ISSN 2355-0082

Preliminary Assessment of Telemedicine Systems using Virtual Testbed

A. A. N. Ananda Kusuma, Tahar Agastani, Muhammad Iqbal, Triawan Nugroho
Pusat Teknologi Elektronika (PTE)
Badan Pengkajian dan Penerapan Teknologi (BPPT)
Gedung Teknologi 3, Kawasan Puspiptek, Tangerang Selatan 15314, Indonesia
{ngurah.ananda, tahar.agastani, muhammad.iqbal5607, triawan.nugroho}@bppt.go.id

Accepted 30 October 2019
Approved 20 December 2019

***Abstract*—Telemedicine is the use of ICT (Information and Communication Technology) infrastructures to provide medical services. It is needed in Indonesia, particularly in remote areas where the number of medical facilities and specialist doctors are limited. BPPT has developed telemedicine systems which can provide tele-ECG, tele-USG, and tele-Consultation. Its feasibility requires some verification on its intended operations. This paper presents a virtual testbed for testing telemedicine systems, and shows some assessment results on their security and performance aspects.**

***Index Terms*— digital signature, load testing, performance, securiy, telemedicine, virtualization**

I. INTRODUCTION

Telemedicine is a system which makes use of ICT (Information and Communication Technology) infrastructures to deliver health services, without physical constraints on distance and location. This is needed in Indonesia where the distribution of specialty doctors is not uniform, mostly concentrated in big cities, and some populations live in remote areas far away from state hospitals. Fig. 1 shows an example of health services which can be provided by implementing telemedicine. It can be seen that the system consists of three parts: 1) primary/community care where patients are treated and their bio-signals are recorded by primary care doctors; 2) central system or hospital where specialty doctors can remotely analyze patients' status and diagnose; 3) ICT infrastructures, namely, communication networks, data centers, and other computing resources needed.

In general, there are various medical instruments used to capture patients' bio-signals depending on their condition, e.g. ECG (Electrocardiography), USG (Ultrasonography), stethoscope, vital sign monitor, etc. Those instruments are then connected to an aggregator, as an interface to the ICT infrastructure, and the bio-signals measured can be stored locally or in remote servers using specified standard formats. In addition to sharing medical records which include patients' bio-signals, medical doctors participating in a telemedicine session need to interact to discuss patients' status and the necessary actions; thus, the system needs to provide teleconference for tele-Consultation. Based on services currently in-demand, only tele-ECG, tele-USG, and tele-Consultation are considered, and they become focal-points to be explored in this paper.

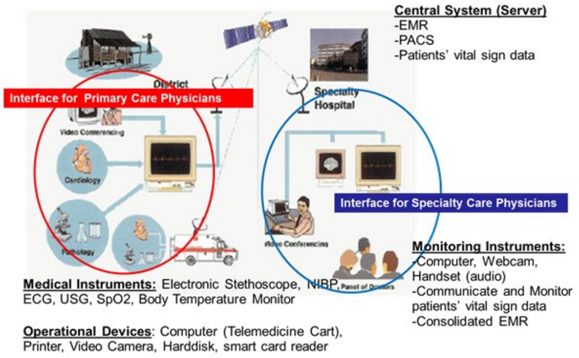


Fig. 1. Health services using telemedicine [1].

The successful implementation of telemedicine system depends on users' acceptance, where users may involve medical people, medical organizations, and patients. One aspect which drives acceptance is their experience in using this system, where it is known that users' experience depends on the level of quality offered by the system. However, providing a certain level of quality is challenging for telemedicine, due to non-uniformity regarding the parameter thresholds that contributes to the quality assessment [2]. Thus, the telemedicine services need to be classified before deriving each service's characteristics and its performance.

