Advances in Multimedia

Multimedia Immersive Technologies and Networking

Guest Editors: Mohammed Ghanbari, Feng Wu, Cha Zhang, Ghassan Alregib, and Athanasios Vasilakos

Multimedia Immersive Technologies and Networking

Advances in Multimedia

Multimedia Immersive Technologies and Networking

Guest Editors: Mohammed Ghanbari, Feng Wu, Cha Zhang, Ghassan Alregib, and Athanasios Vasilakos

Contents

Hindawi Publishing Corporation
Advances in Multimedia
Volume 2008, Article ID 983735, 2 pages
doi:10.1155/2008/983735

Editorial
Multimedia Immersive Technologies and Networking

Mohammed Ghanbari,[1] Feng Wu,[2] Cha Zhang,[3] Ghassan Alregib,[4] and Athanasios Vasilakos[5]

[1] *School of Computer Science and Electronic Engineering, University of Essex, Colchester CO4 3SQ, UK*
[2] *Microsoft Research Asia, Beijing 100080, China*
[3] *Microsoft Research, One Microsoft Way, Redmond, WA 98052, USA*
[4] *School of Electrical and Computer Engineering, Georgia Institute of Technology, 210 Technology Circle, GA 31407, USA*
[5] *University of Western Macedonia, P.O. Box 65251, 15410 Attica, Greece*

Correspondence should be addressed to Cha Zhang, chazhang@microsoft.com

Received 16 December 2008; Accepted 16 December 2008

Immersive telecommunication technologies are typically used for capturing, processing, analyzing, transmitting, and enabling the remote fruition of objects, environments, and bioentities. Applications of immersive telecommunication technologies may span over a very wide range from industrial automation, health care, education to entertainment.

Over the past two decades, the joint work of networking and the multimedia has led to a wide range of tools and supports, enabling the commercial world-wide deployment of multimedia-based services and products. All the related research and standardization activities enabled multimedia data to be adapted to different networking technologies, wired and wireless, established and emerging, with different and time-varying channel conditions. Also, the restrictions due to the terminal processing power of handheld devices are on the way to be successfully overcome.

On the other hand, computer graphics, computer vision, and virtual/augmented reality communities have often developed conceptual models and tools working separately, mainly for fulfilling local and specific needs of predefined contexts. For instance, computer vision has often aimed at performing specific tasks (e.g., tracking, object recognition) in some specific scenarios (e.g., providing localization and visualization for robotic application or video surveillance). Computer graphics has developed a set of tools, such as rendering and texturing, which have been mainly applied to animation and games and, more in general, in the entertainment industry, mainly aiming to a local use, though forms of remote collaborative environment (such as 3D gaming) are starting to take off the ground. Similar approaches have been followed so far by virtual/augmented reality research community.

In this special issue, we present several papers to bridge the traditional gap existing between immersive technologies and networking, focusing on how traditional and emerging fields (e.g., pervasive computing) can be brought together under the networking umbrella.

The first paper "Enabling cognitive load-aware AR with rateless coding on wearable network," by Razavi et al., proposes a block-based form of rateless channel coding for wearable network, which minimizes energy consumption by reducing the overhead from FEC. Compared with the packet-based rateless coding, data loss is reduced and energy consumption is improved with this form of block-based coding.

The second paper "Providing QoS for networked peers in distributed haptic virtual environments" deals with haptic information, where the quality of service (QoS) required to support haptic traffic is significantly different from that used to support conventional real-time traffic such as voice or video. In this paper, Marshall et al. present a peer-to-peer distributed haptic virtual environment (DHVE) architecture of positions. The paper aims to enable force interactions between two users whereby force data is sent to the remote peer in addition to positional information. The work presented involves both simulation and practical experimentation where multimodal data is transmitted over a QoS-enabled IP network.

In the third paper "A reliable and efficient remote instrumentation collaboration environment," Calyam et al. address an important problem in remote access of scientific instruments over best effort networks. They provide an analytical model that characterizes the user's quality of experience (QoE) given the limitations imposed by the

network. The model is tested via objective and subjective measurements using a remote microscopy testbed. The authors package the model into a Remote Instrumentation Colaboration Environment (RICE) software with detailed explanation of potential functionalities that include VoIP and health monitoring.

The fourth paper "Sensor network-based localization for continuous tracking applications: implementation and performance evaluation," by Denegri et al., presents a localization platform that exploits a single-hop wireless sensor network (WSN), based on a Microchip MCU and a Cypress RF device, for tracking of its moving nodes. The authors divided the nodes into three sets: the anchor nodes that generate ultrasonic pulses, the moving nodes which estimate the pulse trip-time, and finally the nodes that collect data from the surrounding field. The computed positions of the moving nodes and transferred information are sent to external users on the Internet.

In the fifth paper "Remote laboratory experiments in a virtual immersive learning environment," Berruti et al. introduce the Virtual Immersive Learning (VIL) test bench that focuses on remote lecturing as an application. The importance of this work is the ability of the proposed system to function as the base for various innovations and algorithms that can be easily implemented and tested on the proposed and developed framework. Besides its flexibility, the system is portable and has a low price tag. The authors in this paper address the major features of the framework supported with performance measurements.

Mohammed Ghanbari
Feng Wu
Cha Zhang
Ghassan Alregib
Athanasios Vasilakos

Hindawi Publishing Corporation
Advances in Multimedia
Volume 2008, Article ID 853816, 12 pages
doi:10.1155/2008/853816

Research Article

Enabling Cognitive Load-Aware AR with Rateless Coding on a Wearable Network

R. Razavi, M. Fleury, and M. Ghanbari

Department of Computing and Electronic Systems (CES), University of Essex, Wivenhoe Park, Colchester CO4 3SQ, UK

Correspondence should be addressed to M. Fleury, fleum@essex.ac.uk

Received 29 February 2008; Revised 27 June 2008; Accepted 17 September 2008

Recommended by Feng Wu

Augmented reality (AR) on a head-mounted display is conveniently supported by a wearable wireless network. If, in addition, the AR display is moderated to take account of the cognitive load of the wearer, then additional biosensors form part of the network. In this paper, the impact of these additional traffic sources is assessed. Rateless coding is proposed to not only protect the fragile encoded video stream from wireless noise and interference but also to reduce coding overhead. The paper proposes a block-based form of rateless channel coding in which the unit of coding is a block within a packet. The contribution of this paper is that it minimizes energy consumption by reducing the overhead from forward error correction (FEC), while error correction properties are conserved. Compared to simple packet-based rateless coding, with this form of block-based coding, data loss is reduced and energy efficiency is improved. Cross-layer organization of piggy-backed response blocks must take place in response to feedback, as detailed in the paper. Compared also to variants of its default FEC scheme, results from a Bluetooth (IEEE 802.15.1) wireless network show a consistent improvement in energy consumption, packet arrival latency, and video quality at the AR display.

1. INTRODUCTION

Augmented reality (AR) allows a video display of the outside world to be supplemented with computer-generated graphics, annotations, instrument readings, and other sources of information [1]. Emergency workers typically view a head-mounted display (HMD), which may be partially mediated, that is, the subject is also able to view the outside world directly. (A Carl Zeiss look-around HMD is suitable for this task. It supports true color with maximum resolution of 800 × 600 pixels, though currently fed through a US B cable, refer to http://www.wearitatwork.com/Carl-Zeiss-Look-around-HMD.173.0.html.) Unfortunately, the level of information delivered to the operative may cause saturation (cognitive overload) due to the limited capacity of human memories [2, 3]. Cognitive overload may cause an operative's performance to deteriorate. For example, firefighters as they pass through the rooms may have a steerable view of a scene in a burning building with the addition of a room plan within the building and possibly a feed from wireless sensor nodes in the building when the view is impaired [4]. Another example of how cognitive load can impact upon an AR includes the displays presented to some aircraft crew and military personnel [5]. To remedy this problem, in augmented cognition [6], biosensors upon the person feedback information to the AR unit and these act to reduce the level of viewable information. In augmented group cognition, that information may come from other personnel such as other firefighters within a building but again the cognitive load arising from the information transfer is controlled by inputs from biosensors. However, if feedback is over a wireless network then the extra traffic arising from bisosensor data in addition to the AR video stream will present difficulties.

Wearable wireless networks that include biosenors are generally composed of low-power devices, and consequently energy efficiency is an important issue. The devices are also close to each other. This proximity of wireless devices implies two things: firstly, that the signal propagation time is negligible with the result that immediate feedback from the receiver is possible, and secondly, the chance of data losses owing to wireless channel interferences is higher. More specifically, there can be local scattering due to changes in the geometry of the wearer's body. Unfortunately, the video

stream that supports AR is highly susceptible to errors as it must be compressed to avoid the unacceptably large datarates of raw video. Because successive video frames are broadly similar (except at scene cuts) to gain encoding efficiency only the difference between successive frames is coded. Removing temporal redundancy introduces a dependency on previously transmitted data, which implies that lost packets will have an impact on future frames. Consequently, loss of packets from an intra- or spatially-coded I-frame is particularly harmful, though loss of data from other predictively-coded frames (ones employing motion compensation to reduce temporal redundancy) also has a knock-on effect at the decoder. Video frames are generally organized as a group of pictures (GOPs) of typically 12 or 15 pictures or frames initiated by an I-frame, but until the arrival of the next I-frame a decoder will face problems in reconstructing the sequence if reference frame packets are lost. To reduce latency redundant data in the form of forward error control (FEC) is introduced rather than rely on error control through some form of automatic repeat request (ARQ) for retransmissions of complete packets. However, introducing FEC increases transmission overhead and in general further stretches the capacity of a wearable wireless network, already under strain because of the additional biosensor traffic.

Our proposed solution is a power efficient algorithm based on a novel block-based rateless FEC that tries to avoid unnecessary packet retransmissions. When a packet arrives at a receiver, it is declared lost because it cannot be decoded, though individual blocks within it may become decodable if redundant blocks were to be available. If a packet or rather the blocks within it is found to be not decodable, the method resends redundant blocks in a current packet rather than resend the complete packet. Immediate feedback from the receiver, which in some wireless technologies such as Bluetooth comes for free because of its time division duplex (TDD) operation, allows the implementation of the block-based rateless scheme. The focus of this paper is the AR application with biosensors in which Bluetooth is modeled as a suitable wireless technology. However, the block-based rateless coding described in the paper is applicable to centrally scheduled, packet-switched wireless networks with low latency feedback. IEEE 802.11e operating with hybrid coordinator function (HCF) controlled channel access (HCCA) can operate in this fashion and could be considered for a wearable wireless network, as this variant of IEEE 802.11 was designed with multimedia traffic in mind. Though obviously unsuitable for this AR application, the IEEE 802.16 is also centrally scheduled and uses TDD.

It is the datarate levels that present the most critical impact upon the coexistent video stream needed to support AR rather than the traffic patterns generated by the biosensors. This is because the need to support video of a suitable quality already stresses a wireless network. We have tested video transmission against these critical data rates and show that, with our form of block-based rateless coding, overhead is reduced, while error protection is preserved. Therefore, the main contribution of the paper is the block-based rateless coding which reduces overhead and consequently makes running the AR application with additional biosensor traffic easier to accomplish. As part of our introduction, we now further describe the rateless coding solution and the main features of the wearable wireless network. We also consider the suitability of existing wireless technologies for this application.

1.1. Block-based rateless coding

Because wireless networks are subject to various forms of interference such as fast and slow fading, shadowing, and radio frequency noise, the paper proposes that Fountain or rateless channel error coding [7] of the video stream will reduce erasures, while jointly improving energy consumption taken up in transmission of the video stream. Error bursts arising from slow fading are difficult to correct with standard block-based channel coding but can be addressed through erasure coding. This paper demonstrates an innovatory form of block-based rateless coding that is better able to reduce transmission energy consumption through reduced overhead than other FEC-based methods, including packet-based rateless coding. As nonrateless block-based FEC is a default option in the wireless network under test, by comparison with the default scheme the reader will be able to judge the relative advantage of opting for block-based rateless coding.

With the proposed block-based method of rateless coding, redundancy is reduced because the unit of coding is not a packet but a block within a packet. By piggybacking redundant blocks onto newly transmitted packets, redundancy is incrementally achieved until either prior video-bearing packets with erased blocks are reconstructed or the display deadline of the frame of which that packet is a part expires. Furthermore, compared to the other FEC methods tested, fewer packets are dropped through late arrival, which is important for video in general and for a real-time application in particular. Lower average delay leads to smaller playout buffers, with consequent saving in memory energy consumption. Compared to simple packet-based rateless coding, data loss is reduced and energy efficiency is improved in worsening channel conditions. However, block-based rateless coding requires cross-layer attention to be paid to the feedback channel and block packing for multiple packets must be recursively applied in the reply packet.

In the multimedia broadcast multicast system (MBMS) [8, 9], rateless coding at the application layer has been introduced by 3GPP for video streaming. However, MBMS differs from the use of rateless coding in this paper because (a) it is for multicast not for unicast, (b) it sends separate FEC packets and, hence, is properly described as packet-based, and (c) there is no feedback, because rateless coding is employed for its maximum distance separable property, that is, the source packets can be reconstructed with high probability from any set of k or just slightly more than k received symbols. In [10], rateless coding is applied to unicast streaming over the fixed Internet but at the packet level and with large values of k.

1.2. Wearable wireless network

We model a wearable wireless network to find the impact of data traffic from the biosensors upon a video stream passing from a camera worn on the operative to the HMD after augmentation. A central wearable computer processes incoming encoded video from the camera unit, adding additional information from an internal source and/or transferred from an external wireless source. Hardware chroma-keying for this purpose allows video-rate display to be approached. The video is retransmitted in encoded form to the display device.

For a wearable AR system, energy consumption is important because batteries are carried upon the person and cannot easily be replaced in stressful scenarios. In [11], it was reported that there is approximately a linear relationship between bitrate and energy consumption and in [12] it was shown that transmission accounts for more than a third of the total energy consumption in communication on a mobile device. Therefore, reducing data transmission is an effective way to reduce energy usage. In packet-based rateless coding, each packet stream certainly contains $k(1+\varepsilon)$ packets, where ε is a small fractional overhead, typically amounting to two extra packets for values of k over 1000 [10], to ensure with high probability that all k information packets are decodable if received without error (rateless codes are constructed in probabilistic fashion). Raptor codes [13] have constant time encode and linear decode computational complexity, though additional precoding is performed prior to formation of the rateless code. Though it is outside the scope of the paper, use of rateless codes also will help reducing energy consumption at the transmitter, the transceiver at the AR processor and the receiver for the HMD, as their coding complexity is linear, $O(n)$ compared to Reed-Solomon (RS) erasure codes for which coding complexity can be quadratic, $O(n^2)$, though decoding in the frequency domain reduces to $O(n \log n)$.

A wide variety of biosensors have been proposed such as the emotion mouse [14], eyetrackers [3], oxygen saturation meters [15], and heart-rate monitor or electrocardiography (ECG) [16]. These measure cognitive load indirectly by the physical and emotional states of an operative. Direct information may come from monitoring the brain's activity-electroencephalography (EEG) [17]. If a person is instrumented by sensors of these kinds, it is important [18] to reduce the wiring in order to improve the comfort of wearing what is in effect a wearable computer system, with display at the HMD, input from the various sensors and interpretation at a central processor unit. In the case of augmented group cognition then input/output will take place with nearby personnel. These considerations suggest a wireless sensor network [18] which is worn upon the body. In this paper, such a network is modeled for the case of a fireworker with an HMD for which the video input is moderated at the point of augmentation, according to input from ECG (physical input) and EEG (direct input).

1.3. Wireless network technology

In this system, both the encoded video stream from the camera and the EEG will potentially load the wireless sensor network, which is why in [18] a Bluetooth (IEEE 802.15.1) wireless network [19] was considered rather than the lower bandwidth Zigbee (IEEE 802.15.4) network [20]. Both are lower powered relative to WiFi (IEEE 802.11a,g,n) [21], with Zigbee having a lower duty cycle than Bluetooth. (Bluetooth has typical current of 0.3–350 mA, compared to IEEE 802.11's 480–700 mA.) Bluetooth and Zigbee support a star topology network with a central privileged node. However, in a Bluetooth network or piconet with up to eight active nodes (an extension mechanism allows the construction of scatternets, with more nodes when necessary), access to the wireless channel is by time division multiple access (TDMA), whereas Zigbee access is distributed, which can lead to unpredictable delays, which will have an adverse effect on delay-intolerant video. The small user payload of only 101 bytes could also create problems if error resilence methods were applied to video data. A Zigbee network's maximum (shared) bitrate is 250 kbps, whereas for Bluetooth v. 2.1 with enhanced data rate (EDR) [22], according to modulation type and channel conditions, the gross (shared) air rate is 3.0 Mbps which equates to 2.2 Mbps mean gross user payload. In [18], the bandwidth of a 128-channel EEG is assessed at 2 Mbps (1 kHz at 16-bit sampling [23]) and an ECG at 6 kbps (512 Hz at 12-bit sampling [24]). We suppose in this paper that either compression of the EEG signal has occurred or a reduced number of sensor patches are applied. In [25], compression was already applied to ECG signals, and we assume the bitrate of this source also is reduced as a result compared to the figure quoted in [18]. Notice that wireless EEG already exists in [26] with a 32-channel system for epilepsy monitoring rather than a wearable system.

Wireless networks are now preferred rather than the pioneering but cumbersome wired networks on older wearable computers [27]. Though a near-field intrabody network [28] may be ideal for low-power sensor/actuator devices, they do not have sufficient bandwidth for augmented cognition using EEG and indeed for video transmission. Bluetooth is similar in topology, frequency-hopping spread spectrum (FHSS), and modified TDMA to BBN's BodyLAN [29], which was originally designed as a body network. Fabric area networks (FANs) [30], through near-field communication at connection points between wired items of clothing, reduce the problem of interference, a detraction of some systems. However, commercial difficulties have impeded the implementation of FANs. Therefore, Bluetooth is the most feasible wireless system currently available for this application, and in this paper we model the AR encoded video stream as a Bluetooth piconet. Bluetooth is very suitable for application of rateless coding as there is automatic feedback to the sender and because of its short range (typically for class 2 devices less than 10 m) that feedback is of low latency. In Bluetooth, fast ARQ is available by virtue of TDD polling, which is necessary for transmit/receive recovery, allowing a single-chip implementation. Bluetooth packets also automatically contain a cyclic redundancy check (CRC) for the user payload, allowing detection of a failed decode. It is assumed in this paper that the CRC is applied to the payload after rateless decoding.

The remainder of this paper is organized as follows. Section 2 presents essential background to the understanding of the simulated results. Section 3 introduces the proposed block-coding version of rateless channel coding, detailing the algorithm employed. It also sets out the scenario modelled taking into account the video stream and other traffic from biosensors and an external feed with additional augmented information and/or biosensor information from other operatives in the event of augmented group cognition. Section 4 presents our results in terms of packet loss due to buffer overflow or inability to repair packets in error before their display deadlines expire. Relative energy consumption and delivered video quality are presented. The section also considers packet arrival latency. Finally, Section 5 draws some conclusions.

2. BACKGROUND

In this paper, rateless codes are employed to protect the AR video stream with minimal overhead. This section outlines the essential features of rateless channel codes, previous work in this field, and then goes on to describe the Bluetooth wireless network.

2.1. Rateless codes

Rateless coding is ideally suited [7] to a binary erasure channel in which either the error-correcting code works or the decoder fails and reports that it has failed. In erasure coding, all is not lost as flawed packets may be reconstructed from a set of successfully received packets (if sufficient of these packets are received). Unlike fixed-rate erasure coding, rateless coding relies on feedback. An (n, k) RS erasure code over an alphabet $q = 2^L$ (where L is the number of bits in a packet) has the property that if *any* k out of the n packets transmitted are received successfully then the original k packets can be decoded. However, in practice not only must n, k, and q be small but also the computational complexity of the decoder is of order $n(n-k)\log_2 n$. Erasure coding in video communication has normally been applied to packets (refer to Section 2.2), but can be applied to any symbols, such as blocks within a packet. In the literature, the term block is sometimes used synonymously for packet but in this paper we reserve the term to blocks within a packet.

The class of Fountain codes [7] allows a continual stream of additional packets to be generated in the event that the original packets could not be decoded. It is the ability to easily generate new packets that makes Fountain codes rateless. Decoding will succeed with small probability of failure if any of $k(1+\varepsilon)$ packets are received. In its simplest form, the packets are combined in an exclusive OR (XOR) operation according to the order specified by a random low-density generator matrix and in this case, the probability of decoder failure is $\partial = 2^{-k\varepsilon}$, which for large k approaches the Shannon limit. The random sequence must be known to the receiver but this is easily achieved through knowledge of the sequence seed. Luby transform (LT) codes [31] reduce the complexity of decoding a simple Fountain code (which is of order k^3) by means of an iterative decoding procedure, provided that the column entries of the generator matrix are selected from a robust Soliton distribution. In the LT generator matrix case, the expected number of degree one combinations (no XORing of packets) is $S = c\log_e(k/\partial)\sqrt{k}$, for small constant c. Setting $\varepsilon = 2\log_e(S/\partial)S$ ensures that by sending $k(1+\varepsilon)$ packets these are decoded with probability $(1-\partial)$ and decoding complexity of order $k\log_e k$.

Furthermore, if the packets are pre-encoded with an erasure code a weakened LT transform can be applied to the packets and their parity packets. The advantage of this Raptor code [13] is a decoding complexity that is linear in k. Notice that an essential difference between Fountain erasure codes and RS erasure codes is that Fountain codes in general are not systematic and that even if there were no channel errors there is a very small probability, assuming correct design, that the decoding will fail. In compensation, they are completely flexible, have linear decode computational complexity, and generally their overhead is considerably reduced compared to fixed erasure codes. In the 3GPP standard, a systematic Raptor code is arrived at [13] by first applying the inverse of the inner LT to the first k symbols before the outer precoding step.

2.2. Related work

Rateless codes are now attracting applications in video streaming applications. In video streaming for cognitive radio [32], rateless error coding compensates an opportunistic secondary video packet source from interference by the primary occupant of the wireless channel. In essence, this is the same network coding technique as applied in [33], because it allows a set of subchannels distributed across the available wireless spectrum to stream scalable video without coordination between the sources. In [32, 33], the rateless code was applied to packets and not blocks within the packets, as also was the application to unicast video streaming over the Internet in [10]. Because the symbols of a Fountain code are generated from a sparse distribution, any uncoordinated sources are unlikely to construct the same two symbols. However, this is not how the present paper proposes to employ rateless coding, as in [32, 33] there are multiple uncoordinated channels, whereas herein there is a single channel that is coordinated with the receiver. For the same reason, BlueTorrent, concerning which [34] mentions in passing network coding for Bluetooth, is not related to the current paper.

In [35], it was observed that classic error control methods work poorly in terms of energy conservation, in line with similar comments in Section 1. It was proposed in [35] that the channel should be probed to find the error conditions, whereupon the level of ARQ retransmissions is adjusted. However, the volatility of the wireless channel may make measurements unreliable. The work in [36] proposed a scheme of error control which varied according to the channel conditions and to the relative energy budget for RS coding and selective repeat ARQ. As in our paper, a two-state type model allowed (Rayleigh) fading conditions to be modeled, in way that is independent of packet size. It was found that there was a threshold, beyond which FEC was

necessary, despite the increase in energy budget. In [37], packet-level FEC (not block-level as in our paper) and power allocation are jointly optimized across cellular radio. The work combines layered video coding with FEC, with the degree of protection varying according the priority of the layer. The layers actually transmitted depend on the power resources of the sender.

In [38], rateless coding is selected for reasons of reduced decode computational complexity in an energy reduction scheme for wireless mesh networks. This scheme is compared to network coding and similar schemes for data broadcast. Others have noticed the advantage of rateless coding for energy conservation, for example in [39], rateless coding is applied in a sensor network context but for data not video and from the reduced decode complexity point-of-view and not necessarily because of reduced transmission overhead.

