pp. 1–6, Hooghly, 2015.
- [24] M. Pattaranantakul, K. Sanguannam, P. Sangwongngam, and C. Vorakulpipat, "Efficient key management protocol for secure RTMP video streaming toward trusted quantum network," *ETRI Journal*, vol. 37, no. 4, pp. 696–706, 2015.
- [25] T. Zhao, Q. Liu, and C. W. Chen, "QoE in video transmission: a user experience-driven strategy," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 1, pp. 285–302, 2017.
- [26] S. N. B. Gunkel, M. Schmitt, and P. Cesar, "A QoE study of different stream and layout configurations in video conferencing under limited network conditions," in *2015 Seventh International Workshop on Quality of Multimedia Experience (QoMEX)*, pp. 1–6, Pylos-Nestoras, Greece, 2015.
- [27] L. García, J. Lloret, C. Turro, and M. Taha, "QoE assesment of MPEG-DASH in polimedia e-learning system," in *2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pp. 1117–1123, Jaipur, India, 2016.
- [28] D. Henni, A. Ghomari, and Y. Hadjadj-Aoul, "Videoconferencing over OpenFlow Networks: an optimization framework for QoS routing," in *2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomous and Secure Computing; Pervasive Intelligence and Computing*, pp. 491–496, Liverpool, UK, 2015.
- [29] E. Yang, L. Zhang, Z. Yao, and J. Yang, "A video conferencing system based on SDN-enabled SVC multicast," *Frontiers of Information Technology & Electronic Engineering*, vol. 17, no. 7, pp. 672–681, 2016.
- [30] C. Al Hasrouty, C. Olariu, V. Autefage, D. Magoni, and J. Murphy, "SVC videoconferencing call adaptation and bandwidth usage in SDN networks," in *GLOBECOM 2017-2017 IEEE Global Communications Conference*, pp. 1–7, Singapore, Singapore, 2017.
- [31] A. Garcia-Saavedra, X. Costa-Perez, D. J. Leith, and G. Iosifidis, "Fluid RAN: optimized vRAN/MEC orchestration," in *IEEE INFOCOM 2018—IEEE Conference on Computer Communications*, pp. 2366–2374, Honolulu, HI, USA, 2018.
- [32] P. Shantharama, A. Thyagaturu, N. Karakoc, L. Ferrari, M. Reisslein, and A. Scaglione, "LayBack: SDN management of multi-access edge computing (MEC) for network access services and radio resource sharing," *IEEE Access*, vol. 6, pp. 57545–57561, 2018.
- [33] J. Herranz, "Design of an architecture and communication protocol for video transmission and videoconference, [Ph.D Thesis]," Universitat Politècnica de València, 2018.