Telemedicine system relies on well-tested software, which ensures various aspects of usability, accessibility, security, and performance under various operational loads. Thus, appropriate software testing for verification and validation is needed to assess a telemedicine system. In this case, it involves observing the execution of the software on various subsets of all possible inputs and setting, and providing an evaluation of the output according to certain metrics [3]. Extensive software testing on real condition is time-consuming and expensive; thus, there needs to be an alternative environment where testing can be done in an efficient, flexible, and automated way. This is the motivation of providing virtual testbed, which in essence a controlled and software-based infrastructure which mimics the real condition.

The use of virtual testbed is common for assessing various system performances, e.g. wireless testbed [4], web security assessment [5], SFU (Selective Forwarding Unit)-based video conference [6], video surveillance [7], etc., but there is lack of reported use in telemedicine. In comparison, some field-trials of telemedicine deployment for measuring users' acceptance are reported, e.g. recent experiments presented in [8]. This paper presents an initiative to develop a virtual testbed, i.e. a flexible, scalable, and user-controlled testing environment for telemedicine.

The rest of this paper is structured as follows. Section II provides a quick overview of the telemedicine system developed by PTE-BPPT. Section III describes the virtual testbed set-up that mimics the real telemedicine system. Experimental results conducted on virtual testbed, for validating some aspects of security and performance, are presented in section IV. Finally, this paper concludes with some insights and future work.

II. TELEMEDICINE SYSTEM

PTE-BPPT has developed a telemedicine system with its tele-ECG, tele-USG, and tele-Consultation services that are currently being trialed in Tangerang regency, South Tangerang city, and BPPT's own clinic. An overview of this telemedicine system is depicted in Fig. 2, where some blocks related to discussion in this paper are shown.

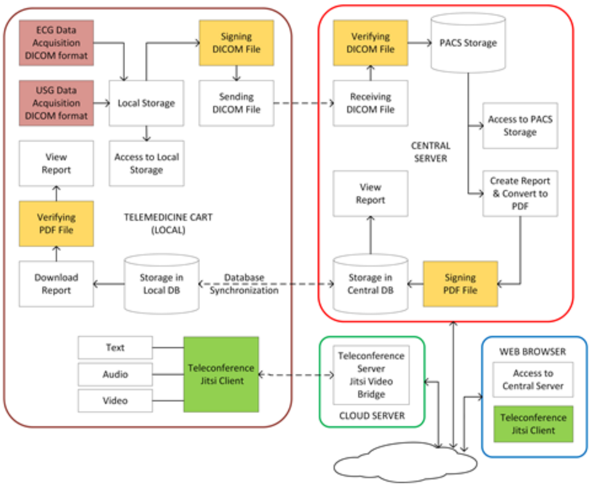


Fig. 2. Block diagram of a simplified PTE-BPPT's telemedicine system.

It can be inferred that there are three major parts in this system, i.e. 1) Telemedicine cart/workstation (local), 2) Central server, and 3) Remote users accessing the system via web browsers. Telemedicine cart/workstation is a device positioned at the patients' site, and it functions as an aggregator to various medical devices that measure the patients' bio-signals. Fig. 2 shows blocks for ECG and USG data acquisition and storage inside telemedicine cart. Both ECG and USG data are stored in DICOM (Digital Imaging and Communications in Medicine) format, as waveform and image, and they are signed digitally by the doctor who measures them. Note that when network bandwidth and local storage are limited SCP-ECG (Standard Communication Protocol for Computer-Assisted Electrocardiography) format is the preferable option for ECG data.

The ECG and USG data are sent reliably to central server, and their validity is verified before being stored in the PACS (Picture Archiving and Communication System) storage. This central server can be implemented in the hospital's data centre or in the cloud, and becomes the target point for remote access, e.g. by the specialist doctor who will analyse patients' status in a tele-Consultation session. It can be seen that patients' report, which includes diagnose and recommended action, is stored in pdf format, signed digitally by the specialist doctor, and then disseminated back to the patients' site (local) via database synchronization. Applying digital signatures to data in DICOM format and medical records in pdf format for telemedicine system is discussed in [9].