2.3. Bluetooth

Bluetooth's short range, FHSS transmission, centralized multiple access control through TDMA and master-driven TDD system means it is less prone to interference from other Bluetooth networks. Bluetooth employs variable-sized packets up to a maximum of five frequency-hopping timeslots of 625 microseconds in duration. Every Bluetooth frame consists of a packet transmitted from a sender node over 1, 3, or 5 timeslots, while a receiver replies with a packet occupying at least one slot, with the result that each frame has an even number of slots. Bluetooth v. 2.1's EDR supports gross air rates of 3.0 Mb/s and 2.0 Mb/s, in addition to the original 1.0 Mb/s basic rate. EDR allows higher quality video streaming, while a choice of channel rates allows the lower rate to be employed in adverse channel conditions, retaining the basic channel rate for packet headers and very poor (<10 dB signal-to-noise ratio (SNR)) channels.

Because of packet quantization effects, the Bluetooth asynchronous connection-less (ACL) packet sizes become significant. From Figure 1, it is apparent that selection of a small packet size results in relatively low throughput, and fully-filled packets are favored over partially-filled packets. Therefore, we assume that a Bluetooth controller's behavior is that, given a maximal Bluetooth packetisation scheme, packets up to the maximum user payload will be formed. However, if the arriving packets do not justify the preset maximal scheme a reduced scheme is applied. The packet structure of Bluetooth is further described in the course of the following section.

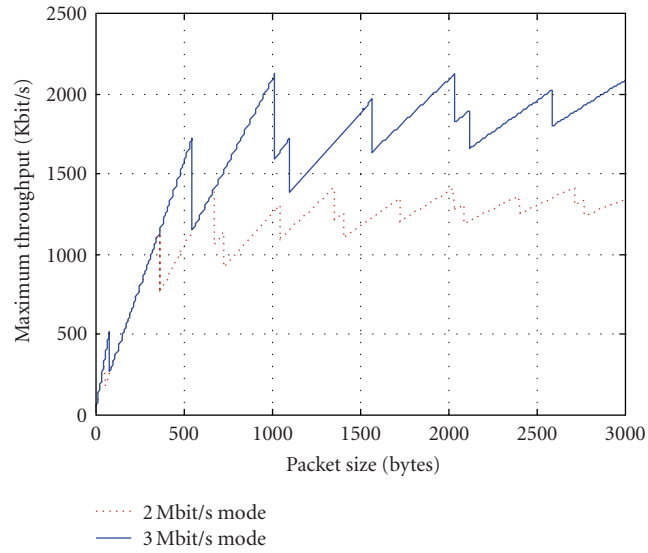


Figure 1: Throughput quantization effects for Bluetooth EDR modes.

3. METHODOLOGY

This section now describes our block-based rateless coding system and then goes on to describe the scenario that is simulated.

3.1. Redundant block transmission algorithm

Figure 2 shows the partition of a video-bearing Bluetooth packet payload into three parts: (1) a variable-sized redundant block portion, with the blocks within this portion generated by the rateless algorithm from prior packets; (2) the data of the next packet divided into blocks with an additional ε blocks generated by the rateless algorithm, as $k(1+\varepsilon)$ blocks are required for reconstruction of the original k blocks with high probability; (3) a CRC which is a default part of a Bluetooth packet but which we assume is applied to the decoded k blocks of the current packet. Upon failure of the CRC, additional blocks are requested from the sender and these are sent in the first part of the next packet together with any other blocks from yet to be reconstructed packets.

The complete block-based rateless coding algorithm is described in detail in Figure 3. Assume initially that just the one prior packet has failed then redundant blocks are now piggybacked upon the current packet to add to the original $k(1+\varepsilon)$ blocks already transmitted to increase the probability of a successful decode. After an attempted decode, the CRC of that prior packet is applied to establish whether there has been an erasure. If there is an erasure additional blocks are requested through Bluetooth's fast ARQ mechanism, unless the duration of block retransmissions already exceeds the display deadline of the video frame of which that packet's data forms a part. The display deadline in the simulations was set to a constant d number of retries.

Critical to the operation of rateless error correction is the number of blocks contained in part 1 of a Bluetooth packet payload. Bluetooth's native block-based FEC scheme employs 15-bit blocks as a consequence of its error coding scheme (refer forward to Section 3.2) and, for comparison and ease of implementation, 15-bit blocks are also employed in the block-based rateless coding scheme. If redundant blocks are to be sent then a minimum and a maximum number of 15-bit blocks is defined, being 5 and 50, respectively, in the simulations of Section 4. Leaving aside initialisation packets, the starting number of redundant blocks was the minimum number (five blocks) in our simulations. Upon receipt of a consecutive sequence of n

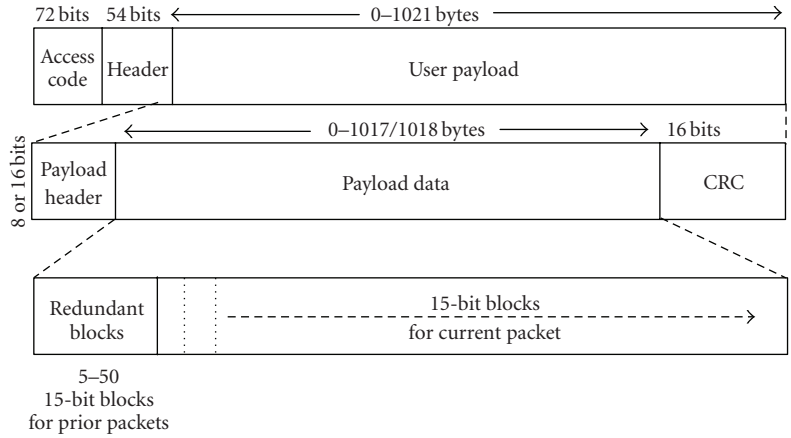


Figure 2: Bluetooth packetisation structure, showing the incorporation of redundant blocks into the payload.

successfully-transmitted packets, 100 in the simulations, then the limit is reduced by one. Upon a failure to reconstruct any packet after the dth transmission of its blocks the number of redundant blocks included in a packet is increased in the future by a factor α, set to 1.5 in the simulations. This conservative policy for a volatile channel results in a rapid increase in redundancy when uncorrectable errors first occur.

If more than one prior packet of the same frame type has errors then the redundant block allowance is split according to the proportion of retransmissions remaining for each packet, allowing for some irregularity due to the need to apportion an integer number of blocks. The ratio is calculated as a proportion of factor d. A simple acknowledgment of the differing importance of video frame types was made by altering the allocation in the ratio 3:2:1 for I-, P-, and B-frame packets, respectively. (I- and predictively code P-frames are reference frames (refer to Section 1) while bi-predicted B-frames themselves have no predictive role.) Other priority-based schemes are possible.

Table 1: Bluetooth EDR ACL packet types, including additional DM packets. (Length and master to slave bitrates, for a single ACL master-slave logical link, with DM = data medium rate (FEC added) and DH = data high rate (no FEC). 2-DH3 is 2.0 Mbps modulation three time-slot packet.)

Packet type	User payload in bytes	Asymmetric max. rate (Kbps)
2-DM1	0–36	230.4
2-DM3	0–245	782.9
2-DM5	0–453	965.7
2-DH1	0–54	345.6
2-DH3	0–367	1174.4
2-DH5	0–679	1448.5
3-DM1	0–55	354.1
3-DM3	0–368	1184.3
3-DM5	0–681	1452.0
3-DH1	0–83	531.2
3-DH3	0–552	1776.4
3-DH5	0–1021	2178.1

3.2. Other FEC schemes

Bluetooth already has FEC-bearing data medium (DM) packets, available at the basic rate of version 1 in the event of poor SNR. The Bluetooth system of FEC and error control is already extensively described in [22] and summarized in [19]. To avoid extending the length of this paper, the reader is referred to those documents, while this section describes variants of the Bluetooth FEC system. An expurgated (15, 10) Hamming code is applied to 15-bit blocks and can cope with burst sizes of two, depending on decoder [40]. As a point of comparison with rateless codes, it is supposed that the DM packet scheme is extended to the EDR transmission modes. This extension scheme has already been proposed in [41] and is entirely feasible. Table 1 summarizes the additional EDR asynchronous connection-less (ACL) mode packet types currently available (according to the specification), as well as EDR DM-type packets in the event that symbol-level FEC were to be added to EDR.

In the simulations of Section 5, the FEC-bearing packets are the 3DM packets from Table 1. Additional comparison is made with an adaptive FEC-bearing scheme that assumes perfect channel knowledge. FEC-bearing packets are only selected when the channel enters a bad state. The adaptive scheme is introduced as it has the ability to save energy by reducing the overhead when channel conditions ease. Because the native FEC-bearing Bluetooth scheme already has a rate of 1/3, that is, considerable overhead, automatic ARQ is normally turned off to avoid increasing the FEC overhead. (ARQ is effectively turned off [22] by setting the Bluetooth flush timeout to a minimal value.)

In addition, a comparison is made with a simple packet-based rateless coding scheme in which 10% redundancy is added (i.e., one in ten packets is redundant), compared to 33% redundancy for fixed FEC. The block-based rateless and the adaptive FEC schemes do, of course, have a varying percentage of redundancy.

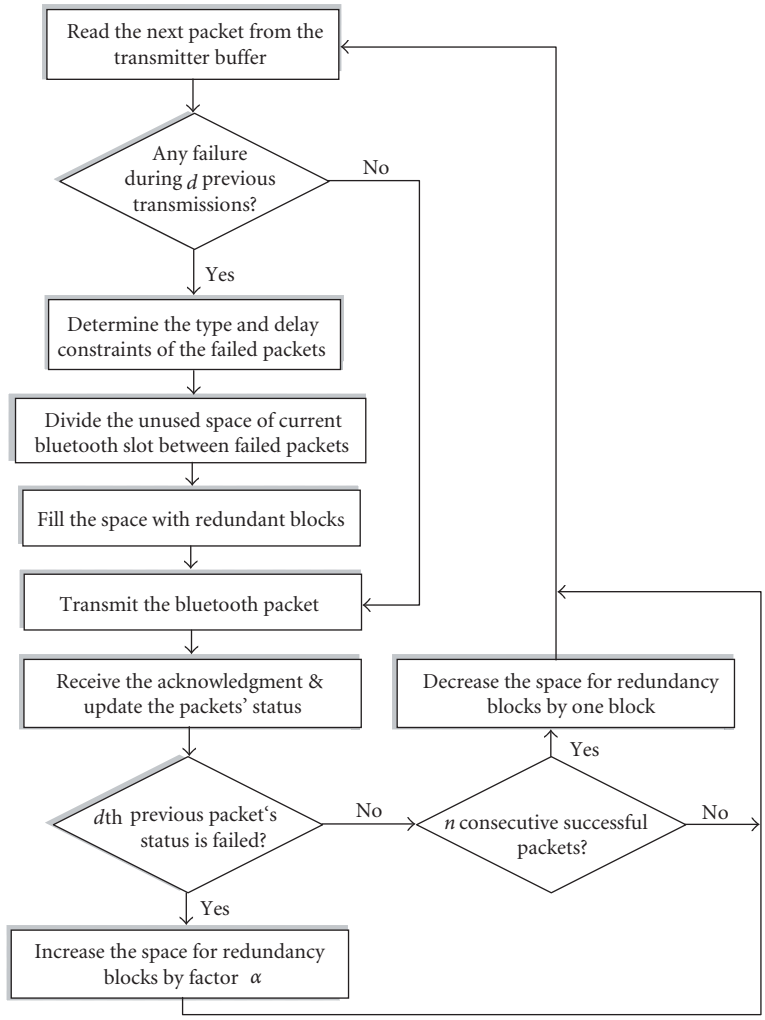


Figure 3: Block-based rateless coding algorithm, where a packet failure is a failure to reconstruct a packet with redundant blocks.

3.3. Channel model

A Gilbert-Elliott two-state discrete-time ergodic Markov chain models the wireless channel error characteristics between a Bluetooth master and slave node. The mean duration of a good state, T_g, was set at 2 seconds and in a bad state, T_b was set to $a \times T_g$, where a is a parameter which is varied to alter the duration of bad states. In units of 625 microseconds (the Bluetooth time slot duration), $T_g = 3200$ which implies from

$$T_g = \frac{1}{1 - Pgg}, \qquad T_b = \frac{1}{1 - Pbb} \tag{1}$$

that, given the current state is good (g), Pgg, the probability that the next state is also g, is 0.9996875. Both good and bad states are modeled by a Rayleigh channel with the mean SNR being 35 ± 1 dB and 25 ± 1 dB in the g and b states, respectively.

3.4. Wearable network

The Bluetooth network in Figure 4 contains the two biosensors, the video camera source and the HMD, along with an external source which may act as a means of exchanging biosensor data with other operatives and as a source of external sensor data. The video source is assumed to be of variable bit-rate (VBR) to ensure higher quality within the restrictions of the available bandwidth. The encoded video is transferred to the central node where, after decoding, augmentation of the display takes place along with moderation of that display in line with interpretation of biosensor data. Notice that if the display contains text then good resolution is needed. The EEG and ECG biosensors are assumed to output at a constant bit rate (CBR). The external source was modeled as an on-off source in the ratio 1 second on to 2 seconds off with its bitrate divided equally in the two directions. However, polling packets from the master node and null return packets to the external source will occupy a significant portion of the available bandwidth when the external source is off. The assumed data rates of the sources are detailed in Table 2 along with packet sizes. The rate for the ECG source is taken from [42]. ECG, EEG, and the external source all used 3DH5 packets (refer to Table 1). For video, the payload size is determined by the amount of redundant blocks required

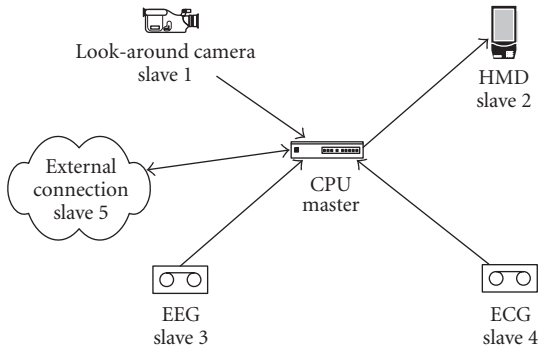


Figure 4: Bluetooth wireless sensor network, showing master and slave nodes.

Table 2: Traffic flows across the simulated wearable network (from Figure 4, S1 = Slave 1, S2 = Slave 2,... M = master node).

Communication	Mean bitrate (Kb/s)	Type (packet size)
S1 to M	256	VBR
M to S2	256	VBR
S3 to M	1000	CBR (800 B)
S4 to M	3.6	CBR (800 B)
S5 to M	50	CBR (800 B)

for correction of the prior failed packets. Thereafter, 3DH5 packets are selected.

3.5. Simulation setup

This research employed the University of Cincinatti Bluetooth (UCBT) extension to the well-known ns-2 network simulator (v. 2.28 used). The UCBT extension supports Bluetooth EDR but is also built on the air models of previous Bluetooth extensions such as BlueHoc from IBM and Blueware. The Gilbert-Elliott channel model was coded in C++ to be called by an ns-2 otcl script. All links were set at the maximum EDR 3.0 Mbps gross air rate. The UCBT simulation parameter for antenna type was set to omnidirectional, and for the distance between devices was set to 1 m, as is appropriate for a wearable network. (Experiments were also conducted with the distance set 2 m with no noticeable difference in the results.) All other settings were the default ones. Simulation runs were each repeated 100 times, and the results averaged to produce summary statistics.

The simulations were principally carried out with input from an MPEG-2 encoded bitstream at a mean rate of 256 kbps for a 30-second video clip with moderate motion. PSNR was found by reconstructing with a reference MPEG-2 decoder. The display rate was 25 frame/s, resulting in 750 frames in each run. The source video was common intermediate format (CIF)-sized (352 × 288 pixels) with a GOP structure of $N = 12$, and $M = 3$ (when in standard codecs N designates the GOP length and M is the number of pictures between anchor pictures). In [43], it was demonstrated that forming fully-filled Bluetooth packets outweighed the need to preserve MPEG-2 slice boundaries, which over the fixed Internet are preserved for error-resilience purposes. Therefore, as mentioned previously, 3DH5 packets are selected by default and, thereafter, fully-filled packets of the 3DH type are formed from the arriving encoded video stream. The results of simulating the video stream under these circumstances are now described.

4. SIMULATION RESULTS

Experiments were conducted streaming the video of Section 3.5. Metrics were recorded across both hops in Figure 4 over which video was streamed. For example, packet loss is recorded as a total across both hops. A varying number of redundant blocks were included in the packet payload if one or more prior packets were found to have failed. For any one packet in error, retransmissions continued until the number of retransmissions, d, exceeded ten, assuming that after ten attempts at reconstructing the packet the display deadline would be exceeded. At a frame rate of 25 frame/s, a frame is displayed every 0.040 seconds, while ten retransmissions take 0.375 seconds. Assuming a worse case of each of 18 slices in an MPEG2 CIF-sized frame then a small playout buffer of about 20 frames is adequate even if all 18 were in error. A large buffer would result in passive energy drain, reducing the lifetime of the system before recharging is necessary. After d is exceeded then the packet is declared as lost. A send buffer size of fifty packets was sufficient to avoid packet loss by buffer overflow, though increasing the video arrival rate could change that.

Figure 5 shows how there is a sharp reduction in the packet loss ratio (the number of lost packets to the number of packets transmitted) at a given average SNR for a relatively small investment in redundant blocks. All loss occurred through exceeding the retransmit limit. Notice also that in Figure 5 that there is a single-state Rayleigh channel, whereas later tests use the two-state channel model of Section 3.3. Figure 6 shows that as the retransmission depth, d, increases then there is a higher chance of recovering a previously failed packet, as packet losses decline with d. However, d cannot be chosen arbitrarily as it can lead to missed display deadlines. Ideally, d should match the playout buffer size, as an arbitrary choice can lead to missed display deadlines.

In Figure 7, the augmented cognition traffic and the other traffic sources from Table 2 are turned on, while the packet loss ratio for each of the other FEC-bearing schemes of Section 3.2 is compared with the proposed rateless coding scheme. The packet loss ratio is the ratio of packets lost against total packets transmitted in the video streams. The loss ratio is adjudged against worsening channel conditions as regulated by parameter a from Section 3.3. From Figure 7, it is apparent that the proposed rateless scheme outperforms the native schemes and increasingly so as the bad state durations increase. In comparison to a simple packet-based scheme, the block-based rateless scheme reduces packet loss because it is able to adapt to increasing burst lengths. The packet-based rateless coding scheme has a reduced level of redundancy compared to the fixed FEC scheme, resulting in more packet loss. The adaptive FEC scheme is less efficient

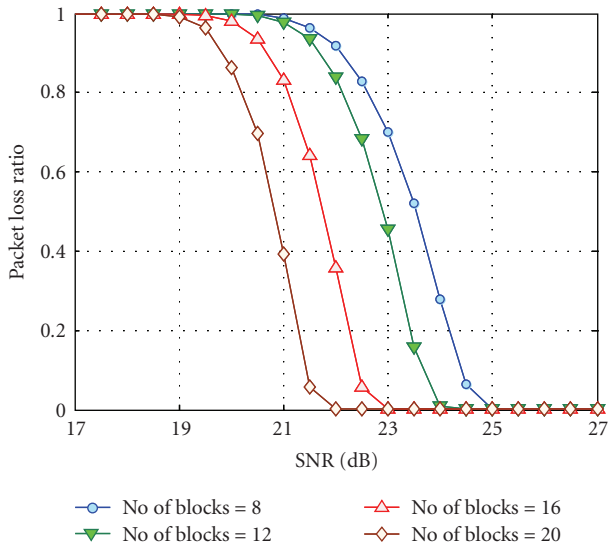


Figure 5: Packet loss ratio according to the number of redundant blocks in a Rayleigh channel with varying SNR.

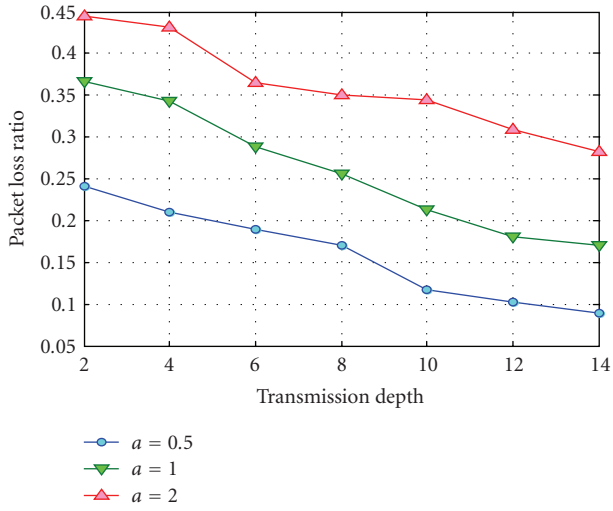


Figure 6: Packet loss ratio according to the transmission depth (d), for varying duration (indexed by a) of bad state in a two-state Rayleigh channel.

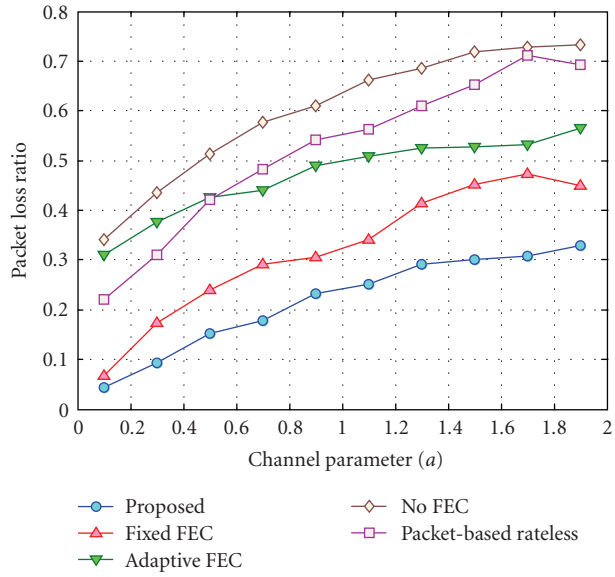


Figure 7: Comparison of packet loss for block-based rateless coding and various FEC-bearing streams with competing biosensor traffic, for varying duration (indexed by a) of bad state in a two-state Rayleigh channel.

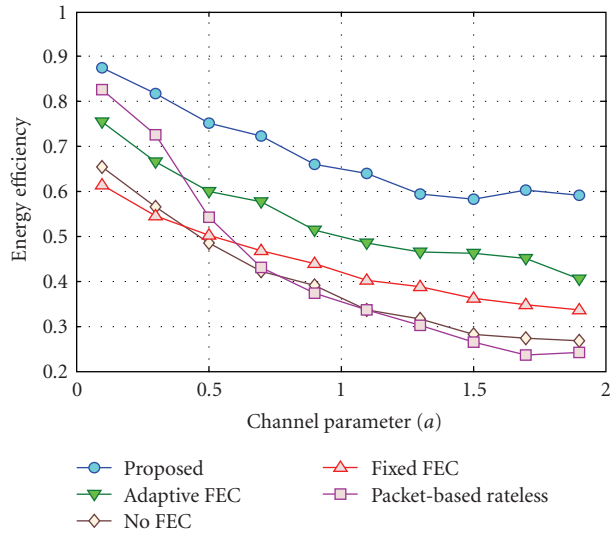


Figure 8: Comparison of energy efficiency for block-based rateless coding and various FEC-bearing streams with competing biosensor traffic, for varying duration (indexed by a) of bad state in a two-state Rayleigh channel.

at preventing errors compared to the packet-based rateless scheme but is better able to compensate when the channel parameter increases.

The various schemes were also compared, see Figure 8, in terms of energy efficiency. The energy efficiency is the delivery ratio adjusted according to the transmission power. However, assuming the transmission power is normalized to one, the energy efficiency is the same as the delivery ratio, that is, the ratio of the data successfully transmitted to the total data transmitted. Though the "no FEC" plot involves no overhead from FEC, it still has a poor energy saving efficiency compared to the proposed scheme because of the fewer bits transmitted successfully. Adaptive FEC is relatively better at energy reduction than fixed FEC but, of course from Figure 7, the number of unrecoverable packets is greater. In better channel conditions, block-based and packet-based rateless coding have similar energy efficiencies. However, when the channel conditions worsen, fewer packets are successfully received through the packet-based variant, causing the energy efficiency to deteriorate sharply.

According to the AR scenario, packets make two hops across the Bluetooth network before display on the HMD. Processing delay at the master node is neglected. The data rates are the same as those in Table 2. A playout buffer of

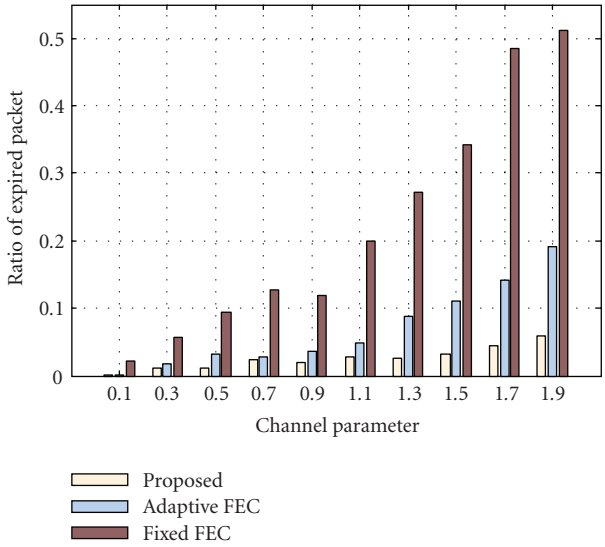


Figure 9: Proportion of packets missing frame display deadlines according to error correction scheme.

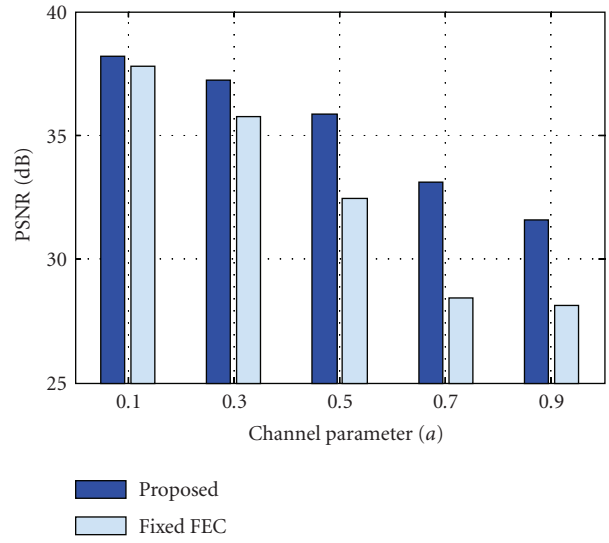


Figure 10: Video quality comparison for transmission using the rateless coding scheme and the EDR FEC-scheme.

size fifty packets is assumed, which is the same size as used in [44] for a mobile application. Larger playout buffers lead to memory drain through passive energy consumption. In these tests, ARQ is enabled but set to a maximum number of four retransmission requests, while the blocks in the rateless scheme can be resent up to ten times. From the figure, fixed FEC packets suffer from considerable delay when the channel conditions worsen, as they must be retransmitted when error correction is insufficient. Adaptive FEC fares better as the level of FEC can be adjusted to cater for the longer error bursts. However, though in the proposed scheme blocks may be repeatedly piggybacked upon outgoing packets, they miss fewer display frame display deadlines, which is the relevant criterion for video transmission. The impact on packet arrivals is examined in Figure 9. The figure records the percentage of packets that miss the display deadline for the block-based rateless and FEC schemes when the channel bad state duration increases in duration.

In Figure 10, a comparison is made between the delivered video qualities for selected bad state durations for which PSNR is of a reasonable level. The figure shows that in terms of delivered video quality the rateless scheme also outperforms the Bluetooth FEC scheme when applied to EDR packets. The relative improvement increases with lengthening bad state duration.

Figure 11 is a comparison between the overhead incurred by the proposed block-based rateless scheme (with 12 redundant blocks) and a simple packet-based rateless coding scheme. A single-state Rayleigh channel is simulated as also occurs in Figure 5. Both schemes must sustain the small percentage ε (10%) overhead from a finite length rateless coding scheme mentioned in Section 2.1. Figure 11 shows that the overhead of the proposed scheme remains reduced at much lower SNRs compared to the simple packet-based system. The channel conditions are responsible for the sudden increases in overhead (just as in Figure 5 in which there are rapid increases in the packet loss ratio). Though the effect is exaggerated in these conditions, the gain in reduced overhead is apparent.

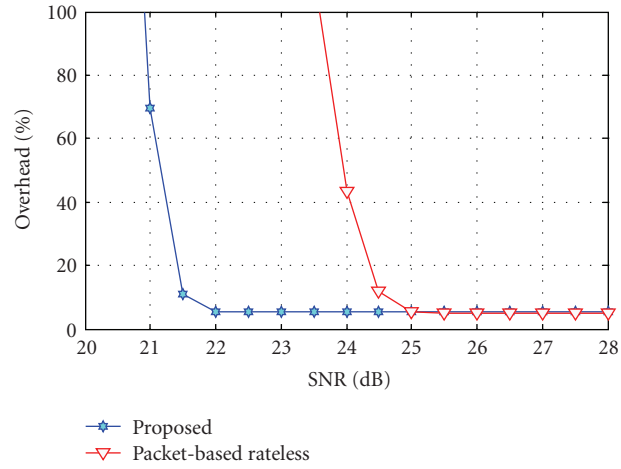


Figure 11: Overhead with block-based and simple packet-based rateless coding in a Rayleigh channel with varying SNR.

5. CONCLUSIONS

Augmented reality is normally associated with wearable computers. It is now feasible to provide wearable wireless networks with sufficient bandwidth capacity to allow a video stream from camera to head mounted display. An important class of wearable computer users is those that can be broadly termed emergency workers. As these operatives often work under stressful conditions, it becomes necessary to monitor the worker with biosensors. Some of these sensors such as EEG have high bitrates leading to delay for other traffic and reduced video data rates. Therefore, it is important

to model the wearable network, which we have done by considering the video stream from camera via the CPU, where augmentation takes place, to the display. Some form of forward error control is required, because a wireless channel is also prone to interference and noise and because lost packets have an enduring effect upon the compressed video stream (until the next intrapicture refresh point). We have proposed block-based rateless coding, which, from the paper's results, compared to block-based forward error correction variants can jointly improve energy consumption, packet arrival latency, and delivered video quality. Block-based rateless channel coding is well suited to a Bluetooth wireless network, because feedback to the centralized packet scheduler occurs immediately and with little propagation delay. Simple packet-based rateless schemes are less suited, as without feedback they are unable to adjust to worsening channel conditions, despite advantages in coding efficiency. However, to construct a scheme for block-based rateless coding requires recursive piggy-backing of redundant blocks onto outgoing packets. This is a cross-layer approach, whereas simple packet-based rateless coding is applied at the application layer. Nevertheless, this paper reports an implemented scheme that results in around 5 dB gain in video quality in poor channel conditions.

ACKNOWLEDGMENT

This work was supported by the EPSRC, UK, under Grant no. EP/C538692/1.

REFERENCES

[1] J. P. Rolland and H. Fuchs, "Optical versus video see-through head-mounted displays," in *Fundamentals of Wearable Computers and Augmented Reality*, pp. 113–150, Lawrence Erlbaum, Hillsdale, NJ, USA, 2001.

[2] J. Sweller, "Cognitive load during problem solving: effects on learning," *Cognitive Science*, vol. 12, no. 2, pp. 257–285, 1988.

[3] C. Ashwnanden and J. Stelovsky, "Measuring cognitive load with EventStream software framework," in *Proceedings of the 36th Annual Hawaii International Conference on System Sciences (HICSS '03)*, p. 5, Big Island, Hawaii, USA, January 2003.

[4] M. Klan, T. Riedel, H. Gellersen, et al., "LifeNet: an ad-hoc sensor network and wearable system to provide firefighters with navigation support," in *Proceedings of the 9th International Conference on Ubiquitous Computing*, pp. 124–127, Innsbruck, Austria, September 2007.

[5] C. C. Tappert, A. S. Ruocco, K. A. Langdorf, et al., "Military applications of wearable computers and augmented reality," in *Fundamentals of Wearable Computers and Augmented Reality*, pp. 625–647, Lawrence Erlbaum, Hillsdale, NJ, USA, 2001.

[6] M. Grootjen, M. A. Neerincx, J. C. M. van Weert, and K. P. Truong, "Measuring cognitive task load on a naval ship: implications of a real world environment," in *Foundations of Augmented Cognition*, L. Reeves and D. D. Schmorrow, Eds., vol. 4565 of *Lecture Notes in Computer Science*, pp. 147–156, Springer, Berlin, Germany, 2007.

[7] D. J. C. MacKay, "Fountain codes," *IEE Proceedings: Communications*, vol. 152, no. 6, pp. 1062–1068, 2005.

[8] J. Afzal, T. Stockhammer, T. Gasiba, and W. Xu, "System design options for video broadcasting over wireless networks," in *Proceedings of the 3rd IEEE Consumer Communications and Networking Conference (CCNC '06)*, vol. 2, pp. 938–943, Las Vegas, Nev, USA, January 2006.

[9] J. Afzal, T. Stockhammer, T. Gasiba, and W. Xu, "Video streaming over MBMS: a system design approach," *Journal of Multimedia*, vol. 1, no. 5, pp. 25–35, 2006.

[10] S. Ahmad, R. Hamzaoui, and M. Al-Akaidi, "Robust live unicast video streaming with rateless codes," in *Proceedings of the 16th International Packet Video Workshop (PACKET VIDEO '07)*, pp. 78–84, Lausanne, Switzerland, November 2007.

[11] J.-C. Chen, K. M. Sivalingam, P. Agrawal, and S. Kishore, "A comparison of MAC protocols for wireless local networks based on battery power consumption," in *Proceedings of the 17th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM '98)*, vol. 1, pp. 150–157, San Francisco, Calif, USA, March-April 1998.

[12] M. Tamai, T. Sun, K. Yasumoto, N. Shibata, and M. Ito, "Energy-aware video streaming with QoS control for portable computing devices," in *Proceedings of the 14th International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV '04)*, pp. 68–73, Cork, Ireland, June 2004.

[13] A. Shokrollahi, "Raptor codes," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2551–2567, 2006.

[14] W. C. Ark, D. C. Dryer, and D. J. Lu, "The emotion mouse," in *Proceedings of the 8th International Conference on Human-Computer Interaction (HCI '99)*, pp. 818–823, Munich, Germany, August 1999.

[15] Avant 4000 Brochure, "Wearable Digital Pulse Oximetry with Bluetooth Wireless Technology," Nonin Medical Inc., Plymouth, Minn, USA, 2005.

[16] G. G. Messier and I. G. Finvers, "Traffic models for medical wireless sensor networks," *IEEE Communications Letters*, vol. 11, no. 1, pp. 13–15, 2007.

[17] R. Rosipal, B. Peters, G. Kecklund, et al., "EEG-based drivers' drowsiness monitoring using a hierarchical Gaussian mixture model," in *Foundations of Augmented Cognition*, vol. 4565 of *Lecture Notes in Computer Science*, pp. 294–303, Springer, Berlin, Germany, 2007.

[18] C. S. Ikehara, E. Biagioni, and M. E. Crosby, "Ad-hoc wireless body area network for augmented cognition sensors," in *Foundations of Augmented Cognition*, vol. 4565 of *Lecture Notes in Computer Science*, pp. 38–46, Springer, Berlin, Germany, 2007.

[19] J. Haartsen, "The bluetooth radio system," *IEEE Personal Communications*, vol. 7, no. 1, pp. 28–36, 2000.

[20] N. Golmie, D. Cypher, and O. Rebala, "Performance analysis of low rate wireless technologies for medical applications," *Computer Communications*, vol. 28, no. 10, pp. 1266–1275, 2005.

[21] E. Ferro and F. Potortì, "Bluetooth and Wi-Fi wireless protocols: a survey and a comparison," *IEEE Wireless Communications*, vol. 12, no. 1, pp. 12–26, 2005.

[22] Specification of the Bluetooth System — 2.1 + EDR, July 2007, http://www.bluetooth.com/.

[23] J. Shelley and R. W. Backs, "Categorizing EEG waveform length in simulated driving and working memory dual-tasks using feed-forward neural networks," in *Augmented Cognition: Past, Present, and Future*, pp. 155–161, Strategic Analysis, Arlington, Va, USA, 2006.

[24] M. Raju, "Heart rate and EKG monitor using the MSP430FG439," Tech. Rep. SLAA280, Texas Instruments, Dallas, Tex, USA, October 2005.

[25] A. Koski, "Lossless ECG encoding," *Computer Methods and Programs in Biomedicine*, vol. 52, no. 1, pp. 23–33, 1997.

[26] M. Modarreszadeh and R. N. Schmidt, "Wireless, 32-channel, EEG and epilepsy monitoring system," in *Proceedings of the 19th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, vol. 3, pp. 1157–1160, Chicago, Ill, USA, October-November 1997.

[27] S. Mann, "Wearable computing: a first step towards personal imaging," *Computer*, vol. 30, no. 2, pp. 25–32, 1997.

[28] T. G. Zimmerman, "Personal area networks: near-field intrabody communication," *IBM Systems Journal*, vol. 35, no. 3-4, pp. 609–617, 1996.

[29] R. LaRowe and C. Elliott, "Computer networks for wearable computing," in *Fundamentals of Wearable Computers and Augmented Reality*, pp. 715–745, Lawrence Erlbaum, Hillsdale, NJ, USA, 2001.

[30] A. P. J. Hum, "Fabric area network—a new wireless communications infrastructure to enable ubiquitous networking and sensing on intelligent clothing," *Computer Networks*, vol. 35, no. 4, pp. 391–399, 2001.

[31] M. Luby, "LT codes," in *Proceedings of the 34rd Annual IEEE Symposium on Foundations of Computer Science*, pp. 271–280, Vancouver, Canada, November 2002.

[32] H. Kushwaha, Y. Xing, R. Chandramouli, and H. Heffes, "Reliable multimedia transmission over cognitive radio networks using fountain codes," *Proceedings of the IEEE*, vol. 96, no. 1, pp. 155–165, 2008.

[33] J.-P. Wagner, J. Chakareski, and P. Frossard, "Streaming of scalable video from multiple servers using rateless codes," in *Proceedings of IEEE International Conference on Multimedia and Expo (ICME '06)*, pp. 1501–1504, Toronto, Canada, July 2006.

[34] S. Jung, U. Lee, A. Chang, D.-K. Cho, and M. Gerla, "BlueTorrent: cooperative content sharing for bluetooth users," in *Proceedings of the 5th Annual IEEE International Conference on Pervasive Computing and Communications (PerCom '07)*, pp. 47–56, White Plains, NY, USA, March 2007.

[35] M. Zorzi and R. R. Rao, "Error control and energy consumption in communications for nomadic computing," *IEEE Transactions on Computers*, vol. 46, no. 3, pp. 279–289, 1997.

[36] P. Lettieri, C. Fragouli, and M. B. Srivastava, "Low power error control for wireless links," in *Proceedings of the 3rd Annual ACM/IEEE International International Conference on Mobile Computing and Networking (MobiCom '97)*, pp. 139–150, Budapest, Hungary, September 1997.

[37] S. Zhao, Z. Xiong, and X. Wang, "Joint error control and power allocation for video transmission over CDMA networks with multiuser detection," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 6, pp. 425–437, 2002.

[38] N. Rahnavard, B. N. Vellambi, and F. Fekri, "Efficient broadcasting via rateless coding in multihop wireless networks with local information," in *Proceedings of the International Wireless Communications and Mobile Computing Conference (IWCMC '07)*, pp. 85–90, Honolulu, Hawaii, USA, August 2007.

[39] A. W. Eckford, J. P. K. Chu, and R. S. Adve, "Low-complexity cooperative coding for sensor networks using rateless and LDGM codes," in *Proceedings of IEEE International Conference on Communications (ICC '06)*, vol. 4, pp. 1537–1542, Istanbul, Turkey, July 2006.

[40] L. Lampe, M. Jain, and R. Schober, "Improved decoding for bluetooth systems," *IEEE Transactions on Communications*, vol. 53, no. 1, pp. 1–4, 2005.

[41] L.-J. Chen, T. Sun, and Y.-C. Chen, "Improving bluetooth EDR data throughput using FEC and interleaving," in *Proceedings of the 2nd International Conference on Mobile Ad-hoc and Sensor Networks (MSN '06)*, pp. 724–735, Hong Kong, December 2006.

[42] U. Varshney and S. Sneha, "Patient monitoring using ad-hoc wireless networks: reliability and power management," *IEEE Communications Magazine*, vol. 44, no. 4, pp. 49–55, 2006.

[43] R. Razavi, M. Fleury, E. Jammeh, and M. Ghanbari, "An efficient packetization scheme for bluetooth video transmission," *Electronic Letters*, vol. 42, no. 20, pp. 1143–1145, 2006.

[44] Q. Li and M. van der Schaar, "Providing adaptive QoS to layered video over wireless local area networks through real-time retry limit adaptation," *IEEE Transactions on Multimedia*, vol. 6, no. 2, pp. 278–290, 2004.

Hindawi Publishing Corporation
Advances in Multimedia
Volume 2008, Article ID 841590, 14 pages
doi:10.1155/2008/841590

Research Article

Providing QoS for Networked Peers in Distributed Haptic Virtual Environments

Alan Marshall, Kian Meng Yap, and Wai Yu

School of Electronics, Electrical Engineering and Computer Science, Queen's University Belfast, Belfast BT7 1NN, Northern Ireland, UK

Correspondence should be addressed to Kian Meng Yap, m.yap@qub.ac.uk

Received 29 February 2008; Accepted 17 June 2008

Recommended by Mohammed Ghanbari

Haptic information originates from a different human sense (touch), therefore the quality of service (QoS) required to support haptic traffic is significantly different from that used to support conventional real-time traffic such as voice or video. Each type of network impairment has different (and severe) impacts on the user's haptic experience. There has been no specific provision of QoS parameters for haptic interaction. Previous research into distributed haptic virtual environments (DHVEs) have concentrated on synchronization of positions (haptic device or virtual objects), and are based on client-server architectures. We present a new peer-to-peer DHVE architecture that further extends this to enable force interactions between two users whereby force data are sent to the remote peer in addition to positional information. The work presented involves both simulation and practical experimentation where multimodal data is transmitted over a QoS-enabled IP network. Both forms of experiment produce consistent results which show that the use of specific QoS classes for haptic traffic will reduce network delay and jitter, leading to improvements in users' haptic experiences with these types of applications.

1. INTRODUCTION

There has been recent interest in the transmission of multimodal information over the Internet [1], and in particular the transmission of haptic information [2, 3] (haptic is sense of touch and force feedback and comes from the Greek word "haptikos" *to grasp, touch*. Telehaptics concerns remote haptic operations over network connections). The introduction of the haptic sense of touch (i.e., reflected force) refers to the perceptual kinaesthesia sensing of events such as heat, pressure, force, or vibration. This paper involves research into how new types of distributed applications which involve haptic devices, in addition to visual and aural information, can be carried over the Internet. Specifically, it considers an emerging class of applications that enable users to interact haptically with virtual environments.

By definition, a virtual environment (VE) is a space that provides users with the illusion of acting in a real world. However in addition to audio and visual information, the provision of haptic feedback (the sense of touch) can profoundly improve the way we interact with virtual environments. Systems that support interfaces between a haptic device and a virtual environment are called haptic virtual environments (HVEs). HVE uses include military and space exploration; the sense of touch will also enable blind people to interact with each other within a virtual environment. The HVE modalities include graphics (and possibly video), sound, and force. Recent research [2, 3] has shown that to have a satisfying experience in interacting with an HVE, the graphics and haptic update rates need to be maintained at around 30 Hz and 1 KHz, respectively. In distributed HVEs (DHVE) for remote collaborations, the haptic device is separated from the virtual environment and remotely affects and manipulates it. In DHVEs, one or multiple users may interact with the virtual environment, and possibly with other users with haptic devices. Users may take turns in manipulating a virtual object as in collaborative environments or may simultaneously modify the same object as in cooperative environments [4].

Today most haptic applications are standalone systems. However, it is apparent that the ability to provide distributed haptic applications across a universally accessible medium

such as the Internet will dramatically increase their profile to a much wider range of users. Typically, different types of data are exchanged between hosts in DHVE systems (e.g., graphics, audio, positional information, and reflected force). However in order to produce useful performance, haptic applications require feedback within small and guaranteed timescales to achieve stable haptic interactions. It is clear that the best effort service offered by current IP networks is insufficient to meet the needs of the distributed haptic applications. In order to interact successfully with haptic devices, haptic applications require stringent quality of service (QoS) from the network. Impairments such as time delay, packet jitter, and packet loss each have different (and severe) impacts on remote haptic collaborations. This creates significant challenges but also opens up enormous potential for new applications and new network architectures. Therefore, the effective transmission of haptic data in DHVEs is a new research area which presents a number of challenges to the underlying network. Methods to impart some level of prioritised service into the next generation Internet have resulted in the development of new network architectures that provide different quality of service (QoS) levels for different types of traffic. The most prominent QoS architectures and protocols that are now recommended by the IETF include: RSVP, DiffServ, and MPLS [5–7]. However these have been designed to support the transmission of real-time services such as voice and video. The provision of high (or specific) QoS for multisensory communication and effective human computer interaction has not been addressed to date.

Because it originates from a different human sense (touch), the QoS required to support haptic traffic is significantly different from that used to support conventional real-time traffic such as voice or video. To date there has been little or no attempts to quantify or qualify the QoS requirements of DHVEs. Each network impairment affects the sense of force feedback in a particular way. For example, considerable network delay may make the user feel that the virtual object is heavier. Subsequently, the user tries to push the virtual object with larger force. Delay also desynchronizes the different copies of the virtual environment. Jitter makes the user feel that the object's mass is variable, and can also make the system unstable. Packet loss can reduce the power of the force felt by the user. Previous work [8, 9] suggests that the bandwidth of haptic feel is between 500 Hz and 1 KHz, and that users can tolerate end-to-end delays of approximately 30 milliseconds without much degradation to their perception of force. However subsequent trials [2] have established that they are much more sensitive to network jitter; after 3 milliseconds all the users noticed significant degradation of the force impression, generally in the form of instability in the DHVE or oscillations at the surfaces of virtual objects. The effect of packet jitter can be reduced in real-time voice and video applications through the use of a playout or "jitter buffer"; this approach can also be used for haptic traffic, however in this case it can also significantly increase the delay experienced by DHVE application; there is subsequently a need to define the optimum length of the jitter buffer without affecting the quality of the perceived touch interactions. Techniques to reduce this delay in the network will therefore benefit the overall quality of interaction with the DHVE. Recent studies by the authors have shown that the haptic experience deteriorates as network-induced packet delay and packet jitter increases beyond 115 milliseconds and 11 milliseconds, respectively [10]. It is recognized that the performance of multimedia traffic can be improved by using QoS architectures that reduce these network impairments [11], and it is therefore expected that the performance of DHVE-based applications can also be enhanced by applying QoS mechanisms. The work presented in this paper presents an investigation into providing specific network QoS (e.g., Diffserv [5]) for haptic traffic.

2. RELATED RESEARCH

A number of systems have been developed specifically for collaboration in virtual environments, including DIVE, CALVIN, and COVEN [12]. Eraslan [13] and Yu etal. [14] investigate the behaviour of a DIVE application in best effort and differentiated services networks with different queuing disciplines. In this work, a DVE application called virtual environment supporting multiuser interaction over IPv6 (VESIR-6) network is deployed. Some experiments have been conducted on network quality issues such as packet loss and delay. The outcome is that IPv6 offers high-quality network infrastructure possibility for real-time DVEs. Allison [15] considers the effects of varying amounts of simulated constant delay on the performance of a simple collaborative haptic task. The task was performed with haptic feedback alone or combined with visual feedback. Subjects were required to pull a virtual linear string as rapidly as possible, while maintaining a target simulated spring force between their end effectors and that of their collaborators. In their experiment, they incorporate the TiDeC [16] in order to reduce the effect of network delay. TiDeC is a proprietary time-delay compensation system based on prediction of human movement. This does reduce the effect of constant delay, however it neither considers packet loss nor works well with large levels of network jitter. When delay increased, it resulted in a decrease in performance, either in deviation from target spring force and in increased time to complete the task. Performance of TiDeC have been studied and compared with other time compensation techniques, that is, dead reckoning, some results are published in [17]. Traylor [18] describes their recent work with UDP over Ethernet as a communication channel between a remote computer and a custom embedded controller built for a fingerscale 3 DOF force-feedback haptic interface (the 3-DOF ministick). Jay [19] describes an experiment to model the effect of latency across two connected peers sharing a collaborative environment. Although QoS is not considered, their experiment showed that consequences of latency on human interaction can be complex and can vary according to both modality and movement type. The participants in the experiment were clearly able to perceive the effects of delay, and rated the difficulty of the task and the disruption of feedback to be consistently higher with every increment in the level of latency above 50 milliseconds.