In addition to dealing with medical data to support tele-ECG and tele-USG services, the system also provides multimedia communication facility for tele-Consultation. It is based on SFU (Selective Forwarding Unit) teleconference using Jitsi media server, which provides end-to-end communication supporting text, audio, video streams routed via Jitsi video bridge. The choice of media selected during tele-Consultation depends on the available network bandwidth.

It can be seen that users simply use web browser to use this telemedicine system as message interactions involved are based on web technology. This ensures flexibility in software development and inter-operability.

From discussion above, it can be seen that there are two kinds of network traffic that need attention, i.e. medical data traffic and multimedia (video, audio, text) traffic. Each of them requires different objective which may result in different transport protocol used. Medical data traffic requires reliability; hence TCP (Transmission Control Protocol) is used. On the other hand, multimedia communication (particularly video) prefers minimum delay; hence UDP (User Datagram Protocol) is preferred. How these traffic flows mixed in the network and behave will affect system performance, and it requires specific assessment before deployment.

The description above mostly involves software and network protocol, so that this telemedicine system is suitable for a virtual testbed. Furthermore, detailed assessment on its function and performance can be done effectively in this environment.

III. VIRTUAL TESTBED

The design of telemedicine virtual testbed aims at creating a scalable testing environment, which can also be used for future development. The virtual testbed is essentially a software package which allows the deployment of some major parts of the system in a single computer (or a cluster) using some virtual machines and a virtual network. In this work, the virtual testbed has been developed in one Dell PowerEdge R920 server, equipped with 1.8 TB virtual disks data with 2x10 core CPU and 8x16 GB memory. This server supports up to 10 NIC cards. The virtualization layer was created by VMWare ESXi 6.7 and it is horizontally scalable, i.e. it can support many hosts with additional vCenter management platform.

As discussed in the previous section, there are three major roles in the developed telemedicine system: 1) telemedicine cart/workstation, 2) telemedicine central server which provides PACS services, and 3) teleconference server. Each role has a specification in virtual domain as shown in Table 1.

TABLE I. ROLES IN TELEMEDICINE SYSTEM AND THEIR SPECIFICATION.

Telemedicine Cart	
CPU	**4 x 2 core CPU**
Memory	**8 GB**
Hard disk	**466 GB**
NIC	**2 NIC (connect to private switch)**
Telemedicine Central Server	
CPU	**4 x 2 core CPU**
Memory	**8 GB**
Hard disk	**279 GB**
NIC	**1 NIC (connect to public switch)**
Teleconference Server	
CPU	**2 x 1 core CPU**
Memory	**2 GB**
Hard disk	**30 GB**
NIC	**1 NIC (connect to public network)**

The virtualized system was converted from the real system by using p2v (physical to virtual) technology from VMWare vCenter Converter Standalone version 6.2. This software supports the operating systems used in PTE-BPPT's telemedicine system, i.e. Windows 10 for telemedicine cart and Ubuntu 16.0 for telemedicine server. Both of them were virtualized by a hot-clone method, i.e. the target machine was in online mode (power on) [10]. The teleconference server, used in tele-Consultation, is hosted in BPPT's cloud, and can be accessed from the other virtual machines via BPPT's local area network.

Fig. 3 describes the virtual testbed configuration. One component inside the telemedicine virtual infrastructure needs further attention, i.e. Mikrotik virtual appliance. This virtual appliance acts as a router which connects the telemedicine cart's network to other networks managing telemedicine central server and teleconference server. This router will play the role in managing allocated bandwidth, ingress and egress telemedicine cart's network. In essence, this provides the model for various access problems at patients' site, which may have limited connectivity. During system testing, bottlenecking bandwidth can be applied to assess its impact on the performance.

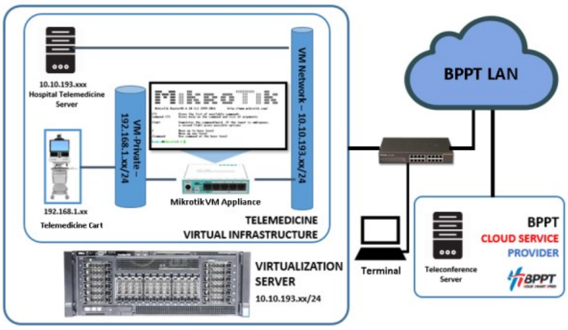


Fig. 3. Virtual testbed configuration.