Some researchers have attempted to characterize the network parameters required for medical applications that use haptics. In [20] it is reported that a good user experience using a haptic autohandshake requires: 128 kbps bandwidth, packet loss <10%, delay <20 milliseconds and jitter <1 millisecond. Conversely, in order to achieve a good user perception of remote stereo viewing requires: 40 Mbps bandwidth, packet loss <0.01%, delay <100 milliseconds and is not sensitive to jitter. Jeffay [1] and Hudson [21] investigate the problem of supporting continuous data generated by distributed virtual environment (DVEs) applications. They use a nanoManiputor as a haptic device which integrates 3D graphics and force feedback to give a virtual environment interface to scanned probe microscope (SPM). Their experiment described considers the effect of delay and delay-jitter on the haptic force display. Instead of presenting a solid, sharp-edged, stable surface, delayed force feedback results in soft, mushy surfaces, making the use of haptics ineffective or unstable. Their experiments were conducted in a router for three types of flow control: (i) first in, first out (FIFO), (ii) random early detection (RED), and (iii) class-based threshold (CBT). The best QoS was achieved using the CBT flow control with a packet drop-rate of 1.3%, average latency 28.4 milliseconds and an average TCP throughput of 790 kBps.

Nishino et al. [22] propose a new distributed virtual reality architecture to realize a practical system on a dedicated long-haul international network. Some preliminary experiments using the Korea-Japan high-speed research network to validate the proposed method are also mentioned. Their applications handle two tasks, one is a lifting task, and the other is handshaking. The first task can achieve an acceptable rate of completion with up to 32 milliseconds delay, packet loss up to 53%, and jitter up to 60 milliseconds. The second can achieve a reasonable performance with delay up to 13 milliseconds, packet loss up to 40%, and jitter up to 25 milliseconds. Their experimental tasks are mainly based on client-server approaches and the graphical update has to be performed in server side which could result in scalability problems. Cheong [23] uses motion synchronization control with a peer-to-peer shared virtual environment. This type of control can be effective when the round trip delay is less than 300 milliseconds. Lee [24] proposes an intramedia synchronization scheme which adjusts the play out of haptic media according to network delay. Their peer-to-peer architecture describes an adaptive control to reduce the transmission rate by using a buffer and transmission rate control based on number of haptic updates in the buffer.

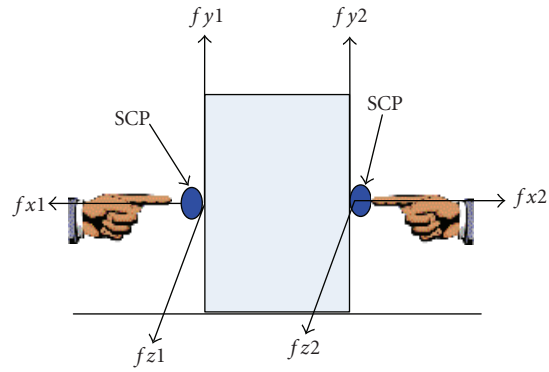
None of the client-server or peer-to-peer architectures mentioned consider applying QoS to improve their performance. While some of the preceding works have investigated the effects of network impairments on specific haptic applications, over specific communications links, to date there have been no attempts to characterize the levels required of the Internet's QoS mechanisms in order that it can provide service for a complete class of haptic applications, that is, DHVEs. In [25], we presented an investigation into how DHVE traffic could be supported over a QoS-enabled network. Here we extend this work to include a consideration of how specific network architectures can be deployed to provide the DHVE, and subsequently the QoS required by these architectures. We also consider how these architectures can be used to support a larger number of traffic flows from these types of applications. Our study has been conducted with both experimental and simulation models in order to study the network QoS characteristics required for haptic media in networks carrying multimodal traffic. The contributions of the work presented in this paper are: (i) a new peer-to-peer DHVE application has been developed in order to generate haptic traffic [10], (ii) from analysis of the traffic, a custom OPNET PDF model [3] has been developed and used in the simulations in order to allow us to examine large-scale haptic traffic, (iii) examination of the behaviour of haptic traffic in multimodal systems when carried over an IP network with and without QoS, (iv) an empirical investigation into the network parameters required for haptic traffic transmission over a QoS-enabled IP network, and subsequently (v) we provide recommendations to improve the transmission of haptic traffic by using Class-based weight fair queue (CBWFQ) and an implementation of Diffserv's code point (DSCP) QoS mechanism. The major research objective is therefore to reduce haptic traffic delay and jitter in distributed multisensory environments. The challenge is to apply QoS to this type of traffic and ensure its effective transmission in real time. Finally, we conclude by stating our findings and future work.

3. DISTRIBUTED HAPTIC VIRTUAL ENVIRONMENT ARCHITECTURES

DHVEs support interfaces between multiple haptic devices and multiple virtual environments regardless of geographical constraints. The force feedback device used in this paper is the PHANToM desktop [26] from SensAble Technologies Inc. It is used to manipulate moving virtual objects and to provide the user with feedback from the virtual environment. The PHANToM desktop has an arm workspace of 16 cm $\times$ 12 cm $\times$ 7 cm and can provide force up to 3.3 N in 3 axis directions; the force computation is based on the spring-damper model [26]. Contact with virtual objects is simulated by computing the force that resists the haptic device's haptic interface point (HIP) from penetrating the virtual object's surface. This approach uses a proxy that transforms the HIP and is referred to as the surface contact point (SCP). The PHANToM desktop has maximum stiffness of (3*1020 N/m) to allow realistic simulation of contact with walls and hard objects. It can generate 1000 packets/s of position and force data during haptic collaboration actions.

DHVEs may have two modes of operation. In collaborative mode users take turns in manipulating the virtual objects while in co-operative mode they can simultaneously modify them [4]. A specific DHVE may operate with just one or both modes. Most collaborative (or co-operative) virtual environments adopt one of two commonly available network distribution architectures: client-server or peer-to-peer. Each architecture has its own specific advantages and shortcomings. Client-server architectures

provide consistency and synchronization among the clients because simulation activities are processed in a centralized server. Also, the required computing power of each client is lower than that required for peer-to-peer systems. The main disadvantage of the client-server approach is that the local view of the environment is only updated after a round-trip to the server, which may impart a significant delay. The client-server architecture also has a scalability problem as the number of clients increase so the load on the server can increase exponentially. We use a peer-to-peer architecture as a DHVE system throughout our studies [10]. Peer-to-peer systems offer the benefits of scalability and decentralized control, however, there are significant challenges associated with synchronizing not only the virtual environments across networked peers, but also the transmitted forces [10].



$fx1$: remote x-direction force.
$fy1$: remote y-direction force.
$fz1$: remote z-direction force.
$fx2$: local x-direction force.
$fy2$: local y-direction force.
$fz2$: local z-direction force.
SCP: Surface contact point.

Figure 1: Force feedback in collaborative action.

3.1. Network parameters for DHVE traffic

Network QoS performance is generally described using four basic parameters: (i) delay: the difference between the time when the packet has been sent and the time when it is received. (ii) Jitter: the statistical variance of delay measured as the average time between two successively received IP packets. (iii) Packet loss: expressed as a percentage of the number of packets not received, to the number of packets sent. (iv) Throughput: the number of packets that can be transmitted in a fixed amount of time. Each of these imparts specific effects into a user's experience in a DHVE. Delay makes the user's device go through a virtual object before it is felt. This is because the position of the remote virtual objects is delayed making the user push the virtual object with greater force. This degrades the users' perception of "effective collaboration." Delay also desynchronizes the different copies of the virtual environment. Jitter makes the user feel that the virtual object's mass is variable, and can make the system unstable (e.g., it can produce oscillations on the surfaces of objects). Packet loss can reduce the amount of force felt by the user and changes the apparent weight of objects. The HIP is the representation of the haptic device cursor in the virtual environment, and packet loss can also result in loss of contact between the HIP and the virtual object. Packet loss can also cause abrupt force feedback. A minimum throughput is required for successful transmission of haptic traffic in distributed haptic application. Out-of-sequence packets cause abrupt movements (backwards or forwards) in DHVE applications.

Real time transmission with low latency over long distance is the main challenge for networked haptic applications. The aim of network level QoS is to provide stable bandwidth, controlled jitter (i.e., consistent latency) in addition to improved packet loss. The QoS parameter values for haptic traffic are different from traditional real-time (e.g., VOIP) Internet applications; for example, network latency >50 milliseconds can lead to instability in tele-haptic interaction. The network characteristics considered for the DHVE flows are the bandwidth of the connection, the packet delay, packet jitter, and packet loss. Table 1 shows the DHVE haptic traffic network parameters versus other types of network service. It is clear that haptic media is more sensitive to delay and jitter then other traffic types.

4. SYNCHRONIZATION OF POSITION AND FORCE INTERACTION IN HAPTIC COLLABORATIONS

Positional synchronization is a major challenge in distributed shared virtual environments [29]. This becomes even more challenging in peer-to-peer architectures, and any techniques that can reduce the delay and jitter between peers can be expected to improve synchronization and hence the overall system performance. In our peer-to-peer architecture, each peer has their own copy of the virtual environment database. Position synchronization and force collaboration are both implemented using this network architecture. Position synchronization is achieved by transmitting the difference in position, which is calculated from current and previous positions. The difference in position of the local peer is transmitted to the remote peer who adds this difference to its local position in order to achieve position synchronization. When two forces push a virtual object at the same time, their vector sum will decide in which direction the virtual object will move. As shown in Figure 1, the reaction force is computed in proportion to the remote force, the depth of the PHANToM cursor inside the virtual object, and velocity between the cursor and the virtual object. During collaboration, if local and remote forces are applied to the opposite faces of the cube, they cancel each other. In contrast, if local and remote forces are applied to same face of the cube, the resultant force is the summation of the local force and remote force.

4.1. Position synchronization and force interaction algorithm

Figure 2 shows a time event diagram which illustrates how our algorithm achieves position synchronization and force computation. The next position of the virtual object is the summation of the local position and remote position

Table 1: Different traffic types parameters [2, 3, 27, 28].

Traffic	Characteristics	QoS requirements
Haptic	Transmission rate of 1000 packet/sec Constant packet rate Sensitive to jitter and delay	Delay < ~50 ms Throughput ~500 kbps–1 Mbps Jitter < ~2 ms Packet loss < ~10%
Voice	Alternating talk spurts Silence interval Talk-spurts produce constant packet	Delay < ~150 ms Throughput ~22 kbps–200 kbps Jitter < ~30 ms Packet loss < ~1%
Video	Highly bursty traffic Long-range dependencies	Delay < ~400 ms Throughput ~2.5 Mbps–5 Mbps Jitter < ~30 ms Packet loss < ~1%
Data	Poisson type Long-range dependencies	Zero or near-zero packet loss Delay may be important

displacement. When the local PHANToM touches a virtual object, the movement of the local virtual object follows the velocity generated by the local force without adding the remote box position displacement. When the local PHANToM is not touching the virtual object, the total movement of the local virtual object is equal to the summation of local and remote position displacement. In terms of force manipulation, when the PHANToM touches the local object, the total force is the addition of both local and remote forces. In contrast, when the local PHANToM does not touch the local object, the total force at the local site is equal to the remote force only. When there are two forces applied to a single virtual object the resultant force is the vector summation of these forces. Therefore, the movement of the virtual object follows that of the resulting force. The differences in position, force, and time at the local peer are sent to the remote peer and vice versa.

Tests have been conducted to evaluate the performance of this synchronization algorithm. Figure 3 provides some results that can represent the accuracy of the algorithm. The X-position discrepancy of a moving virtual box is obtained by capturing (in real time) the X-position of the virtual cube across two networked peers, computer A and computer B. Figure 3(a) shows the x-position trajectory of the virtual cube. Figure 3(b) shows that there is less than 5mm discrepancy in the X-position of each peer, which is very low. In addition, the coefficient of correlations [30], which shows the covariance of the X-position between computer A and B, is 0.99760. This is to demonstrate if any linear relationship of the two X-position values at computer A and B can be obtained. The correlation is 1 in the case of an increasing linear relationship, −1 in the case of a decreasing linear relationship, and 0 in the case of an independent correlation. A correlation coefficient of 1 indicates a perfect match between the X-position values. Other values between all these cases indicate the degree of linear dependence between the X-position values. Thus, the high value shows that the X-position trajectories of the two networked peers are very closer to each other and hence the algorithm is working well.

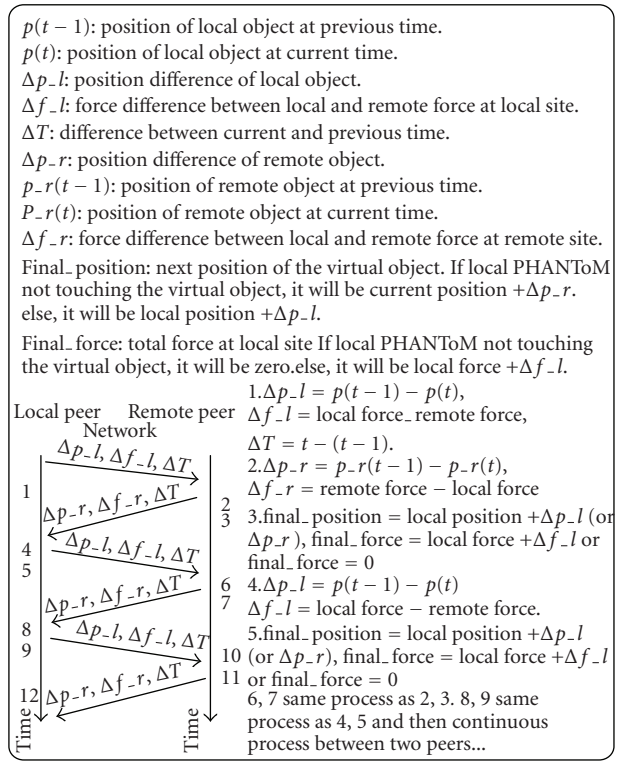


Figure 2: Position, force, and time events on local and remote peers.

5. EXPERIMENTAL AND SIMULATION ARHCITECTURES

Figure 4 shows the approach taken. Haptic traffic from a DHVE application was first captured in an experimental test bed, and the subsequent traffic patterns analyzed.

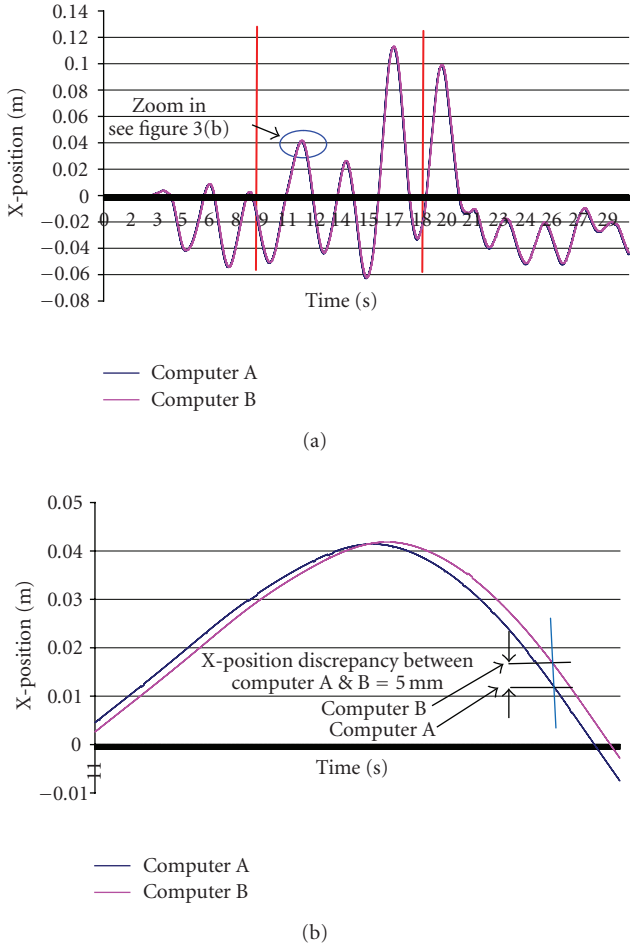


FIGURE 3: (a) X-position discrepancy of the two networked peers A & B with no network impairments. (b) X-position discrepancy of the two network peers A & B, zoom in section of (a).

A custom PDF model was then created for use in the network simulation tool OPNET [3]. A simulation model of DHVE applications running over a network was then developed. The OPNET network model is similar to the experiment test bed. The PDF model is used to generate haptic application traffic to run in the simulated DiffServ network. Subsequently, the effect of running haptic traffic over a DiffServ IP network is obtained. This approach is used to overcome some of the limitations of a test bed. Using this, we are able to simulate a larger scale DHVE environment without the restriction of physical resources. However, the limitation of the simulation model is that we cannot simulate the user's haptic perception which is something that can only be studied in a real-world environment. The experimental architecture applies a QoS mechanism (in the form of Class-based WFQ) which is able to reduce the delay of the haptic traffic and so improve the user's haptic experiences.

5.1. Experimental architecture

The objective of the experimental system is to enable us to generate haptic traffic and study the network QoS characteristics for haptic traffic transmission over a QoS-enabled IP network. We use Matlab Simulink, Real-time Workshop v6.1, Virtual Reality Toolbox v4.0.1, and the proSENSE toolbox from HandshakeVR [16] to develop our experimental system. This experimental architecture is scalable and the current research mainly focuses on 2 users which will be extended to multiusers in the forthcoming work.

5.1.1. Design of experimental system

We have developed an experimental platform based on a peer-to-peer network architecture in order to study network QoS characteristic for haptic traffic. A difference is that in our operation, we transmit the haptic interface point (HIP) position, the virtual objects' positions, timestamp, and the force vectors between the networked peers. In Figure 5, the PCs are running with the VR environment and haptic rendering (e.g., workstation #1 and workstation #2). In this case, the force feedback device used is the PHANToM omni [26] from SensAble Technologies Inc. This is used to manipulate moving virtual objects and to provide the user with force feedback from the virtual environment when the HIP touches a virtual object. The PHANToM Omni generates 1000 packets/s of position and force data during haptic collaboration actions. The workstation connects to the PHANToM Omni through a FireWire interface. Haptic traffic flows between workstations 1 & 2 over the Ethernet network connection. The complete DHVE network architecture which is based on this basic architecture is described in the next section.

5.1.2. Experimental system overview

In Figure 6, four computers are connected through a bottleneck Ethernet link. The gigabit link is running on limited bandwidth of 10Mbps through the two Cisco routers A and B. The experimental hardware is comprised of haptic devices, host and target system hardware, background traffic generator hardware, and network devices. In the test bed, the host and target system is executed in the same PC (i.e., PC 1 and PC 2). A network monitoring tool called "IP Traffic" [31] is used as the traffic generator software as well as being used as the traffic capturing tool.

In operation, PCs 1 and 2 are running DHVE Matlab applications, and PCs 3 and 4 function as background traffic generators for the bottleneck link. The mean background traffic setting is throughput 10137 kbps, packet size 1460 bytes, and with UDP protocol. The buffer sizes (transmitter and receiver) in each interface of the switches, router A and B are zero. The haptic traffic is given various CBWFQ weights in contrast with a constant background traffic weight. Figure 7 shows the Matlab haptic environment which consists of a virtual environment workspace comprising one moving cube, one static cube and two ball spheres which represent local and remote PHANToM cursors (HIPs). The size of the virtual cubes is 4 cm × 4 cm × 4 cm. The workspace boundary is 7 cm on each side. The cubes are modeled to simulate the mass, damping, form, position, velocity, and

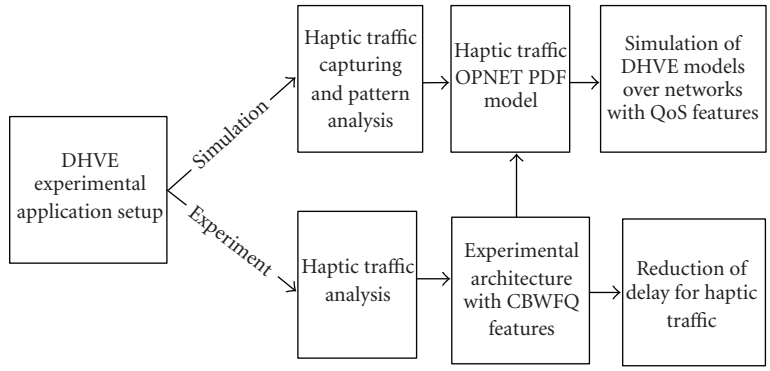


Figure 4: Experiment and simulation approaches.

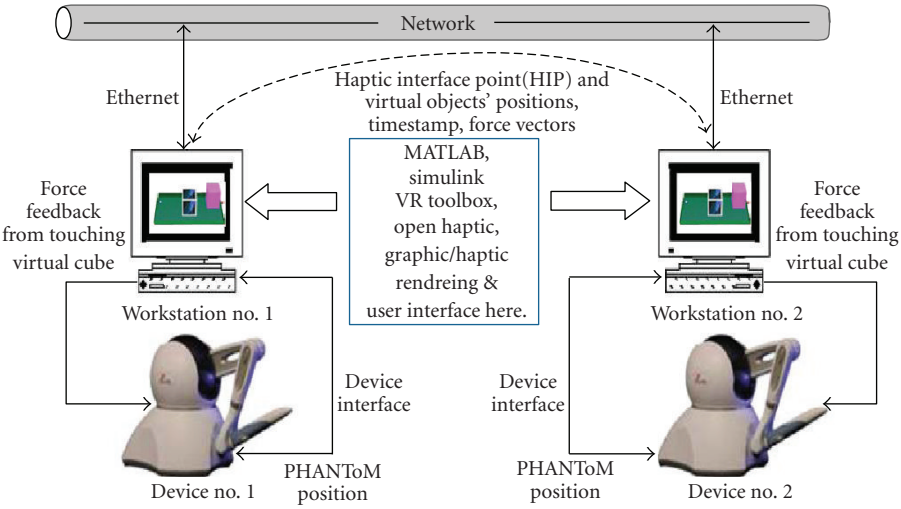


Figure 5: Design of distributed haptic peer-to-peer application.

acceleration of the dynamic virtual objects. Their physical properties are: mass = 5 kg, stiffness = 300 N/m, and damping factor = 7, respectively.

In Figure 8, subjects are able to feel the two virtual cubes but not the work platform; each peer has the same virtual environment as shown in Figure 7. The blue cube in the middle of Figure 7 is movable and whereas the pink cube on the right hand side of the subject is static. Subjects are able to push the moving blue cube by using two PHANToM devices and feel the momentum, force, and velocity of the virtual cube. In addition, they are able to perform collaborative and co-operative tasks on one or both cubes. When running, users at PC 1 and PC 2 push the 3D cubes in the virtual environment, (see Figure 8) force is generated at a PHANToM whenever its HIP touches a virtual cube. This force data together with the HIP and virtual objects' positions are transmitted from PC1 to PC2 and vice versa.

Previous works on distributed haptic environments have concentrated on synchronization of positions (haptic device or virtual objects) [32]. The peer-to-peer architecture presented here further extends this to enable the force interaction between two users. Thus, the force data is sent

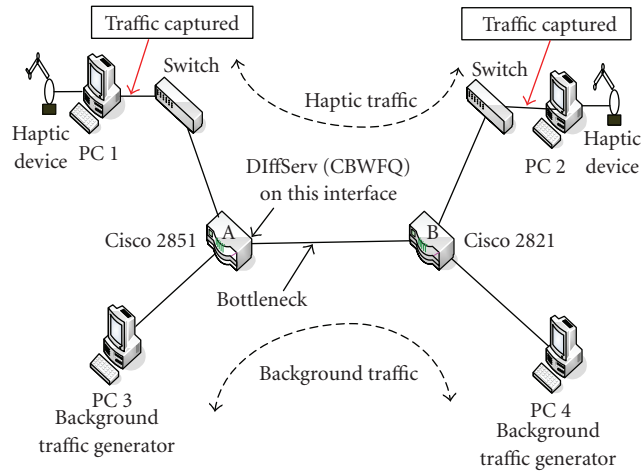


Figure 6: Experimental model of distributed peer-to-peer architecture.

over to remote peer in addition to the position information. From this, the traffic flows were found to require 736 kbps

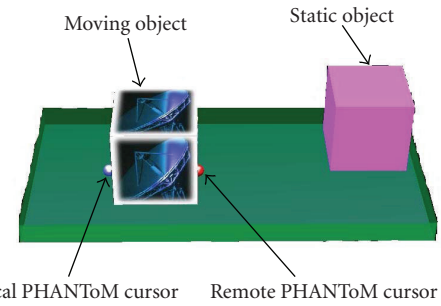


Figure 7: Snapshot of the implemented collaborative haptic virtual environment.

Figure 8: Two users are collaborating a moving cube in the DHVE.

for haptic traffic in each direction. From analysis of the distributions of the traffic delay and bandwidth, subsequent PDF models for use OPNET were developed and used to simulate haptic traffic along with other multimedia traffic sources, in the network simulator.

5.1.3. Haptic traffic queue configurations

In the experimental test bed, DHVE traffic is classified and prioritized in the routers using Class-based weight fair queuing (CBWFQ) from Cisco systems [33]. CBWFQ is a congestion management mechanism that is offered by Cisco for its router platforms and is typical of the QoS mechanisms found in today's routers. It is not available in the OPNET [34] modeling environment, however, it is based on proportionally-fair fluid-flow packet scheduling techniques similar to weighted fair queuing (WFQ), and both CBWFQ and WFQ provide similar functionalities for traffic queuing and bandwidth management. CBWFQ extends WFQ by allowing users to define the classes used in WFQ. The classes can be determined by protocol, access control lists (ACLs), IP precedence, or input interface. Each class can be allocated different bandwidth guarantees in terms of its scheduler queue weight. This approach allows greater control of the haptic traffic when it is received together with other traffic. Figure 9 shows the processing applied to haptic traffic packets at the ingress to a number of interfaces in a router; traffic is forwarded to a specific interface according to its type, where it is subsequently classified and scheduled. The priority queue is served as long as it is nonempty; the CBWFQ queues are then served in proportion to their weights. When CBWFQ queues have consumed any reserved bandwidth or become empty, the best effort queue is then served.

Figure 9 shows the queue setup of haptic and background traffic for the output port (egress port) of Cisco router A in the experimental test bed shown in Figure 6. The haptic traffic class is set with weights of 0, 1, 5, 10, 15, or 30. A Similar setup is applied to the egress interface of Router A in the simulation model shown in Figure 10. Voice and video applications are treated in different classes during WFQ classifications. The background traffic class is set to best effort. In order to improve the haptic traffic transmission under background traffic load, the CBWFQ weight of the haptic traffic class was varied. The haptic class weight was not set higher than 30 because after that the delay was found to be almost zero. This is because the CBWFQ guarantees enough bandwidth (736 kbps in our application) for haptic traffic. The percentage background traffic is calculated with the ratio of 10 Mbps. For example, 10% background traffic will generate 1 Mbps from router A to router B.

5.2. DHVE simulation model

Current network simulators are designed to model existing traffic types such as voice, video, and data traffic, and as such there are no models to represent haptic traffic. As there was no generalized distribution model that is able to represent haptic traffic in OPNET, a custom probability density function (PDF) model was created. Details of this model are presented in [3]. In order to customize a simulation haptic network model, empirical haptic traffic is captured from the test bed, analyzed and then the OPNET PDF model is created. This is then applied as traffic in the network simulation. Figure 10 shows eighteen PCs connected with two switches and routers. The two routers A and B are connected across a 10 Mbps link in order to study the effect of WFQ on haptic traffic. The link creates a bottleneck between the routers; background traffic builds up traffic congestion at router A and thus permits the implementation of WFQ at the egress interface of router A. The other network links are 100 Mbps. The haptic domains 1, 2 are configured to run a custom application task that simulates a DHVE application by using the custom OPNET PDF model. In addition, there are PCs running video, audio, FTP, Email, HTTP, and database applications as multimedia traffic flows. In this case, video and audio have been set with streaming traffic. The system is running with weight fair queuing (WFQ) enabled in the output interface of router A as shown. WFQ dynamically classifies network traffic into individual flows and assign each flow a fair share of the total bandwidth. Unlike priority queuing, each flow is serviced in according to their weight. The weight assigned to haptic traffic is then increased in steps. Additionally, a low-latency queue (LLQ)

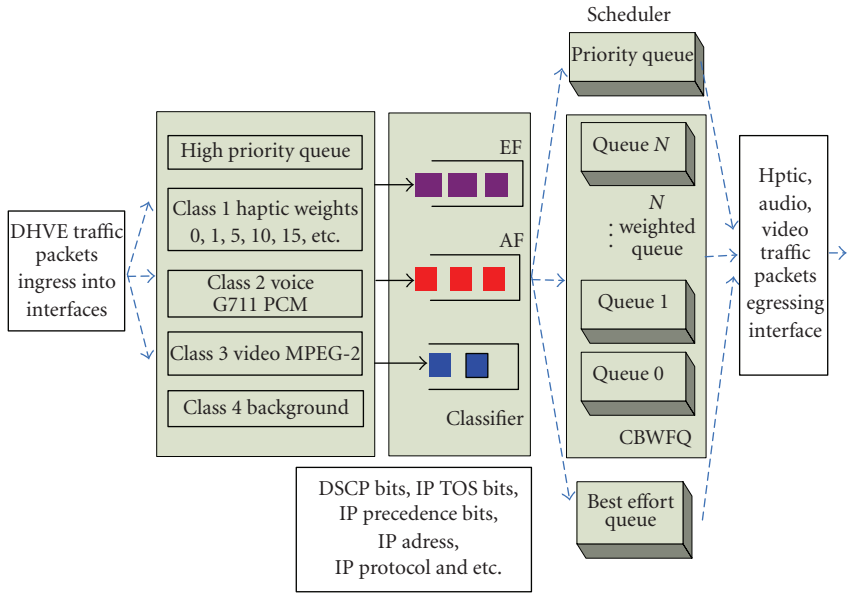


Figure 9: Diffserv treatments for haptic, audio, video and background traffic packets.

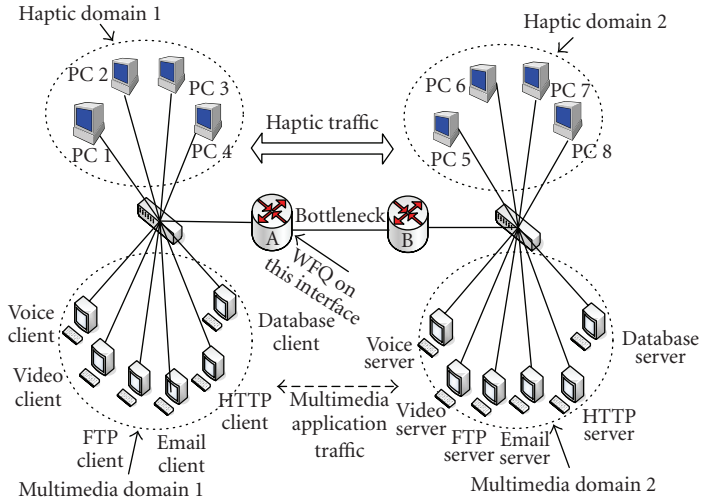


Figure 10: Distributed haptic virtual network simulation model with audio, video, ftp, http, and database applications.

provides a priority queue function which is equivalent to Diffserv's "Expedited forwarding" (EF) queue.

6. EXPERIMENTAL AND SIMULATION RESULTS

6.1. *Experiment results*

As shown in Table 2, the haptic traffic effective throughput is reduced sharply at 97%–99% background traffic load because of traffic congestion starving the bandwidth available for these flows. The effective traffic throughput is also reduced significantly at 95%-96% background load. The effective traffic throughput is maintained at 1000 packets/s at up to 90% background traffic load. In summary, the packet effective throughput from each DHVE machine drops significantly above 90% background load. From the physical experiment, it was observed that at these levels, the user will feel vibration in the PHANToM and also large abrupt force feedback. At this point, the haptic system becomes unstable and the PHANToM is not able to hold stable at position because it keeps vibrating. This highlights that there is a minimum bandwidth required for a DHVE application.

Figure 11 shows the experiment results when haptic traffic is allocated CBWFQ bandwidth weights of 1, 5, 10, and 30. The result shows that when the haptic traffic is given higher bandwidth, the packet transit delay is reduced. In Figure 11, haptic traffic end-to-end delay increases to 200 milliseconds whenever background traffic increases and the haptic traffic is under best-effort treatment. This means that the router A in Figure 6 has not been set with any QoS mechanisms. When CBWFQ is employed, the delay of haptic traffic is reduced from 200 milliseconds (best effort),

Table 2: Haptic traffic with background load.

Background loading (%)	Packet rate (packet/s)	
60	981.957045	
70	991.862847	
80	999.131579	
90	987.389474	
95	793.389474	User feels abrupt force feedback, haptic system becomes unstable
96	509.647368	
97	77.894737	
98	37.368421	
99	60	

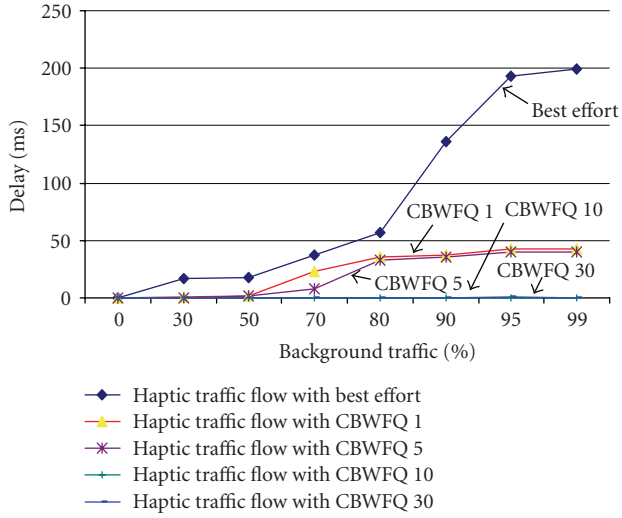


Figure 11: Haptic traffic end-to-end delay versus background load with different WFQ weights at router A.

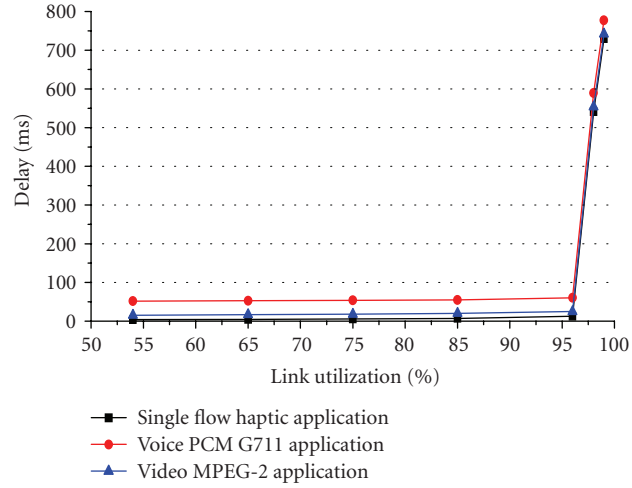


Figure 12: End-to-end traffic delay versus link utilization.

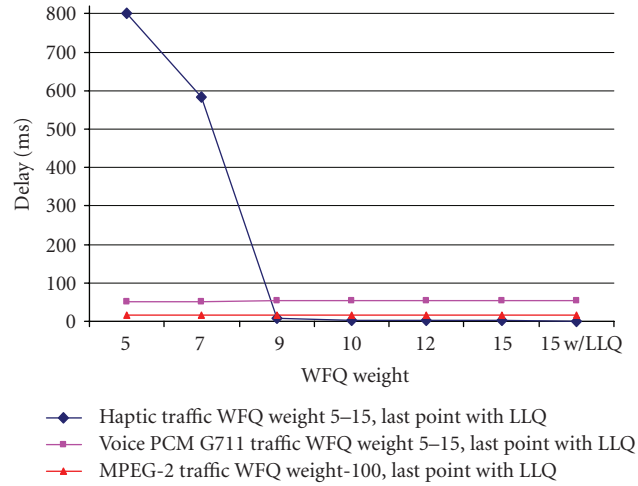


Figure 13: End-to-end delay of haptic, voice and video traffic with different WFQ weights.

to less than 1 millisecond (CBWFQ weight = 30) under a background load of 95% load. Setting the CBWFQ haptic weight = 1 with a guaranteed bandwidth of 1 Mbps results in a significant improvement over best effect, and setting CBWFQ haptic weights of 10 and 30 can definitely reduce the delay further as shown.

6.2. Simulation results

This section investigates the haptic traffic characteristic with WFQ-enabled on output interface of router A in Figure 10. Figure 12 shows end-to-end delay of individual haptic, voice and video traffic flows with 45% background traffic loading on the bottleneck link (10 Mbps). The combined flows increase the bottleneck link utilization up to 98%. Figure 12 shows that with a best-effort only service, haptic traffic incurs nearly 730 milliseconds of end-to-end delay which is totally unacceptable for a haptic operation. The simulation results shown in Figure 13 are the end-to-end delay of the haptic, voice, and video traffic flows with different WFQ weights. The results are obtained by varying the weights for all other traffic (except haptic traffic) flowing through the output interface of router A. The audio and video traffic flows have been set to achieve end-to-end delays of below 100 milliseconds (which is reasonable for audio and video streaming applications). The WFQ weight ranges from best effort (WFQ = 0) to WFQ weight = 15. Initially, the best effort IP network caused end-to-end delay of 800 milliseconds in the haptic traffic; however, this delay is improved by introducing prioritised service class for haptic traffic. It can be observed that the end-to-end delay of the haptic traffic has decreased from 800 milliseconds (WFQ weight = 2) to 1.14 milliseconds (WFQ weight = 15). The delay is further reduced to 0.7 milliseconds with the low-latency queue (LLQ) enabled on the interface. This result shows that the introduction of WFQ improves the QoS provided to the haptic traffic.

Figure 14 shows the throughput of haptic traffic at the Ethernet layer; it highlights the reduction in throughput when WFQ weight <9. The result is obtained by setting

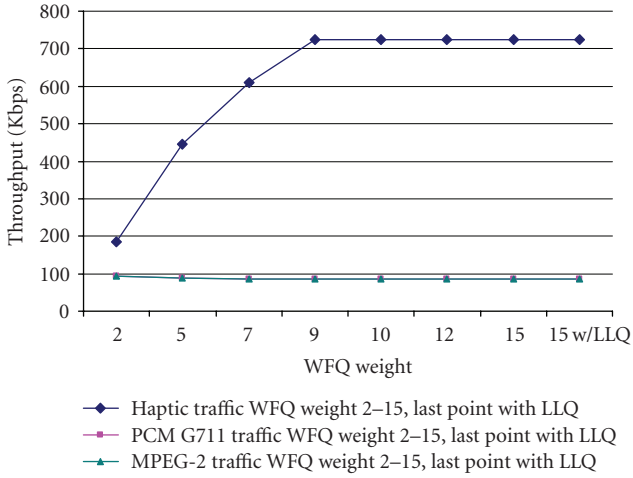


Figure 14: Haptic, voice, and video traffic throughput with different WFQ weights.

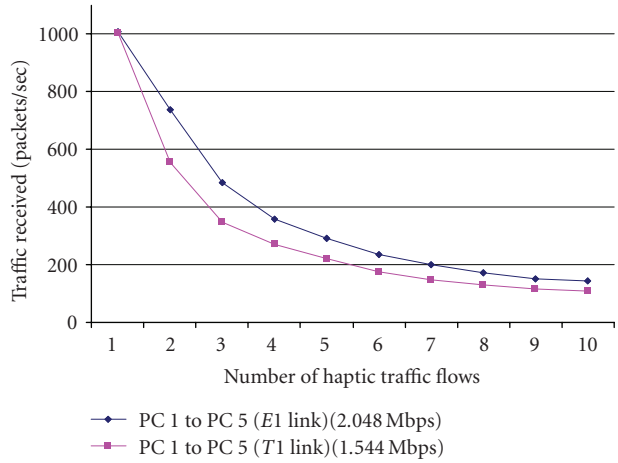


Figure 15: Received traffic rate of E1, T1 Link with multiple haptic flows.

different WFQ weights for the haptic, voice, and video traffic. The haptic packets are 64 bytes, which become 92 bytes at Ethernet layer. Therefore, the total throughput at the Ethernet layer is $92*1000*8 = 736$ Kbps. This is closely matched to the values in Figure 14.

Figure 15 shows the relationship between the rate of the haptic traffic received whenever the rate of the bottleneck link is reduced from 10 Mbps to E1(2.048 Mbps) and/or T1(1.544 Mbps). This is shown for multiple haptic network flows (up to 10 flows). As shown, the rate of received haptic traffic is reduced sharply when there are three flows or more, because of traffic congestion starving the bandwidth available for these flows. It is important for a remote haptic receiver to receive around 1000 packets/s in order for the haptic application to maintain a constant network throughput of 850 Kbps. This in turn helps maintain the local haptic feedback control loop and so eliminate instability. The T1 link exhibits poorer performance than the E1 link simply because an E1 link has higher bandwidth capacity than a T1 link. In summary, the packet rate received from each DHVE machine drops significantly when there are two haptic flows. From the physical experiment, it was observed that at these levels, the user would feel vibration in the PHANToM and also large abrupt force feedback.

6.3. Discussion of simulation and experiment results

Sections 6.1 and 6.2 have presented the experimental and simulation results, respectively. While both approaches have been shown to yield comparable results, there are some discrepancies. The end-to-end delay of the simulated haptic traffic decreases to about 2 milliseconds when the WFQ weight is 9. The test bed result shows that the end-to-end delay drops to 1 millisecond when CBWFQ weight = 10. This is because CBWFQ can allocate a minimum amount of bandwidth in which haptic traffic has exclusive use. Thus, the simulation model is comparable with the experiment test bed although it contains more traffic sources than the experiment test bed. The haptic traffic is therefore able to improve its transmission quality if given a minimum amount of network bandwidth. This is shown in previous sections for both experiment and simulation results. From the test bed experiments, the user haptic perception is improved when CBWFQ is enabled in the network, as compared to a best-effort-only service.

We have also studied the consequence of using DSCP for haptic traffic. This is shown in Table 3. The haptic traffic is studied for maximum end-to-end delay under different AF and EF of DSCP Markings. EF with low-latency queue (LLQ) provides highest priority thus yielding lowest delay. Table 3 shows that the AF21-AF23, AF31-AF33, AF41-AF43, and EF have lower end-to-end delays compared to AF11-AF13. Therefore, AF11-AF13 are not recommended for transmission of haptic, audio or video traffic. This is in agreement with IETF recommendations (RFC4594) for the transport of video and voice traffic [6]. The maximum end-to-end delay also depends on the type of link used and the traffic loading. In this case, we used haptic traffic, real time audio and video streaming traffic plus the other multimedia traffic in our simulation model. A similar result was presented in our paper in Immerscom 2007 [25] in which haptic traffic was investigated with T1 (1.544 Mbps) and T3 (44.736 Mbps) link. BE and AF11-AF13 are not recommended for transmission of haptic traffic. This result confirms the recommendations are also valid for voice and video traffic.

The results in Figure 14 showed that the haptic traffic for our DHVE application has a throughput of 736 kbps. Therefore, it is important to reserve this minimum bandwidth in order for the haptic traffic to be effectively transmitted. This is comparable to the results obtained from the test bed which shows that a CBWFQ of 1 will guarantee more than enough bandwidth (736 kbps in our application) for haptic traffic and hence also reduce the delay of haptic traffic as is shown in Figure 11. The results confirm that haptic traffic is comparable to telephony or video classes but it is very

Table 3: . Maximum end-to-end delay for haptic, voice, and video traffic using different DiffServ Code Point AF and EF marking.[1]

DSCP	End-to-end delay (millisecond)		
	Haptic	Voice	Video
BE	1206.8306	1254.7928	1023.2244
AF11	217.8382	284.8830	235.8420
AF12	217.8382	284.8330	235.8420
AF13	217.8382	284.8330	235.8420
AF21	1.7631	49.4548	6.5655
AF22	1.7631	49.4548	6.5655
AF23	1.7631	49.4548	6.5655
AF31	1.6851	49.3732	5.8811
AF32	1.6851	49.3732	5.8811
AF33	1.6851	49.3732	5.8811
AF41	1.6382	49.3348	5.5212
AF42	1.6382	49.3348	5.5212
AF43	1.6382	49.3348	5.5212
EF	1.5952	49.2930	4.9335
EFLLQ	1.5868	49.2871	4.8084

[1] BE—best effort, AF—assured forward, EF—expedited forward, LLQ—low-latency queue, link T3—44.736 Mbps, 95% link utilisation.

Table 4: Proposed haptic class with DSCP marking scheme in addition to DiffServ service classes and DSCP marking scheme in [28].

Service class	Traffic characteristics	Tolerance to			Protocol	DSCP
		Loss	Delay	Jitter		
Haptic	Fixed packets, real-time, inelastic and constant rate flows	Very low	Very low	Extreme low	UDP	EF
Telephony	Fixed size small packets, inelastic and low-rate flows	Very low	Very low	Very low	UDP	EF
Multimedia streaming	Variable size packets, elastic with variable rate	Low-meduim	Meduim	Yes	UDP	AF31 Af32 AF33
Low-priority data	Nonreal time and elastic	High	High	Yes	N/A	BE

sensitive to jitter [8, 11]. Based on our findings, we proposed a DSCP marking scheme for haptic traffic. The requirement for using haptic traffic in a managed network by the network administrator is proposed in Table 4. The haptic class is proposed to have a DSCP marking of EF or at least AF21 and above.

7. CONCLUSIONS AND FUTURE WORK

This paper presents a study into the provision of QoS for (DHVEs) whenever they are provided over QoS-enabled packet switched networks such as the next generation Internet. Moreover the study is particularly relevant to DVHEs that are implemented as networked peers rather than traditional client-server architectures. A new peer-to-peer DHVE architecture that permits peers distributed across an IP network to perform collaborative and co-operative haptic tasks on virtual objects is presented. The provision of QoS for these types of applications is then investigated. The approach taken employs an experimental test bed network to gather empirical data concerning the statistical distribution of haptic traffic generated by the networked peers. This is then used to generate a traffic model for haptic traffic which is used in a network simulation to analyze the performance of DHVE traffic flows across networks that are QoS-enabled. Haptic traffic is simulated along with voice (G711), video (MPEG-2) and other multimedia traffic. Suitable values for the network-level parameters for haptic traffic are then developed and recommendations are proposed to provide QoS for multimodal traffic flows. The work involves studies of haptic traffic under a best-effort IP network and a DiffServ IP network.

The results show that the network simulation model compares favourably with the physical network, and can be used to generate a scalable haptic network model where multiple connections carrying haptic traffic may be examined. Both approaches show that reducing network delay and jitter by providing "better-than-best-effort" service

which uses specific QoS classes for haptic traffic can lead to improvements in users' haptic experiences with distributed applications such as virtual environments.

The simulation results show that haptic throughput increases correspondingly to an increase in the queue scheduling weight. In the experimental test bed, the end-to-end delay experienced by haptic traffic is found to decrease from 200 milliseconds (best effort) to 40 milliseconds (CBWFQ) by running the haptic application in a DiffServ network. Both simulation and experimental results show that transmission of haptic traffic is improved with implementation of a traffic classification and prioritization mechanism (WFQ and CBWFQ, resp.). The simulation model can be used to simulate large numbers of haptic traffic flows. The results from this lead to the conclusion that WFQ and CBWFQ in DiffServ packet switched network improve network performance for transporting haptic traffic by proper setting of the DiffServ DSCPs and the packet schedulers in the routers. Subsequently, a haptic traffic class with DSCP marking scheme is proposed. This can be used as a reference for configuring a QoS-enabled network to support DHVE applications or multimodal traffic flows.