There are various software functionalities involved in the telemedicine system, and they are also functional inside virtual testbed. However, discussion in this paper only relates to two functionalities: transmission and security. For transmission, scp (secure copy) tool (using TCP) is used to send ECG and USG data reliably from telemedicine cart to telemedicine central server, whereas video transmission relies on Jitsi's WebRTC messages (using UDP). For security, focusing only on validating data, the system utilizes dcmsign version 3.6.3 as a module to sign DICOM (EKG and USG) files with digital signatures [9].

IV. EXPERIMENTS

Two kinds of experiments were conducted with focus on security and performance aspects. Each experiment requires specific requirements and analysis, thus it is described separately as follows.

A. Security Validation

Medical data needs a system that can ensure three targets of security: confidentiality, integrity, and availability. This paper only discusses one of them, i.e. the use of digital signature to ensure data integrity. In this experiment, the integrity target concerning ECG and USG data in DICOM format was tested. The system must ensure those data that are transported via network are valid.

The system relies on some security validation modules; and in this case the focus in on signing and verification modules. These modules rely on DCMTK (DICOM toolkit), an open-source software package that implements large parts of DICOM standard. This package is integrated into the system, which involves various software integration and automated scripts. The signing module is installed on telemedicine cart and the verification module is installed on telemedicine server. All transaction processes related to signing and verification are recorded in the logging systems using text format.

The signing module calls the dcmsign function that creates a digital signature inside the DICOM file. Dcmsign adds (FFFA, FFFA) tagging from DICOM data dictionary comprising a digital signature sequence and its attributes needed for DICOM files' digital signature. The signing module is constructed by three classes: signing DICOM, creating transaction log, and sending DICOM file to the telemedicine server. The DICOM file will be signed by the doctor's digital certificate at patients' site (e.g. at local health center) and every signing process will be recorded in a transaction log. The transaction log records signing results, either "success" or "failed", and it is based on the dcmsign's OFLOG documentation. Fig. 4 shows a snapshot of signing DICOM class diagram.

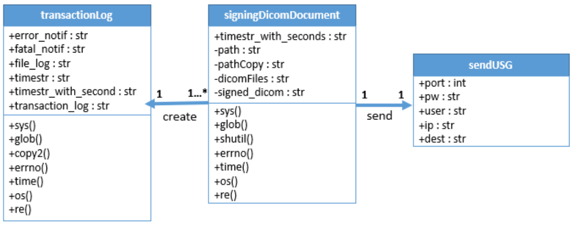


Fig. 4. Signing DICOM class diagram.

In the telemedicine server, DICOM files are validated before being uploaded to the PACS server. The verification module uses the same tool as the one in the signing module, but it was compiled in Linux environment, considering the telemedicine server uses Ubuntu 16.04. The module calls dcmsign function to verify the digital signature and records the validation results. Fig. 5 shows a snapshot of verifying DICOM class diagram.

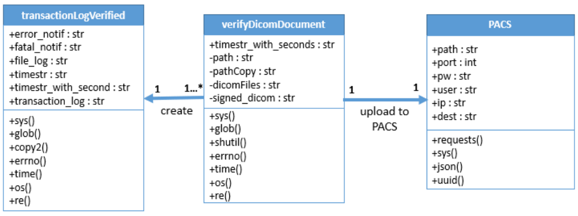
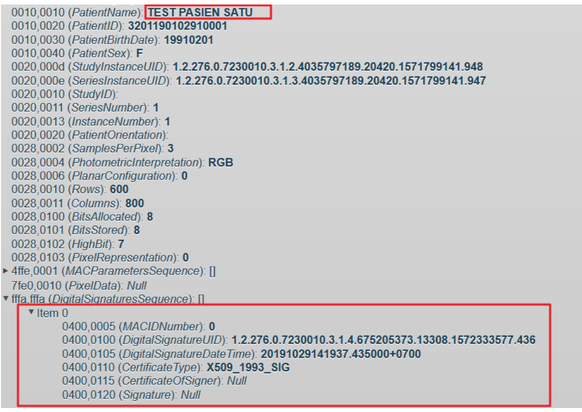


Fig. 5. Verifying DICOM class diagram.