In the future, we intend to investigate haptic user perception tests on the possibility of a DiffServ-enabled IP QoS network that allows consistency force and position collaboration among multiple (>2) users. In addition, we will study the application of weight random early detection (WRED) and interleaving, which are specifically configured to improve haptic traffic under congestion conditions that may result in bursty packet loss in a network.

ACKNOWLEDGMENT

This work was supported by the Industrial Research and Technology Unit Northern Ireland SPUR scheme at Queen's University Belfast.

REFERENCES

[1] K. Jeffay, T. Hudson, and M. Parris, "Beyond audio and video: multimedia networking support for distributed, immersive virtual environments," in *Proceedings of the 27th Euromicro Conference (EUROMICRO '01)*, pp. 300–307, Warsaw, Poland, September 2001.

[2] R. Souayed, D. Gaiti, W. Yu, G. Dodds, and A. Marshall, "Experimental study of haptic interaction in distributed virtual environments," in *Proceedings of Eurohapics*, pp. 260–266, Springer, Munich, Germany, June 2004.

[3] K. M. Yap, A. Marshall, W. Yu, G. Dodds, Q. Gu, and R. T. Souayed, "Characterising distributed haptic virtual environment network traffic flows," in *Proceedings of the 4th IFIP Conference on Network Control and Engineering for QoS, Security and Mobility*, pp. 297–310, Lannion, France, November 2007.

[4] P. Buttolo, R. Oboe, and B. Hannaford, "Architectures for shared haptic virtual environments," *Computers & Graphics*, vol. 21, no. 4, pp. 421–429, 1997.

[5] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, and W. Weiss, "An architecture for differentiated services," RFC 2475, December 1998.

[6] J. Babiarz, K. Chan, and F. Baker, "Configuration guidelines for diffServ service classes," RFC 4594, August 2006, http://www.ietf.org/rfc/rfc4594.txt.

[7] R. Braden, D. Clark, and S. Shenker, "Integrated services in the internet architecture: an overview," RFC 1633, Network Information Center, SRI International, Menlo Park, Calif, USA, June 1994.

[8] S. Matsumoto, I. Fukuda, H. Morino, K. Hikichi, K. Sezaki, and Y. Yasuda, "The influences of network issues on haptic collaboration in shared virtual environments," in *Proceedings of the 5th Phantom Users' Group Workshop (PUG '00)*, pp. 22–24, Aspen, Colo, USA, October 2000.

[9] K. Hikichi, I. Arimoto, H. Morino, K. Sezaki, and Y. Yasuda, "The evaluation of adaptation control for haptics collaboration over the internet," in *Proceedings of International CQR Workshop*, Okinawa, Japan, May 2002.

[10] K. M. Yap, A. Marshall, and W. Yu, "Supporting haptic collaboration across networked peers with real time force feedback in distributed virtual environments," in *Proceedings of the 14th IEEE International Conference on Telecommunications (ICT '07)*, pp. 494–499, Penang, Malaysia, May 2007.

[11] F. Naït-Abdesselam and N. Agoulmine, "QoS control and adaptation in distributed multimedia systems," in *Proceedings of the 11th International Parallel Processing Symposium in Conjunction with the 13th International Parallel Processing Symposium and the 10th Symposium on Parallel and Distributed Processing (IPPS '97)*, pp. 375–383, Springer, San Juan, Puerto Rico, USA, April 1999.

[12] J. C. de Oliveira, S. Shirmohammadi, and N. D. Georganas, "Collaborative Virtual Environment standards: a performance evaluation," in *Proceedings of the 3rd IEEE International Workshop on Distributed Interactive Simulation and Real-Time Applications*, pp. 14–21, Greenbelt, Md, USA, October 1999.

[13] M. Eraslan, N. D. Georganas, J. R. Gallardo, and D. Makrakis, "A scalable network architecture for distributed virtual environments with dynamic QoS over IPv6," in *Proceedings of the 8th IEEE International Symposium on Computers and Communications (ISCC '03)*, vol. 1, pp. 10–15, Kemer-Antalya, Turkey, June-July 2003.

[14] H. Yu, Q. Zhou, D. Makrakis, N. D. Georganas, and E. Petriu, "Quality of service support of distributed interactive virtual environment applications in IP networks," in *Proceedings of IEEE Pacific RIM Conference on Communications, Computers, and Signal Processing (PACRIM '01)*, vol. 1, pp. 196–199, Victoria, Canada, August 2001.

[15] R. S. Allison, J. E. Zacher, D. Wang, and J. Shu, "Effects of network delay on a collaborative motor task with telehaptic and televisual feedback," in *Proceedings of the ACM SIGGRAPH International Conference on Virtual Reality Continuum and Its Applications in Industry (VRCAI '04)*, pp. 375–381, Singapore, June 2004.

[16] HandshakeVR,TiDeC™ is trademark of HandshakeVR, http://www.handshakeVR.com/.

[17] K. M. Yap, A. Marshall, and W. Yu, "Supporting haptic collaboration across networked peers with real time force feedback in distributed virtual environments," in *Proceedings of the 14th IEEE International Conference on Telecommunications (ICT '07)*, pp. 494–499, Penang, Malaysia, May 2007.

[18] R. M. Traylor, D. Wilhelm, B. D. Adelstein, and H. Z. Tan, "Design considerations for stand-alone haptic interfaces communicating via UDP protocol," in *Proceedings of the 1st Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems (WHC '05)*, pp. 563–564, Pisa, Italy, March 2005.

[19] C. Jay, M. Glencross, and R. Hubbold, "Modeling the effects of delayed haptic and visual feedback in a collaborative virtual environment," *ACM Transactions on Computer-Human Interaction*, vol. 14, no. 2, article 8, pp. 1–31, 2007.

[20] P. Dev, D. Harris, D. Gutierrez, A. Shah, and S. Senger, "End-to-end performance measurement of internet based medical applications," in *Proceedings of the Annual Symposium of the American Medical Informatics Association*, pp. 205–209, San Antonio, Tex, USA, November 2002.

[21] T. Hudson, M. C. Weigle, K. Jeffay, and R. M. Taylor II, "Experiments in best-effort multimedia networking for a distributed virtual environment," in *Multimedia Computing and Networking*, vol. 4312 of *Proceedings of SPIE*, pp. 88–98, San Jose, Calif, USA, January 2001.

[22] H. Nishino, S. Yamabiraki, Y.-M. Kwon, Y.-M. Okada, and K. Utsumiya, "A distributed virtual reality framework for Korea-Japan high-speed network testbed," in *Proceedings of the 20th International Conference on Advanced Information Networking and Applications (AINA '06)*, vol. 1, pp. 433–438, Vienna, Austria, April 2006.

[23] J. Cheong, S.-I. Niculescuy, and M. A. Srinivasan, "Motion synchronization in virtual environments with shared haptics and large time delays," in *Proceedings of the 1st Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems (WHC '05)*, pp. 277–282, Pisa, Italy, March 2005.

[24] S. Lee and J. Kim, "Intra-media synchronization scheme for haptic interactions in distributed virtual environments," in *Multimedia Systems and Applications VIII*, vol. 6015 of *Proceedings of SPIE*, pp. 1–12, Boston, Mass, USA, October 2005.

[25] K. M. Yap, A. Marshall, and W. Yu, "Providing QoS for distributed haptic virtual environments in IP networks," in *Proceedings of the ICST/ACM 1st International Conference on Immersive Telecommunications (Immerscom '07)*, Bussolengo, Italy, October 2007.

[26] SenSAble Technologies, http://www.sensable.com/.

[27] D. P. Bertsekas, "Traffic behavior and queuing in a QoS environment," in *OPNETWORK*, Washington, DC, USA, August 2005.

[28] J. Babiarz, K. Chan, and F. Baker, "Configuration guidelines for diffServ service classes. draft-baker-diffserv-basic-classes-03," July 2004, http://www.tools.ietf.org/html/draft-baker-diffserv-basic-classes-03.

[29] Y. Ishibashi, K. Tomaru, S. Tasaka, and K. Inazumi, "Group synchronization in networked virtual environments," in *Proceedings of the IEEE International Conference on Communications (ICC '03)*, vol. 2, pp. 885–890, Anchorage, Alaska, USA, May 2003.

[30] http://www.en.wikipedia.org/wiki/Correlation, January 2007.

[31] IP Traffic Test Software, ZTI Telecom, http://www.zti-telecom.com/pages/main-ip.htm.

[32] R. Iglesias, S. Casado, T. Gutiérrez, A. García-Alonso, W. Yu, and A. Marshall, "Simultaneous remote haptic collaboration for assembling tasks," *Multimedia Systems*, vol. 13, no. 4, pp. 263–274, 2008.

[33] Cisco Systems, http://www.cisco.com/.

[34] OPNET 11.0A, Product Documentation, Methodologies and Case Studies, http://www.opnet.com/.

Hindawi Publishing Corporation
Advances in Multimedia
Volume 2008, Article ID 615186, 17 pages
doi:10.1155/2008/615186

Research Article

RICE: A Reliable and Efficient Remote Instrumentation Collaboration Environment

Prasad Calyam, Abdul Kalash, Ramya Gopalan, Sowmya Gopalan, and Ashok Krishnamurthy

Cyberinfrastructure and Software Development Group, Ohio Supercomputer Center, 1224 Kinnear Road, Columbus, OH 43212, USA

Correspondence should be addressed to Prasad Calyam, pcalyam@osc.edu

Received 9 March 2008; Accepted 3 November 2008

Recommended by Ghassan Alregib

Remote access of scientific instruments over the Internet (i.e., remote instrumentation) demand high-resolution (2D and 3D) video image transfers with simultaneous real-time mouse and keyboard controls. Consequently, user quality of experience (QoE) is highly sensitive to network bottlenecks. Further, improper user control while reacting to impaired video caused due to network bottlenecks could result in physical damages to the expensive instrument equipment. Hence, it is vital to understand the interplay between (a) user keyboard/mouse actions toward the instrument, and (b) corresponding network reactions for transfer of instrument video images toward the user. In this paper, we first present an analytical model for characterizing user and network interplay during remote instrumentation sessions in terms of demand and supply interplay principles of traditional economics. Next, we describe the trends of the model parameters using subjective and objective measurements obtained from QoE experiments. Thereafter, we describe our Remote Instrumentation Collaboration Environment (RICE) software that leverages our experiences from the user and network interplay studies, and has functionalities that facilitate reliable and efficient remote instrumentation such as (a) network health awareness to detect network bottleneck periods, and (b) collaboration tools for multiple participants to interact during research and training sessions.

1. INTRODUCTION

Increased access to high-speed networks has made remote access of computer-controlled scientific instruments such as microscopes, spectrometers, and telescopes widely-feasible over the Internet. Some of these instruments are extremely expensive and could be worth several hundred-thousand dollars. Hence, a major benefit of remote instrumentation is that it allows remote users to utilize these instruments when they are not in use by local users. In addition, routine maintenance and operation of the instruments require significant investment in staffing. Thus, instrument labs can charge remote access on an hourly usage basis to obtain a better return-on-investment on the instruments. Further, remote instrumentation avoids duplication of investment in instrument labs for funding agencies. In fact, the National Science Foundation is mandating remote instrumentation to be available with all their funded instruments [1]. Besides the above advantages, remote instrumentation fosters education and hands-on training of instruments as well as collaboration for remote users. The collaboration enables multiple remote researchers, each with unique expertise, to jointly analyze samples such as metals, proteins, and tissues. All of the above advantages, especially for training and collaboration, drastically shorten the development process involved in innovations related to materials modeling, biological specimens' analysis for cancer research, and so forth. At the same time, they improve user convenience and significantly reduce research and training costs.

Although there are several advantages, remote instrumentation is demanding in terms of network resource consumption. This is because remote instrumentation sessions involve high-resolution (2D and 3D) video image transfers with simultaneous real-time mouse and keyboard controls. If appropriate network bandwidth is not allocated, network congestion occurs that can impact user *quality of experience* (QoE). In addition, user QoE is affected by network fault events such as optical fiber cuts, route asymmetry, and route flapping that degrade network performance. The user

QoE affected by such network bottlenecks is measured by obtaining subjective opinions of user satisfaction after completion of a remote instrumentation session. The network bottleneck could cause impaired video images at the user, which in turn could lead to improper user control of the microscope's mechanical moving parts. Such improper user control may ultimately result in physical equipment damages that are prohibitively expensive to fix. Hence, it is vital to understand the interplay between (a) the user keyboard/mouse actions toward the instrument, and (b) the corresponding network reactions for transport of instrument video images toward the user, for reliably supporting remote instrumentation.

Assuming that a sample has been shipped to an instrument lab and has been loaded into an instrument, there are two basic use-cases of remote instrumentation. The first use-case is called *remote observation*, where a remote user or multiple remote users only view real-time (2D and 3D) instrument video images. The remote user(s) direct an operator physically present at the instrument to perform all the control actions over a telephone or VoIP call. The second use-case is called *remote operation*, where a remote user or multiple remote users view the instrument video images and also control the instrument in real-time. The first use-case is preferred in cases where the remote users are not familiar with the instrument functionalities. It is also preferred if the intermediate network path between the user and the instruments has bottlenecks. The second use-case is preferred for both local and remote users in cases where human presence around the sample could cause undesirable effects. For example in microscopy involving electron microscopes, human presence increases ambient temperature, which alters properties of materials being analyzed at subangstrom levels on the microscope. Nevertheless, both use-cases require collaboration tools that support voice communications (i.e., *VoIP*) and instant messaging (i.e., *chat*) for communicating efficiently during remote instrumentation sessions. For the multiuser case, collaboration tools are required to (i) show who is controlling/viewing the session (i.e., *presence*) and (ii) manage control privilege amongst the users (i.e., *control-lock passing*) such that at any given instant, only one user controls the instrument.

There are two major parts to this paper. In the first part, we study the complex interplay characteristics between the user and the network during remote instrumentation sessions. As an exemplar for the interplay characterization, we focus on the remote access of electron microscopes (i.e., remote microscopy). However, our work is equally relevant for other computer-controlled scientific instruments. We first present an analytical model for characterizing *user* and *network* interplay during remote microscopy sessions in terms of *demand* and *supply* interplay principles of economics, respectively. The various remote microscopy system states affected by transient network conditions are also modeled. To obtain the trends of the session model parameters, we set up a remote microscopy testbed in cooperation with The Ohio State University's Center for Accelerated Maturation of Materials (CAMM). On this testbed, we use a novel methodology to perform QoE experiments involving actual novice/expert users for a variety of network conditions in LAN/WAN connections. We also present the analysis of the subjective and objective measurements obtained from the experiments. Our analysis provides insights about how network health impacts user behavior and ultimately user QoE.

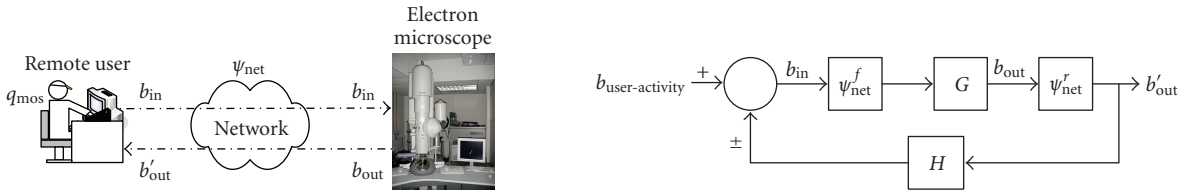
In the second part, we describe our *Remote Instrumentation Collaboration Environment* (RICE) software that leverages our user and network interplay study findings to (a) cope with network bottlenecks, and (b) cater to the multiuser requirements of remote observation and remote operation. In this context, we describe the RICE software functionalities that improve reliability and efficiency of multiuser remote instrumentation sessions that traditionally relied upon off-the-shelf *virtual network computing* (VNC) solutions [2]. The functionalities to improve reliability include real-time network health monitoring coupled with network performance anomaly detection using a "plateau-detector algorithm." This algorithm warns and blocks user's control actions during network congestion periods. We also describe our "session-signaling protocol" used in the RICE tools that improve efficiency of multiuser collaboration. The collaboration tools include VoIP, chat, presence, and control-lock passing that are absent in off-the-shelf VNC solutions. Finally, we present potential applications of RICE for research and training purposes that require multiuser remote instrumentation capabilities.

The remainder of the paper is organized as follows. Section 2 presents related work. Section 3 presents the analytical model for user and network interplay characterization. Section 4 describes the remote microscopy testbed and results of the QoE experiments performed on the testbed. Section 5 describes the RICE software features and its applications for research and training. Section 6 concludes the paper.

2. RELATED WORK

There are several efforts in the United States that are aimed at serving the remote instrumentation needs of researchers and students. Gemini Observatory [3] is an initiative that uses Internet2 to allow remote users to manipulate their twin telescopes. NanoManipulator [4] is another initiative that uses Internet2 to allow remote control and visualization of images from their scanning probe microscopes. Similar remote instrumentation efforts are being supported in other countries also. A notable effort is being led by the National Institute of Materials Science in Japan, where remote instrumentation is being made available to the public and high school education programs [5]. As a part of this effort, remote observation of insects, plants, IC devices, and metals that have been preloaded in a remote-site's scanning electron microscope is being enabled at the National Museum of Emerging Science and Innovation in Tokyo.

One of the early works that developed novel applications for remote operation were done at the Massachusetts Institute of Technology [6]. They developed a custom software for remote control of a Zeiss microscope using a graphical interface running on a workstation computer. The software



(a) Session parameters

(b) Closed-loop control system representation

Figure 1: Basic remote microscopy system.

also allowed several remote users to simultaneously view the microscope in a conference inspection mode, enabling collaboration amongst remote users. Lawrence Berkeley National Laboratory also has developed a custom software application to control their Kratos 1500 keV microscope during in situ experiments [7]. The controls include adjusting external stimuli, adjustment of specimen position and orientation, and manipulation of microscope controls such as illumination, magnification, and focus. To cope with network bottlenecks, they developed schemes to locally automate stage control and microscope focus. Their application has been tested on the Internet along several paths including paths to Berkeley from Washington D.C. and Kansas City. Several other studies have also evaluated performance of remote instrumentation using custom software over the Internet. For example, Research Center for Ultra High Voltage Electron Microscopy (UHVEM) at Osaka University collaborated with National Center for Microscopy and Imaging Research (NCMIR) at University of California San Diego to conduct remote instrumentation experiments on their 3-million volt transmission electron microscope over intercontinental links [8]. The custom software developed by NCMIR has evolved over the years to keep up with the developments of networks, operating systems, and application development tools. The latest variants of their software feature platform-independent Java-based applications for remote instrumentation of several different instruments. These applications have also been integrated into web-services and middleware frameworks [9] that couple remote instrumentation with data and computation services.

Recently, several off-the-shelf remote access solutions have emerged that are either software-based or hardware-based. The most commonly used solution is the software-based virtual network computing (VNC) solution [2] that has several variants such as UltraVNC [10] and RealVNC [11]. It requires preinstalled software at both the instrument and user ends. Alternately, there are hardware-based VNC solutions that are also referred to as *Keyboard, Video and Mouse over IP* (KVMoIP) solutions developed by vendors such as ThinkLogical [12] and Avocent [13]. These solutions use custom hardware and require a pair of encoder and decoder appliances to be installed at the instrument and user ends. Recently, hybrid VNC solutions have also been developed by vendors such as Adder [14] that requires a hardware appliance at the instrument end, and a software client at the user end. Several instrument labs use such off-the-shelf solutions. For example, Oak Ridge National Laboratory uses off-the-shelf VNC solutions for remote control of their High Flux isotope Reactor [15]. Similarly, the California State Polytechnic University also uses off-the-shelf VNC solutions in their Ocean Engineering Program [16].

VNC solutions use raw or copy-rectangle or JPEG/MPEG encoding for video image transfers and TCP for keyboard and mouse control traffic. For sending the video image transfers, VNC uses a Remote Frame Buffer (RFB) protocol that supports various pixel formats such as ZRLE, Zlib, Raw, and Hextile. The pixel updates using the RFB protocol are demand-driven because pixel updates are sent (a) to respond to an explicit TCP-based request from a client, and (b) to update the client's display when there are changes at the server's display. The compression latency of VNC is dependent on factors such as the network health, as well as the client/server CPU speed, other-application task loads, and video card capabilities.

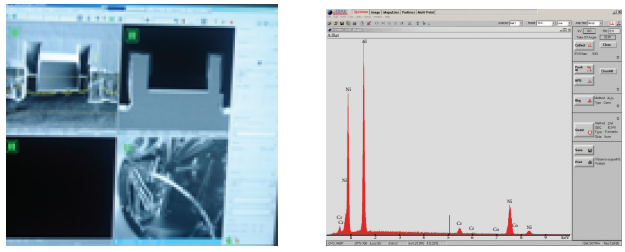
Given the free availability of software-based VNC solutions, QoE evaluations for these solutions can be extensively found in the remote instrumentation literature. However, to the best of our knowledge, there is no literature on systematic QoE evaluations for KVMoIP-based remote instrumentation. The QoE evaluation results presented in Section 4 of this paper focus on the KVMoIP VNC solution on both LAN and WAN paths. We also believe that our work is the first to present an analytical model and characterize user and network interplay in remote instrumentation sessions. Our RICE software presented in Section 5 is based on the UltraVNC solution but has several enhancements targeted for network-aware and collaborative remote instrumentation sessions. Our work in this paper is part of The Ohio State University's CAMM VIM program [17]. This program uses Ohio Supercomputer Center's (OSC) regional network (i.e., OSCnet) to allow remote industry such as Timken and defense labs such as AFRL to access their collection of the world's most powerful scanning/transmission electron microscopes.

3. REMOTE MICROSCOPY SESSION MODEL

In this section, we first describe the parameters involved in a typical remote microscopy session. Next, we model their interactions in different system states borrowing the supply and demand terminology from economics.

3.1. System description

Figures 1(a) and 1(b) show a basic remote microscopy system and its closed-loop control system representation with



(a) Image with live video feeds (b) Image with text and graphs

Figure 2: Comparison of video activity levels in instrument image transfers.

the different session parameters, respectively. The remote user physically controls the functions of the microscope by interacting with a graphical user interface (GUI) application using keystrokes and mouse moves/clicks via VNC or KVMoIP (console). Examples of microscope functions include adjusting stage position, lens focus, and magnification levels. The GUI application actually resides on a computer directly connected to the microscope's video output and control input ports. Figures 2(a) and 2(b) show two distinct video activity levels (i.e., temporal and spatial characteristics in the GUI application images sent from the microscope to the remote user). We can notice that the images contain live video feeds of instrument cameras with high video activity levels, or text and graphs with low video activity levels.

Let b_{action} be the average bit rate of the TCP control traffic that is generated due to keystrokes and mouse moves/clicks at the user end to accomplish a particular microscope function. The user-activity input to the system during a session involving n microscope functions can be denoted by $b_{\text{user-activity}}$ given in

$$b_{\text{user-activity}} = \sum_{i=1}^{n} (b_{i\text{th action}}). \tag{1}$$

For such an input, the average video image transfer rate (i.e., RTP media traffic output at the microscope end (b_{out})) can be denoted as follows:

$$b_{\text{out}} = \psi_{\text{net}}^{f} G(b_{\text{in}} + b_{\text{seed}}), \tag{2}$$

where ψ_{net}^{f} is the network connection quality between the user and the microscope. The network connection quality refers to the end-to-end throughput that is affected by network congestion and network fault events. The G corresponds to the input-output scaling factor which is unique for a microscope function. The b_{seed} corresponds to the rate at which periodic intracoded frames (I-frames) are sent from the encoder (at the microscope) to the decoder (at the user) for quick image refresh upon recovery from network partition events during a session.

Although b_{out} is sent from the microscope, there are two network factors that could degrade the average video image transfer rate at the user end (b'_{out}). The first factor is the network connection quality of the reverse path (i.e., between the microscope and the user (ψ_{net}^{r})). The second factor is the available bandwidth in the intermediate network path. As shown in the following equation, if adequate available bandwidth is provisioned, b'_{out} will be equal to b_{out}; otherwise, b'_{out} is limited to b_{net}, which refers to the bottleneck hop bandwidth:

$$\begin{aligned} b_{\text{net}} &= \min_{i=1..\text{hops}} b_{i\text{th hop}}, \\ b'_{\text{out}} &= \min(b_{\text{snd}}, b_{\text{net}}). \end{aligned} \tag{3}$$

The degradation of b'_{out} manifests to users as video signal impairments such as frame freezing, blurriness, and tiling [18]. Based on the positive or negative b'_{out} feedback received at the user end from the microscope, the subsequent user behavior determines the session state. We refer to this system-state control parameter that is dependent on the user behavior as H. Details of how H parameter impacts the different system states are described in Section 3.2. We can thus express b_{in} as follows:

$$b_{\text{in}} = b_{\text{user-activity}} - H b'_{\text{out}}. \tag{4}$$

Using substitutions in (1)–(4), we can derive the closed-loop transfer function in the classical form as shown in the following equation; this function fully describes the order, type, and frequency response for a remote microscopy system:

$$\frac{b'_{\text{out}}}{b_{\text{user-activity}}} = \frac{G\psi_{\text{net}}^{f}\psi_{\text{net}}^{r}}{1 \pm G\psi_{\text{net}}^{f}\psi_{\text{net}}^{r}H}. \tag{5}$$

Ultimately at the end of a session, the overall user QoE (q_{mos}) will depend on both the effort a user had to expend to perform n actions (i.e., $b_{\text{user-activity}}$) and the perceivable video image quality (i.e., b'_{out} during those actions). Hence, q_{mos} can be expressed as follows:

$$q_{\text{mos}} = f\left(\underbrace{b_{\text{user-activity}}}_{\text{Demand}}, \underbrace{b'_{\text{out}}}_{\text{Supply}}\right). \tag{6}$$

From (6), we can make an analogous comparison of $b_{\text{user-activity}}$ and b'_{out} to the "demand" and "supply" terminology used in economics, respectively. In traditional economics, an increase in demand levels for a commodity causes an increase in supply levels of the commodity. This in turn increases the demand, as the increased supply in large numbers generally drives down the overall commodity price. As long as both the demand and supply increase hand-in-hand by deriving reinforcement from each other, the economy (analogous to q_{mos}) is considered to be in a productive state. However, this is not always the case in the demand and supply reinforcement effect seen in remote microscopy with respect to q_{mos}. The overall network health in both the forward and reverse paths (ψ_{net}) adds complexity in the relationship of the demand and supply variables as elaborated in the next subsection, which severely affects the q_{mos}.

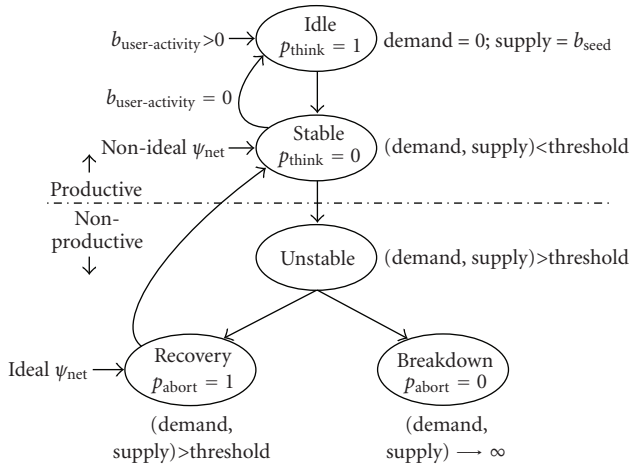


Figure 3: Remote microscopy system state transitions.

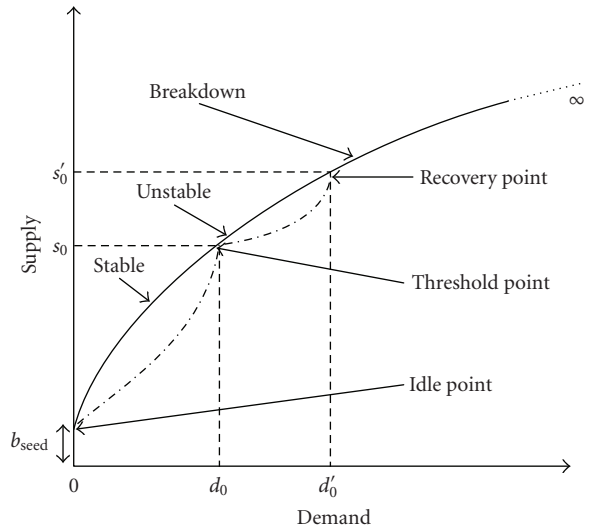


Figure 4: Remote microscopy system performance at different supply and demand conditions.

As a note, the above remote microscopy session model can be applied for both remote observation as well as remote operation use-cases. Recall that the remote operation use-case employs inband TCP control traffic toward the microscope, whereas remote observation use-case employs an out-of-band voice channel (e.g., a telephone) for directing control messages to a local user at the microscope. If we assume that a reliable voice channel exists between the two users and that the local user is responsive enough that the remote user does not perceive annoying control delays, the model is identical for both the use-cases.

3.2. System states

We now explain the interactions of the remote microscopy session parameters due to user behavior that affect the H parameter. The changes in the H parameter influence the $\pm$ sign (positive or negative feedback) of the denominator in (5) which in turn causes the different system state transitions shown in Figure 3. Initially, the system is in the "Idle" state when the user is inactive with a probability p_{idle} and the microscope GUI application is operational. In the Idle state, the demand is zero and the supply equals b_{seed} as shown in Figure 4. The remote microscopy session begins upon user-activity, and the demand and supply steadily increase. Assuming ideal ψ_{net} conditions at a given time t, the system attains a "Stable" state where the demand and supply are below the system's optimum performance threshold point (s_0, d_0). In this state, the user is successfully controlling the microscope functions and is being productive. We can now say that H is causing negative feedback in the system. At random times in this state, it is possible that a user will still be in session but idle in terms of control, presumably due to a thought process driven by a visual inspection of a sample's area of interest. Such an inactive user behavior brings the system back to its Idle state where the system is still productive. During such user inactivity times under ideal ψ_{net} conditions, we refer to p_{idle} as p_{think} as follows:

$$p_{\text{idle}}(t) = \begin{cases} p_{\text{think}}, & \text{if ideal } \psi_{\text{net}}(t), \\ p_{\text{abort}}, & \text{if nonideal } \psi_{\text{net}}(t). \end{cases} \tag{7}$$

If the ψ_{net} were to change to nonideal conditions due to network bottlenecks caused by network congestion and network fault events, the system would enter an "Unstable" state. Here, the demand and supply rapidly increase beyond the system's optimum performance threshold point. This is because the user in this system state experiences QoE degradation effects (e.g., frame freeze) that force him to misjudge his control actions that result in *unwanted* supply. This is subsequently followed by a retry of the previous actions before the unwanted supply transfer completes, which further increases the demand and the QoE degradation effects and so on. Soon, the system becomes nonresponsive to the increasing demand, and is pressured into handling large volume of unwanted supply that is introduced from the microscope end. It is important to note that although the demand and supply rapidly increase hand-in-hand beyond the threshold point, the system is nonproductive. We can now say that H is causing positive feedback in the system. If the user persists in his retry demand behavior, the system soon advances to a "Breakdown" state where the demand and supply tend to ∞. However, if the user aborts any actions and becomes idle at a recovery point (s_0', d_0'), the system transitions into a "Recovery" state. During such user inactivity times under nonideal ψ_{net} conditions, we refer to p_{idle} as p_{abort} as shown in (7). During the Recovery state, the demand and supply gradually tend toward the system's optimum performance threshold point. Once the ψ_{net} returns to ideal conditions (e.g., due to reduced network congestion or stabilization of the impulsive demand and unwanted supply), the system regains its Stable state and becomes productive again.

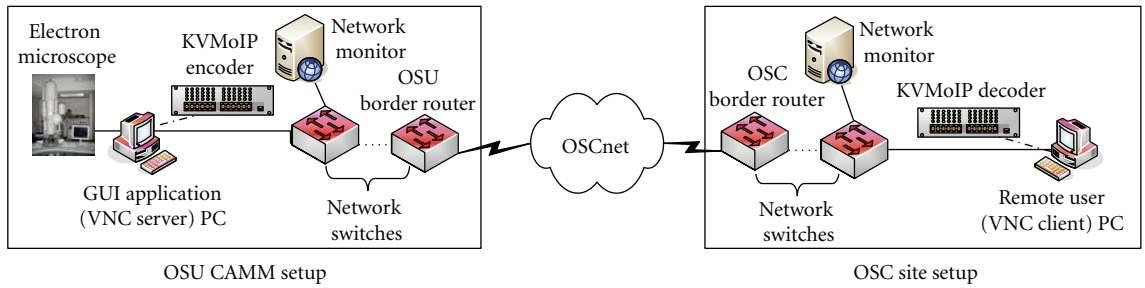


Figure 5: Remote microscopy testbed setup.

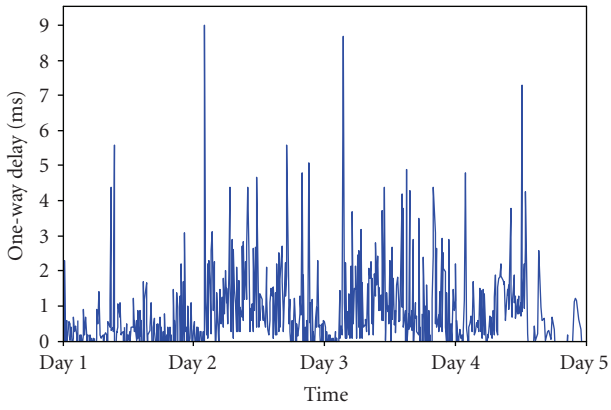


Figure 6: One-way delay between OSC and OSU CAMM.

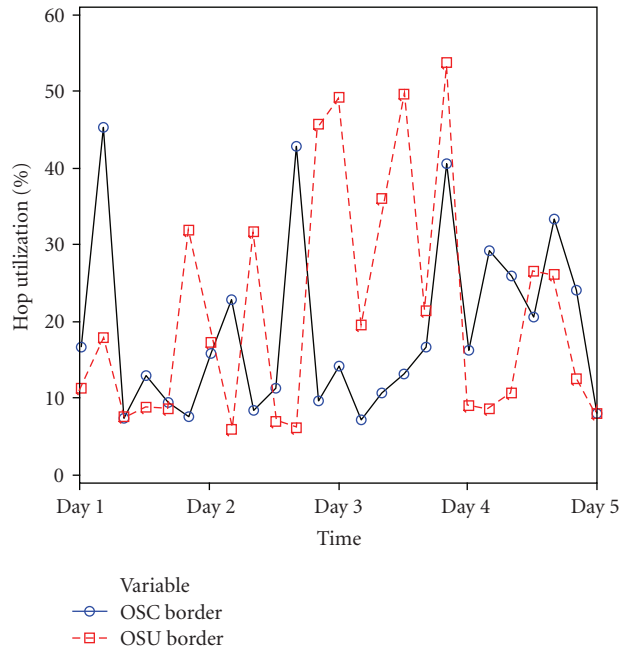


Figure 7: Border hop utilization at OSC and OSU.

4. REMOTE MICROSCOPY TESTBED EXPERIMENTS

In this section, we first describe the remote microscopy testbed used to obtain trends of the different session model parameters under different network conditions. Next, we explain the test cases and performance measurements collected during the QoE experiments on the testbed. Lastly, we discuss the QoE experiment results.

4.1. Testbed setup

For setting up the remote microscopy testbed, we collaborated with The Ohio State University's Center for Accelerated Maturation of Materials (OSU CAMM). The testbed featured four different network connections between the remote user console and the GUI application PC: (i) Direct GigE (ii) Isolated LAN (iii) Public LAN, and (iv) WAN. The Direct GigE connection had a Cisco GigE switch connecting the GUI application PC and the remote user console, which were in adjacent rooms. This connection represents the setup for avoiding users to be physically present at the microscope, especially when human presence around a sample is undesirable as explained in Section 1. The Isolated LAN connection was setup by including the CAMM's Cisco Catalyst 2924 switch to the Direct GigE connection. This connection represents remote microscopy for users in the same LAN as the microscopes, but in different lab rooms. The Public LAN connection was setup by including three additional Cisco Catalyst 2924 switches located at neighboring buildings to the Isolated LAN connection. This connection represents remote microscopy for users working from different LANs and different lab rooms. Finally, the WAN connection was setup as shown in Figure 5 via OSCnet between OSU CAMM and OSC. This connection represents remote microscopy for users at remote sites on the Internet.

The Direct GigE, Isolated LAN, and Public LAN connections were 100 Mbps switched full-duplex connections. To know the baseline performance of the 100 Mbps WAN connection, a number of measurements were collected over a 5-day period using the OSC-developed ActiveMon software [19]. The measurements indicated the available bandwidth, delay, jitter, and loss trends. Figure 6 shows the one-way delay measurements in the path between OSC and OSU (i.e., remote user to microscope) as measured by the OWAMP tool [20] in the ActiveMon measurement toolkit. We can observe that the one-way delay measurements are generally within 10 microseconds. Figure 7 shows the bandwidth utilization levels (sampled once every 4 hours) at the bottleneck hops

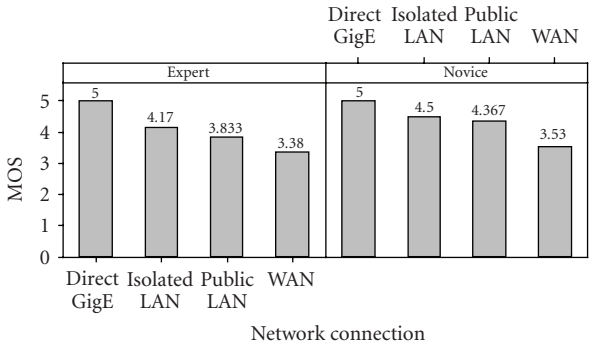


Figure 8: MOS (q_{mos}) rankings comparison for varying ψ_{net} conditions.

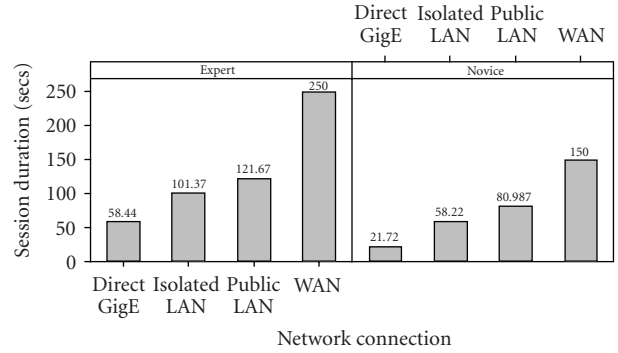


Figure 9: Session duration (T) comparison for varying ψ_{net} conditions.

(i.e., the border routers at OSU and OSC). We can observe that worst-case utilization level is ≈50% and thus, in general, there is at least about 50 Mbps available bandwidth in the WAN path. The other routers in the WAN path are the OSCnet routers with utilization levels <20%, typical to most backbone routers.

4.2. Test cases and measurements

The test cases involved performing preassigned tasks by actual users in remote microscopy sessions using a KVMoIP VNC solution [12] over the different network connections. Raw pixel format with copy-rectangle video encoding was used in all the test cases. The actual users (i.e., human subjects) were classified under two groups: (i) "novice" and (ii) "expert," with three human subjects in each group. The novice users performed a set of sequential tasks with simplistic actions: *Task-1:* move view from one location on the surface of sample material to another location, *Task-2:* focus on high-resolution imaging, and *Task-3:* change the quad-screen to a single screen and grab a high resolution image. The expert users performed a set of sequential tasks with advanced actions that require relatively more effort and skill: *Task-1:* eucentric height adjustment—stage movement in the Z-direction, *Task-2:* beam modulation—column alignment for best image, and *Task-3:* focus for high-resolution imaging.

During execution of the test cases, both objective and subjective measurements were collected. The objective measurements correspond to passive measurements of the control traffic (b_{in}) and video traffic (b'_{out}) collected using the popular TCPdump packet sniffing tool. The subjective measurements are the user QoE measurements, which are collected using the popular mean opinion score (MOS) ranking technique [21]. In this technique, at the end of each test case (i.e., remote instrumentation session task), the user is asked to rate his/her perceived QoE (q_{mos}) on a subjective scale of 1–5, with [1, 3) range being *Poor* grade, [3, 4) range being *Acceptable* grade, and [4, 5] range being *Good* grade. In addition to the MOS rankings, completion times (T) of novice and expert sessions were also recorded.

4.3. QoE experiment results

4.3.1. Network connection and user QoE

First, we analyze the impact of network connection quality on the user QoE MOS (i.e., q_{mos}) in a remote microscopy session. Figure 8 shows the average q_{mos} comparison between the novice and expert users for varying ψ_{net} conditions, with ψ_{net} being the highest for Direct GigE, and the lowest for WAN. For both types of users, we can observe that the q_{mos} rankings decrease notably with decreasing ψ_{net}. The q_{mos} equal 5 rankings of the novice and expert users while using the direct GigE connection indicates "at-the-microscope" QoE. Expectedly, for the other network connections, we can see that novice rankings are relatively more liberal than expert rankings due to the inherent intensity of the actions involved. The q_{mos} in the case of Isolated LAN and Public LAN are comparable. Further, the q_{mos} rankings for the WAN connection are in the acceptable grade, suggesting that user QoE in remote microscopy is highly sensitive to network congestion.

The average time to complete a set of predetermined user actions (i.e., session duration T) is another useful metric that provides insight about the user QoE. Figure 9 shows the session durations of the novice and expert for varying ψ_{net} conditions. For both types of users, we can observe that the session duration T increases with the decrease in ψ_{net}. For instance, the session duration T more than doubles in the Isolated and Public LAN connections in comparison to the Direct GigE connection. In the same context, we can see that the MOS ranking dip is higher for the expert (from 5 to 4.17) than the novice (from 5 to 4.5). We can generalize this observation of lesser MOS rankings for higher session durations across the other cases of both expert and novice users by comparing the session durations in Figure 9 with the MOS rankings in Figure 8. The longer session durations can be attributed to the additional effort (e.g., mouse moves/clicks, keyboard strokes, waiting for image transfer) involved while coping with network health fluctuations. Given that the difficulty level of expert user tasks is higher than novice user tasks, we can see that MOS ranking dip is higher in the expert user cases than the novice user cases.

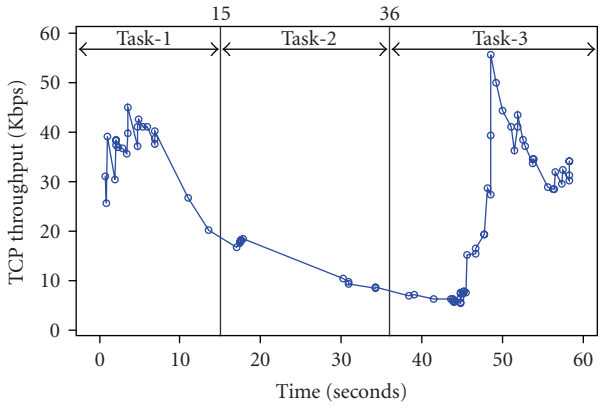


Figure 10: Control traffic (b_{in}) during an expert session on direct GigE network connection.

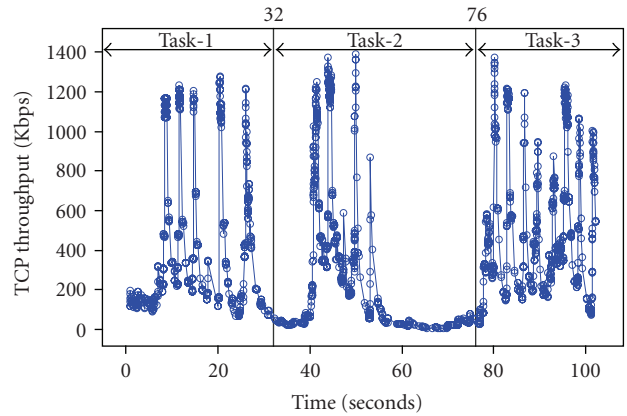


Figure 11: Control traffic (b_{in}) during an expert session on public LAN network connection.

4.3.2. Network connection and user control

Next, we characterize how the network connection quality impacts the trends of user control behavior (i.e., b_{in}). Figures 10 and 11 show the instantaneous b_{in} throughput levels during an expert session on Direct GigE and Public LAN connections, respectively. The throughput levels clearly show the amount of user effort required for accomplishing each of the three tasks of the session. Another notable observation is that user effort is considerably less ($\lceil b_{in} \rceil \approx 60$ Kbps) in the case of the Direct GigE connection as compared to the user effort ($\lceil b_{in} \rceil \approx 1400$ Kbps) on the Public LAN connection. Also, the throughput trends are significantly less dense in case of the Direct GigE connection as compared to the Public LAN connection. Due to space constraints, we do not show the throughputs for the WAN network connection, where the expert user effort was the most when compared to the other connections ($\lceil b_{in} \rceil \approx 2000$ Kbps).

We note that such an inverse relationship between the network connection quality and user control effort is a driver for the "congestion begets more congestion" phenomenon, where a user expends more effort (i.e., mouse moves/clicks and keyboard strokes) on poor network connections, which cumulatively adds to the congestion already inherent in the poor network connections. The nature of the "Unstable" and "Breakdown" states and their transitions explained in Section 3.2 can be attributed to the occurrence of this particular phenomenon with different intensity levels. The intensity levels are based on the instantaneous network connection quality and the impulsive user reactions to video signal impairments such as frame freezing.

4.3.3. User behavior and video image transfers

Lastly, we analyze how a user's control behavior and network conditions impact the video image transfers from the microscope at the user end (i.e., b'_{out}). Figure 12 shows the b'_{out} comparison between the novice and expert for varying ψ_{net} conditions. Cross-referring to the q_{mos} equals 5 results shown in Figure 8 for the direct GigE network connection, we can observe that obtaining an "at-the-microscope" user

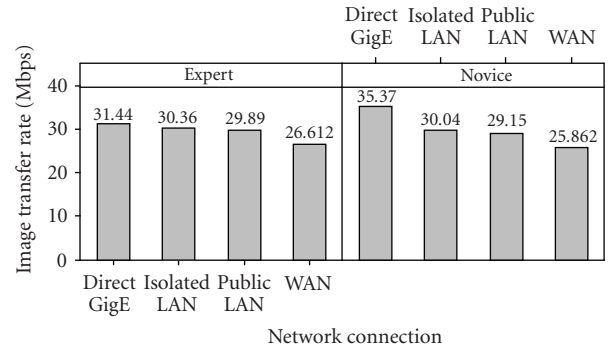


Figure 12: Image transfer rate (b'_{out}) comparison for varying ψ_{net} conditions.

QoE requires end-to-end available bandwidth in excess of 30 Mbps between the user and the microscope ends.

However, it is important to note that the average image transfer rates in remote microscopy can vary based on the microscope functions and user activity. Thus, they may not always be in the range of 30 Mbps. The reason for the high bandwidth consumption in our experiments can be attributed to the KVMoIP solution nature, and activity level in the experiments that had the quad-video panel images in the GUI application shown in Figure 2(a). Such a nature of video may not be present in every user session. For example, there may be sessions whose activity level may be similar to that of Figure 2(b), where the user is mainly plotting graphs, editing parameters while analyzing a sample. For such a session, the end-to-end available bandwidth requirement to achieve "at-the-microscope" user QoE will be considerably less.