To simulate a situation where data integrity is violated, the testing scenario involves modifying DICOM data and verifying them. In this experiment, the modification was carried out by changing DICOM tagging using dcmodify; e.g. modifying the patient's name by changing DICOM tagging (0010,0010). The modified DICOM was re-verified by the verification module in the telemedicine server. All activities were recorded in the transaction log to see whether the verification processes succeed or not. The files with "success" status were uploaded to the PACS server and the files with "failed" status were transferred to the archived folder in the server. This mechanism facilitates future investigation. Fig. 6 shows an example of USG file (created in October 23th 2019) that was successfully uploaded to the PACS server.



```
  0010,0010 (PatientName): TEST PASIEN SATU
  0010,0020 (PatientID): 3201190102910001
  0010,0030 (PatientBirthDate): 19910201
  0010,0040 (PatientSex): F
  0020,000d (StudyInstanceUID): 1.2.276.0.7230010.3.1.2.4035797189.20420.1571799141.948
  0020,000e (SeriesInstanceUID): 1.2.276.0.7230010.3.1.3.4035797189.20420.1571799141.947
  0020,0010 (StudyID):
  0020,0011 (SeriesNumber): 1
  0020,0013 (InstanceNumber): 1
  0020,0020 (PatientOrientation):
  0028,0002 (SamplesPerPixel): 3
  0028,0004 (PhotometricInterpretation): RGB
  0028,0006 (PlanarConfiguration): 0
  0028,0010 (Rows): 600
  0028,0011 (Columns): 800
  0028,0100 (BitsAllocated): 8
  0028,0101 (BitsStored): 8
  0028,0102 (HighBit): 7
  0028,0103 (PixelRepresentation): 0
▸ 4ffe,0001 (MACParametersSequence): []
  7fe0,0010 (PixelData): Null
▾ fffa,fffa (DigitalSignaturesSequence): []
    ▾ Item 0
          0400,0005 (MACIDNumber): 0
          0400,0100 (DigitalSignatureUID): 1.2.276.0.7230010.3.1.4.675205373.13308.1572333577.436
          0400,0105 (DigitalSignatureDateTime): 20191029141937.435000+0700
          0400,0110 (CertificateType): X509_1993_SIG
          0400,0115 (CertificateOfSigner): Null
          0400,0120 (Signature): Null
```

Fig. 6. USG file in PACS server.

Fig. 6 shows the file that was signed in October 29th 2019 with Digital Signature UID: 1.2.276.0.7230 010 .3.1.4.675205373.13308.1572333577.436. This file was then modified and re-verified using the verification module. To cross-check whether the file was already changed or not, viewing DICOM file was needed. In this experiment, MicroDicom viewer was used, and Fig. 7 shows an example of the modified USG DICOM file, where the patient's name was changed to "Changed Name to John Doe". The "failed" status can be seen from the transaction log shown in Fig. 8.

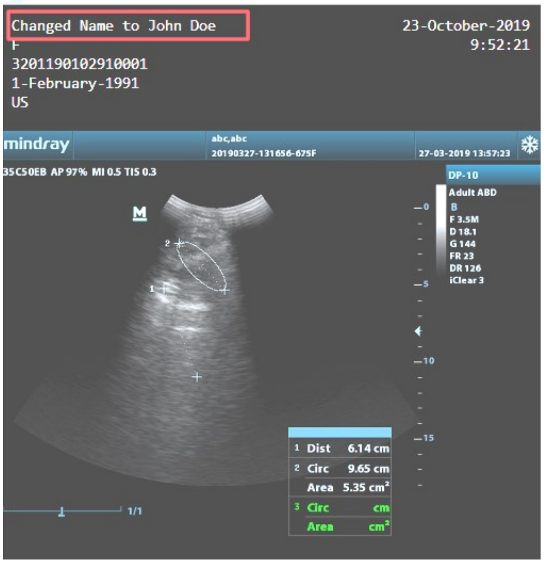
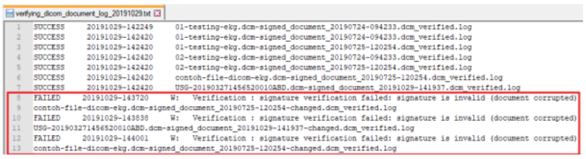


Fig. 7. Patient's attributes after modification.



```
SUCCESS    20191029-142249    01-testing-ekg.dcm-signed_document_20190724-094233.dcm_verified.log
SUCCESS    20191029-142420    01-testing-ekg.dcm-signed_document_20190724-094233.dcm_verified.log
SUCCESS    20191029-142420    01-testing-ekg.dcm-signed_document_20190725-120254.dcm_verified.log
SUCCESS    20191029-142420    02-testing-ekg.dcm-signed_document_20190724-094233.dcm_verified.log
SUCCESS    20191029-142420    02-testing-ekg.dcm-signed_document_20190725-120254.dcm_verified.log
SUCCESS    20191029-142420    contoh-file-dicom-ekg.dcm-signed_document_20190725-120254.dcm_verified.log
SUCCESS    20191029-142420    USG-20190327145652001OABD.dcm-signed_document_20191029-141937.dcm_verified.log
FAILED    20191029-143720    W:   Verification : signature verification failed: signature is invalid (document corrupted)
contoh-file-dicom-ekg.dcm-signed_document_20190725-120254-changed.dcm_verified.log
FAILED    20191029-143838    W:   Verification : signature verification failed: signature is invalid (document corrupted)
USG-20190327145652001OABD.dcm-signed_document_20191029-141937-changed.dcm_verified.log
FAILED    20191029-144001    W:   Verification : signature verification failed: signature is invalid (document corrupted)
contoh-file-dicom-ekg.dcm-signed_document_20190725-120254-changed.dcm_verified.log
```

Fig. 8. Transaction log.

B. *Performance Tests*

Lack of network qualities may impact telemedicine services severely. It is usually assumed that network infrastructures in remote areas contribute to the problems. This is represented by the network connecting the three major components, i.e. telemedicine cart, telemedicine central server, and teleconference server. Thus, the virtual testbed is set-up so that there is a user-controlled network connecting these three components. From Fig. 3, the user-controlled network is modeled by Mikrotik VM appliance whose bandwidth can be varied with a simple queue technique. This is referred to as bottleneck bandwidth.

In this experiment, the bottleneck bandwidth was varied from 64, 128, 256, 512, till 1024 Kbps. Each instance of bottleneck bandwidth was the basis for performance measures. The focus was on measuring transmission latency and achievable throughput (used bandwidth) for sending medical data. Source data used were the ECG and USG DICOM files whose sizes are around 140 Kb and 1.4 Mb respectively. Fig. 9 shows visualization of these ECG and USG data used in this experiment.

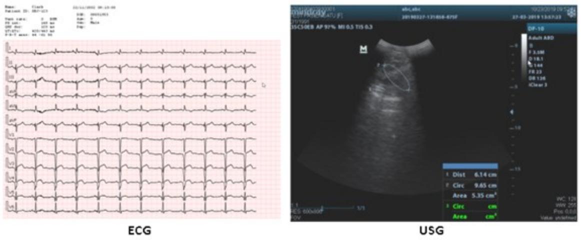


Fig. 9. ECG and USG data used.

Medical data transmission was conducted periodically for the duration of each measurement. This represents a worst-case scenario, as typically only one transmission is needed for each session. Other transmissions may include video streams, by playing a video file containing two people's conversation. This represents a video session in tele-Consultation. For each instance of measurement, the same video stream was required, and this was achieved by playing the same video file utilizing fake-media feature of WebRTC and Jitsi media server [6]. The fake-media is in y4m format, and mainly used as a substitute to webcam. This teleconference is considered a modern SFU (Selective Forwarding Unit) teleconference technique, with features such as scalable video coding, efficient video stream forwarding based on audio activities, and effective used of computing resources [11].

The performance tests were conducted in 4 different scenarios: "ECG", "ECG and Video", "USG", and "USG and Video". Each of them represents tele-ECG only, tele-ECG with tele-Consultation, tele-USG only, and tele-USG with tele-Consultation. How bottleneck bandwidth and mixed traffic affects the performance of medical data transmission is the current research interest. Thus, the measurement objectives were their transmission latencies and achievable throughputs, and the results are shown in Fig. 10 and Fig. 11.

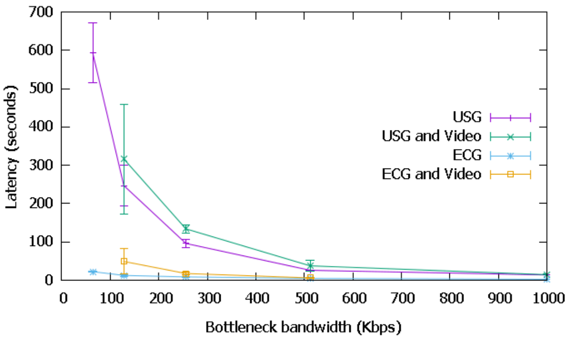


Fig. 10. Transmission latency.

From Fig. 10, it can be seen that for bottleneck bandwidths above 512 Kbps all latencies are acceptable as they are mostly below 30 seconds (0.5 minutes). However, they are worse-off sharply for lower bandwidths. More latencies and higher variabilities are observed for scenarios which include video transmission. This shows "UDP dominance" and

"TCP starvation" phenomenon, where UDP traffic (video) competes with TCP traffic (medical data) in using network bandwidth. In this case, UDP traffic dominates, and TCP traffic needs to back-off. TCP traffic is reliable as long as the communication path is still connected, and if there are losses it requires retransmission to ensure the data reach destination correctly. This increases latency, and worse-off for lower bottleneck bandwidths.

From Fig. 10, it can be seen that tele-ECG is still acceptable for lower end of bottleneck bandwidths, with latency around 22 seconds at 64 Kbps bandwidth. However, since video transmission alone normally requires 256 Kbps bandwidth, then tele-ECG with tele-Consultation is recommended for bandwidth above 256 Kbps. For bandwidth below 256 Kbps, tele-Consultation should fallback to audio or text-based communication.

It can be seen that tele-USG might still be acceptable for 256 Kbps bandwidth, with latency around 96 seconds (1.6 minutes). At 64 Kbps and 128 Kbps bandwidth, the latencies are too high, i.e. around 600 seconds (10 minutes) and 240 seconds (4 minutes). They may be considered for offline mode version of tele-USG, i.e. the doctors are not waiting for USG data to arrive while communicating using audio or text. Tele-USG with tele-Consultation is recommended for bandwidth above 512 Kbps.

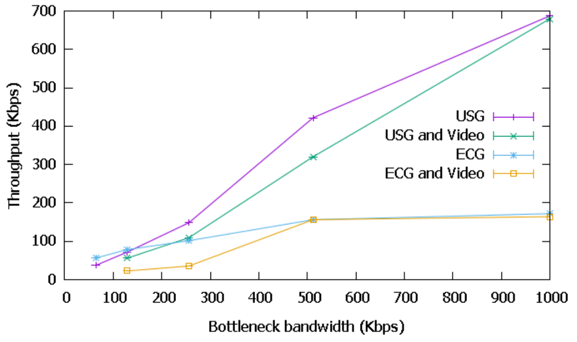


Fig. 11. Achievable throughput (used bandwidh).

Traffic behavior from the four scenarios can be further inferred from Fig. 11. It can be seen that higher latencies correspond to lower achievable throughputs, and in mixed traffic scenarios ECG and USG traffic lost some of their throughputs due to dominating video traffic.

V. CONCLUSION

This paper has presented a virtual testbed which can be used as a model for telemedicine services. Some experiments were conducted to verify the validity of medical records' transmission and to assess the expected performance related to tele-ECG, tele-USG, and tele-Consultation. It is shown that digital signing-related modules work well, and experiment results provide some insights on the required network bandwidth.

This virtual testbed provides a scalable and controlled environment for extensive testing, and to verify the telemedicine system's effectiveness and robustness prior to deployment.

ACKNOWLEDGMENT

The authors thank BPPT (Badan Pengkajian dan Penerapan Teknologi) for financial support to this research project.

REFERENCES

[1] BPPT, Program Manual 2015-2019 Tahun Anggaran 2018: *Inovasi Konvergensi Teknologi Elektronika dan Telekomunikasi Sistem Elektromedika (Inovasi Teknologi Telemedicine)*, Jakarta, 2018.

[2] M. Ullah, M. Fiedler, K. Wac, "On The Ambiguity of Quality of Service and Quality of Experience Requirements for eHealth Services", in Proc. *International Symposium on Medical Information and Communication Technology (ISMICT)*, 2012.

[3] A. Bertolino, "Software Testing Research: Achievements, Challenges, Dreams", in *Future of Software Engineering*, 2007.

[4] L. Navarro, R. B. Vinas, C. Barz, J. Bonicioli, B. Braem, F. Freitag, I. Vilata-i-Balaguer, "Advances in Wireless Community Networks with the Community-Lab Testbed", *IEEE Communications Magazine*, 57(7), 2016, pp. 20-27.

[5] D. D. Kankhare, A. A. Manjrekar, "A Cloud Based System to Sense Security Vulnerabilities of Web Application in Open-Source Private Cloud IAAS", in *Proc. International Conference on Electrical, Electronics, Communication, Computer and Optimization Techniques (ICEECCOT)*, 2016, pp. 252-255.

[6] E. Andre, N. Le Breton, A. Lemesle, L. Roux, A. Gouaillard, "Comparative Study of WebRTC Open Source SFUs for Video Conferencing", in *Proc. Principles, Systems and Applications of IP Telecommunications (IPTComm)*, 2018.

[7] J. Bustos-Jimenez, R. Alonso, C. Faundez, Hugo Meric, "BoxingExperience: Measuring QoS and QoE of Multimedia Streaming using NS3, LXC and VLC", in *Proc. 8th IEEE Workshop on Network Measurements (WNM),* 2014, pp. 658-662.

[8] T. O.C. Edoha, A. D. Korab, P. Pawarc, G. C. Coulibaly, B. R. U. Alahassa, "Predicting telemedicine system user satisfaction in sub-Saharan Africa Predicting telemedicine system user satisfaction in sub-Saharan Africa" , in *ICT Express*, 2(4), 2016, pp. 163-167.

[9] T. Agastani, M. Iqbal, A. A. N. A. Kusuma, C. W. Purnaadi, "Kerekayasaan Modul Penjamin Keabsahan Data Medis pada Inovasi Sistem Telemedicine BPPT" *Jurnal Teknologi Infomasi, Komunikasi dan Elektronika (JTIKE),* 1(1), 2018, pp. 1-6.

[10] I. Gordin, "Virtualizing Real Servers with Unsupported OS by VMWare vCenter Converter v6", in *Proc. RoEduNet International Conference - Networking in Education and Research (RoEduNet NER)*, 2015.

[11] B. Grozev, L. Marinov, V. Singh, E. Ivov, "Last N: Relevance-Based Selectivity for Forwarding Video in Multimedia Conferences", in *Proc. 25th ACM Workshop on Network and Operating Systems Support for Digital Audio and Video (NOSSDAV), 2015.*