5. REMOTE INSTRUMENTATION COLLABORATION ENVIRONMENT (RICE)

In this section, we describe the Remote Instrumentation Collaboration Environment (RICE) software application we have developed. The RICE design leverages our user and network interplay studies to effectively support the remote

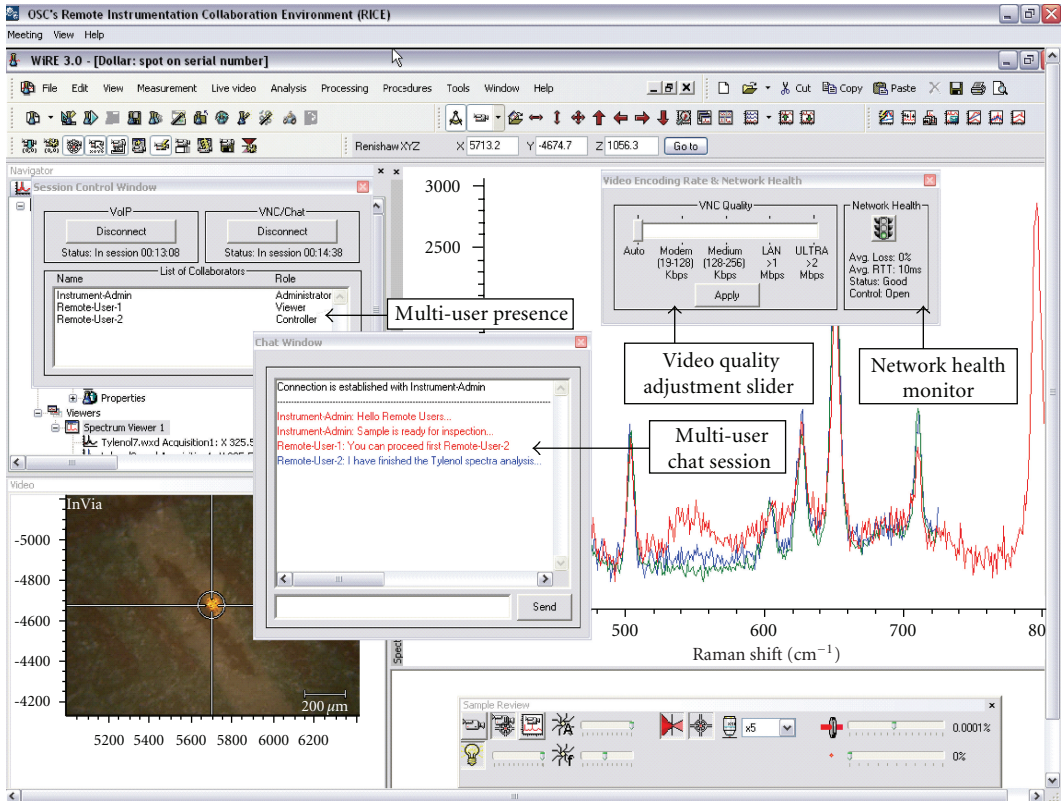


Figure 13: RICE in an active session.

observation and operation use-cases for instructors and researchers to train students and/or conduct research from remote locations on the Internet. RICE is based on the UltraVNC solution [10] but has several enhancements targeted for network-aware and collaborative remote instrumentation sessions that are reliable and efficient. The enhancements allow tuning of image feeds based on last-mile network bandwidth, and limit the provision of excessive control given by off-the-shelf VNC solutions during network congestion periods. Also, the enhancements provide collaboration tools (VoIP, chat, presence, control-lock passing) to orchestrate instrument-control amongst multiple remote users during remote operation of expensive and potentially dangerous instruments. Note that the network-aware control blocking and collaboration tools in RICE are features not available in off-the-shelf VNC solutions. The modular software design used in RICE permits customization to cater to unique considerations and requirements of a variety of users and instruments. Thus, RICE is a self-contained collaboration environment for multiple users to reliably and efficiently participate in remote instrumentation sessions.

Figure 13 shows the main functionality of RICE in an active session during remote operation of a Renishaw Raman Spectroscope deployed at the Department of Chemistry, The Ohio State University. The "video quality adjustment slider" is used by a RICE client user to manually adjust frame rates and video encoding rates based on the last-mile network bandwidth between the remote user site and instrument lab. A "network health monitor" shows real-time network health in terms of average round-trip delay (RTT) and loss. A traffic light indicates the network health grade as *Good* (green), *Acceptable* (amber), and *Poor* (red). The *Good* grade corresponds to RTT values in the range [0–150) milliseconds and loss values in the range [0–0.5)%; *Acceptable* grade corresponds to RTT values in the range (150–300) milliseconds and loss values in the range (0.5–1.5)%; *Poor* grade corresponds to RTT values in the range (>300] milliseconds and loss values in the range (>1.5]%. Such grade levels of RTT and loss have been obtained by studies such as [22, 23] that have conducted empirical experiments on the Internet for real-time multimedia applications. The network health monitoring in RICE is coupled with network performance anomaly detection using a "plateau-detector algorithm" that warns and blocks user's control-actions during impending and extreme network congestion periods, respectively. The control blocking is essential in spite of the fact that "limit switches" in some well-designed instruments mitigate damage due to user error. This is because it is practically infeasible to take into account all the possible user error cases when designing limit switches. The control status in RICE can be either "open," "warning," or "blocked" depending on the network congestion levels. Details of the network health monitoring coupled with the plateau-detector algorithm implementation in RICE are described in Section 5.1.

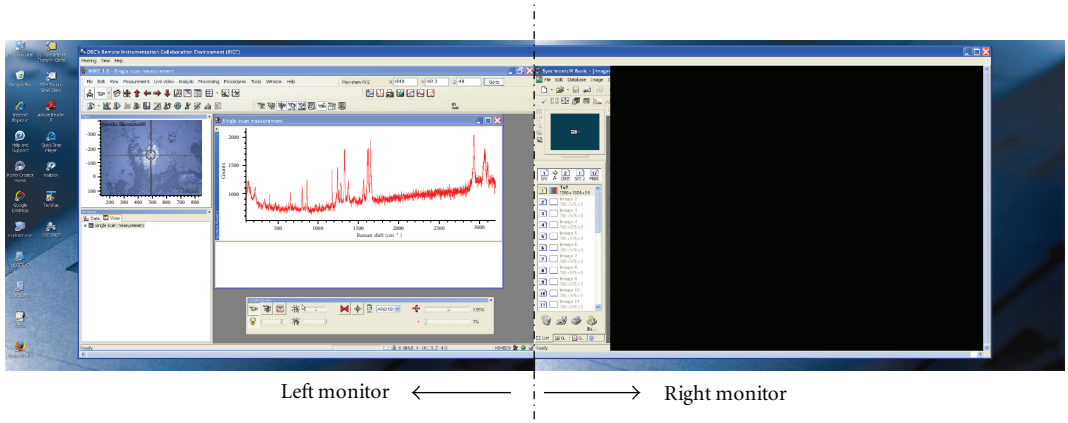


Figure 14: Problem with remote dual-screen resolution with UltraVNC.

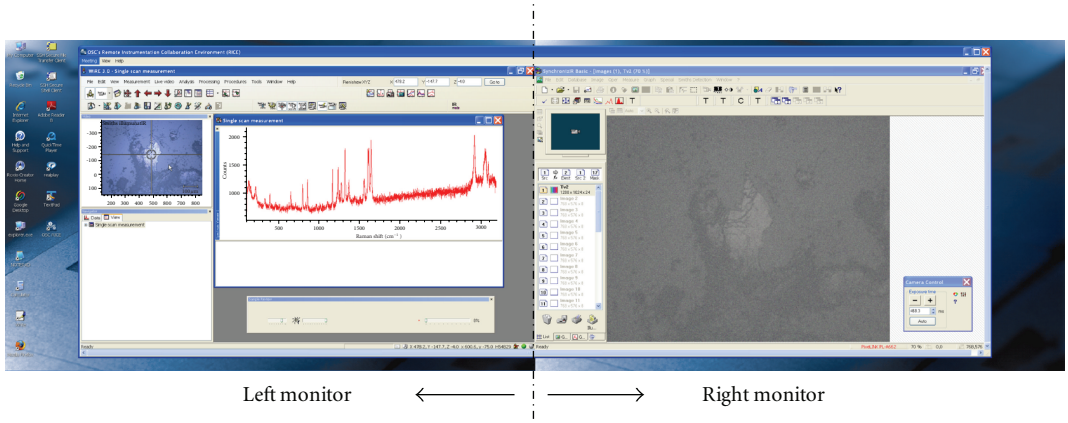


Figure 15: Increased remote dual-screen resolution with RICE.

RICE is designed to allow multiple remote users to simultaneously connect using RICE clients to the instrument console, which runs an instance of the RICE server. In a multiuser remote instrumentation session, RICE provides VoIP and chat functionality for collaboration as shown in Figure 13. The VoIP functionality is supported via the open-source OpenMCU software [24] that is integrated within RICE. The chat is supported by a custom chat application integrated within RICE that features colored text that can distinguish messages of the local user from the remote users. The names and roles of the multiple-users connected in a session are displayed and updated in real-time via a presence functionality. The roles correspond to the "administrator," "viewer," or "controller" (i.e., the users who possess the instrument control-lock, and the users who are observers). By default, the administrator always has control privileges, and he/she has the ability to pass another session control-lock to any one of the remote users. Once a remote user has a control-lock, he/she in turn can directly pass that control-lock in a session to any other remote user via the RICE client interface, without intervention of the administrator. All of the above multiuser collaboration tools of VoIP, chat, presence, and control-lock passing that are provided in RICE have been implemented using a "session-signaling protocol," whose details are described in Section 5.2.

RICE has a feature for simultaneously connecting to and transparently switching between two PCs for accomplishing inspection and analysis tasks of a remote instrumentation session. For example, in remote microscopy, a remote user will want to simultaneously connect and switch views between two PCs. One PC corresponds to the electron microscope console and the other PC corresponds to an energy dispersive spectroscopy (EDS) analysis PC. As another example, in remote telescopy, a remote user will want to simultaneously connect to a PC that shows the focal-plane instrument controls and another PC that is performing environment monitoring.

Further, we also developed a feature in RICE for handling remote dual-screen resolution. This feature allows a remote user to mimic an extended desktop setup with dual-monitors at the instrument computer. The extended desktop provides additional real-estate for users to run multiple application programs simultaneously. The openly-available UltraVNC distribution does not support dual-screen resolution at the remote VNC client as shown in Figure 14. We developed a software patch for the UltraVNC in RICE that increases the display resolution geometry to

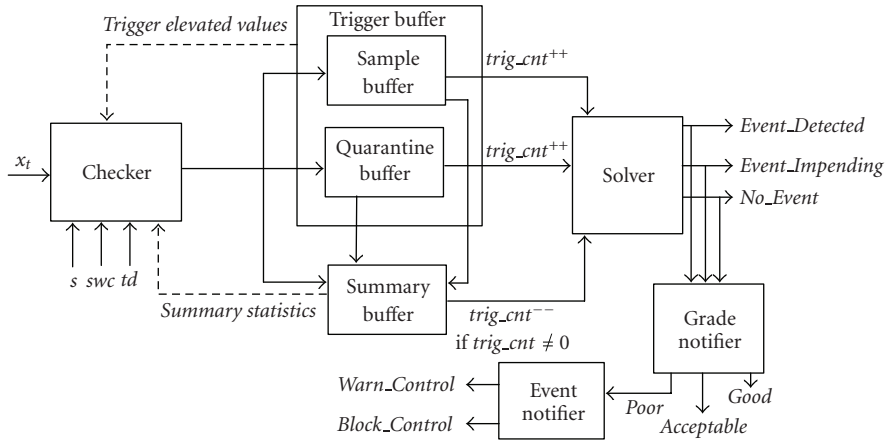


Figure 16: Plateau-detector block diagram.

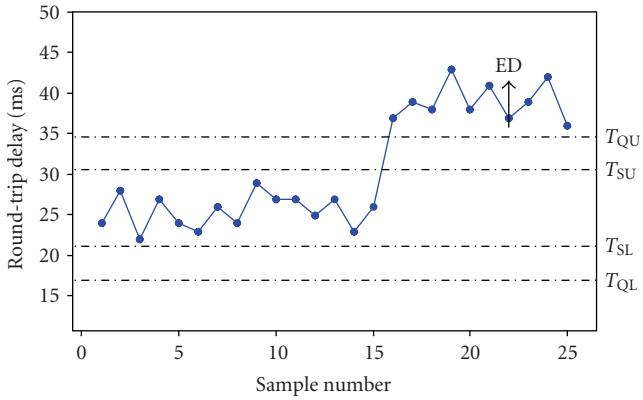


Figure 17: Plateau-detector thresholds illustration.

successfully render the dual-screen resolution at the remote VNC client as shown in Figure 15.

In the following subsections, we first describe the plateau-detector algorithm implementation and its integration in network health monitoring in RICE. Thereafter, we describe the session signaling protocol implementation to provide the collaboration tools functionality in RICE.

5.1. Network health monitoring

We use the popular *Ping* tool to monitor the network health in a RICE session in terms of average RTT and loss measurements. These RTT and loss metrics are useful to assess the impact of network health on the image transfer performance and interaction responsiveness as perceived by the user. Using RTT and loss measurements of Ping can generally indicate network congestion scenarios in most cases. However, it is possible in some cases that Ping's RTT measurements could misrepresent the network health status. For example, some routers are configured to handle Ping's ICMP packets differently to mitigate denial of service (DoS) attacks. In such cases, the RTT measurements might indicate higher values than actual, which can be construed as due to network congestion. Although there are several other sophisticated network measurement tools such as Iperf [25] and Pathchar [26] that can accurately measure available/per-hop bandwidth, they are not suitable for online network-aware adaptation in RICE. The reason is that these tools consume significant amounts of network bandwidth and CPU resources, and thus interfere with the RICE application performance. Besides, these sophisticated network measurement tools have been designed to be primarily used on network backbones for troubleshooting purposes rather than adaptation purposes in actual applications.

Our sampling rate of RTT and loss between the RICE client and server is periodic with a period of 6 seconds between consecutive samples. Each sample uses the default Ping settings (i.e., four ICMP packets each with packet size of 32 bytes). This results in time-series of instantaneous measurements of RTT and loss. We use x_t to denote an instantaneous measurement of RTT or loss that is input to a plateau-detector algorithm that detects network performance anomalies in real time. The purpose of the plateau-detector algorithm is to minimize the false-positive and false-negative anomaly alerts or triggers that arise if a naive mean-based method is used. A false-positive trigger is one that gets reported when there is no actual anomaly, whereas a false-negative trigger is one that does not get reported when in fact there is an actual anomaly. Two instances of plateau-detector algorithm run in-band in every RICE session, one to detect RTT anomalies, and the other to detect loss anomalies. The overall network health grade g and control status c are determined using an *OR* function of the two plateau-detector algorithm instance outputs.

Our plateau-detector algorithm implementation is similar to the basic implementations found in [27, 28] that are used for routine network monitoring. However, our implementation has several modifications to suit the time scales and user reaction times during network congestion periods in RICE sessions. In the mean-based method,

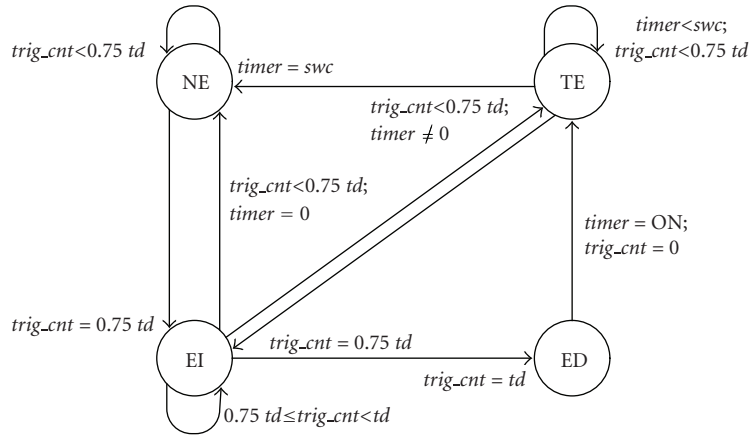


Figure 18: Plateau detector states and state-transitions.

the network health norm is determined by calculating the mean μ and comparing it with the standard deviation σ for a set of x_t values sampled most recently into a summary buffer. The number of samples in the summary buffer is user-defined and is specified using a summary window count *swc*. In comparison to the mean-based method, the plateau-detector requires two additional user-defined inputs called trigger duration *td* and sensitivity *s*. The trigger duration *td* specifies the duration of the anomaly before a trigger is signaled. Thus, the smaller the *td*, the faster a trigger will be signaled in the event of an anomaly. However, the *td* must be chosen to be large enough such that the transient spikes or bursts (i.e., noise events in network health are not signaled). The sensitivity *s* specifies the magnitude of the change for it to be considered as an anomaly. The choice of the *s* again requires consideration of the tradeoffs (i.e., a small *s* results in triggers for slight variations in network performance magnitudes, whereas a large *s* could overlook actual anomalies that should be detected).

Given our network health sampling rate of 6 seconds in RICE, we choose the *swc* to be equal to 20 so that anomalies can start getting detected within 2 minutes of initiation of a RICE session. In accordance with the *td* and *s* values selection in [27, 28], we choose *td* to be approximately 1/3rd *swc* (i.e., 7), and *s* value is chosen to be equal to 2. In this context, we remark that we have validated our selection of $s = 2$ using extensive simulations on synthetic and actual network health time series. The validation results are beyond the scope of this paper. Our general observation has been that values of *s* in the range of 2 produce the least number of false triggers (sum of false positives and false negatives), for a wide selection of *swc* and *td* values.

Figure 16 shows the different components of our plateau-detector algorithm implementation. The values of x_t are first input to a "checker" which compares whether the most recent x_t value lies within the upper and lower thresholds of the (i) summary buffer *sumbuff* (i.e., T_{SU} and T_{SL}) or (ii) quaratine buffer *qbuff* (i.e., T_{QU} and T_{QL}). These thresholds are illustrated in Figure 17 and are calculated using mean μ and standard deviation σ of the summary window as follows:

$$\begin{aligned} T_{SU} &= \mu + s * \sigma, \\ T_{QU} &= \mu + 2 * s * \sigma, \\ T_{SL} &= \mu - s * \sigma, \\ T_{QL} &= \mu - 2 * s * \sigma. \end{aligned} \tag{8}$$

If x_t values lie within these thresholds, the plateau-detector will be in the no event (NE) state shown in Figure 18. In this state, x_t values are put into the *sumbuff*. If x_t values go below T_{QL} or exceed T_{QU}, they are put into the quarantine buffer *qbuff*. Similarly, if x_t values cross T_{SL} and T_{SU}, they are put into the sample buffer *sampbuff*. If x_t is put into either *qbuff* or *sampbuff*, trigger count *trig_cnt* is incremented. Whereas, if x_t is put into *sumbuff*, *trig_cnt* is decremented as long as *trig_cnt* is nonzero. If *trig_cnt* exceeds $0.75^* td$ due to increasing number of x_t values going into *qbuff* or *sampbuff*, then the plateau-detector enters into an event impending (EI) state. If the *trig_cnt* drops below $0.75^* td$, then the plateau-detector returns to NE state. Otherwise, the plateau-detector stays in the EI state until *trig_cnt* equals *td*, after which it enters into an event detected (ED) state. Figure 17 shows an event detection occuring after x_t crosses the thresholds for the *td* of 7 samples. At this point, the *trig_cnt* is reset, and a *timer* is turned ON. The plateau-detector now goes into a trigger elevated (TE) state, where the upper and lower thresholds are calculated as follows:

$$\begin{aligned} T'_{SU} &= 1.2 * \max(x_t) \text{ in } \textit{trigbuff}, \\ T'_{QU} &= 1.4 * \max(x_t) \text{ in } \textit{trigbuff}, \\ T'_{SL} &= 0.8 * \max(x_t) \text{ in } \textit{trigbuff}, \\ T'_{QL} &= 0.6 * \max(x_t) \text{ in } \textit{trigbuff}. \end{aligned} \tag{9}$$

Until the *timer* equals *swc*, the elevated thresholds are used for comparing x_t. The reason for the trigger elevation is to avoid reporting of repeated triggers for the already detected anomaly. It is relevant to note that the plateau-detector can transition from TE state to EI state if another

```
(1) Input: summary window count (swc), sensitivity (s), trigger duration (td), instantaneous measurement (x_t)
(2) Output: mean μ, network health grade g, control status c
(3) begin procedure
(4) repeat
(5)   for each new x_t do
(6)       g = No_Event; c = Allow_Control
(7)       Calculate mean (μ) and standard deviation (σ) of summary window buffer (sumbuff)
(8)       if (timer == 0 /* NE state */) then
(9)         Calculate upper (T_SU, T_QU) and lower (T_SL, T_QL) thresholds using μ and σ of sumbuff and s
(10)      else
(11)        /* TE state */ Calculate upper (T_SU, T_QU) and lower (T_SL, T_QL) elevated thresholds using trigbuff
(12)      end if
(13)      /* Compare x_t with the thresholds and assign x_t to appropriate buffer */
(14)      if (T_QL > x_t or x_t > T_QU) then
(15)        Put x_t into quarantine buffer qbuff; Increment trig_cnt
(16)      end if
(17)      if (T_SL > x_t or x_t > T_SU) then
(18)        Put x_t into sample buffer sampbuff; Increment trig_cnt
(19)      end if
(20)      /* Report anomaly event types */
(21)      if (trig_cnt > td) then
(22)        event_type = Event_Detected; Copy sampbuff and qbuff to sumbuff; Reset trig_cnt
(23)      else if (trig_cnt > 0.75*td) then
(24)        event_type = Event_Impending
(25)      else
(26)        event_type = No_Event
(27)      end if
(28)      /* Update the buffers with latest network health norm */
(29)      if (trig_cnt == 0 and trigbuff not empty) then
(30)        Copy sampbuff to sumbuff; Empty qbuff
(31)      end if
(32)      /* Report μ of sumbuff, g and c*/
(33)      if (magnitude of x_t in Good grade level) then
(34)        return μ, g = Good, c = open
(35)      else if (magnitude of x_t in Acceptable grade level) then
(36)        return μ, g = Acceptable, c = open
(37)      else if (magnitude of x_t in Poor grade level) then
(38)        if (event_type == Event_Detected) then
(39)          return μ, g = Poor, c = Block_Control
(40)      end if
(41)      if (event_type == Event_Impending) then
(42)          return μ, g = Poor, c = Warn_Control
(43)        end if
(44)      end if
(45)    end for
(46) until end of RICE session
(47) end procedure
```

Algorithm 1: Plateau-detector algorithm implementation in RICE.

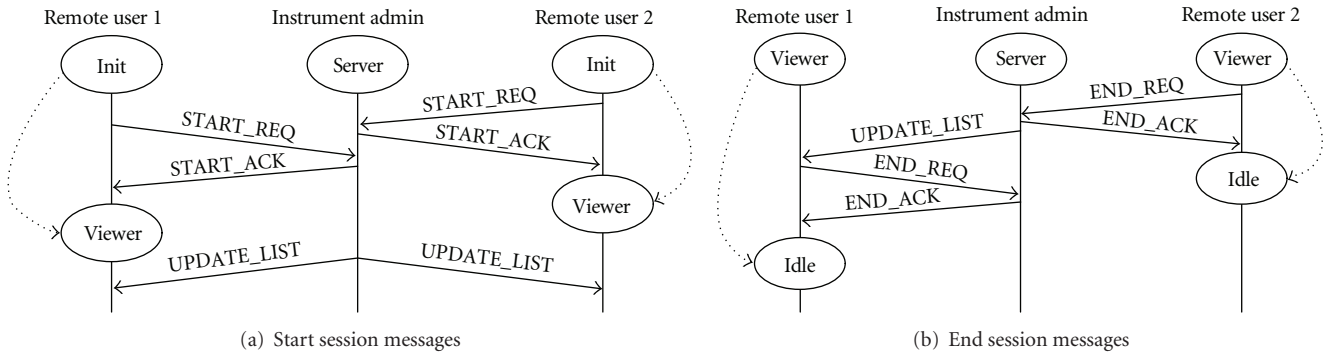
anomaly occurs due to x_t crossing the elevated thresholds. Once *timer* equals *swc*, and x_t does not cross the elevated thresholds, the plateau-detector returns to the NE state.

From Figure 16, we can see that the "solver" tracks the plateau-detector state transitions and outputs the *No_Event*, *Event_Impending*, and *Event_Detected* signals to the "grade notifier." If the network health grade is determined to be *Poor* as described in the RICE overview portion in Section 5, an "event notifier" is invoked to either warn the user of impending network congestion or to block any user control actions if a network congestion event is detected.

The implementation of the above plateau-detector algorithm in RICE can be formally described as shown in Algorithm 1 listing.

5.2. Collaboration tools

Herein, we describe the session signaling protocol we developed to provide the collaboration tools functionality in RICE. The protocol involves exchanging TCP-based messages on port 6000 between the RICE clients of the remote users and the RICE server at the instrument console.



(a) Start session messages (b) End session messages

Figure 19: RICE's Session Signaling Protocol messages for session initiation and termination.

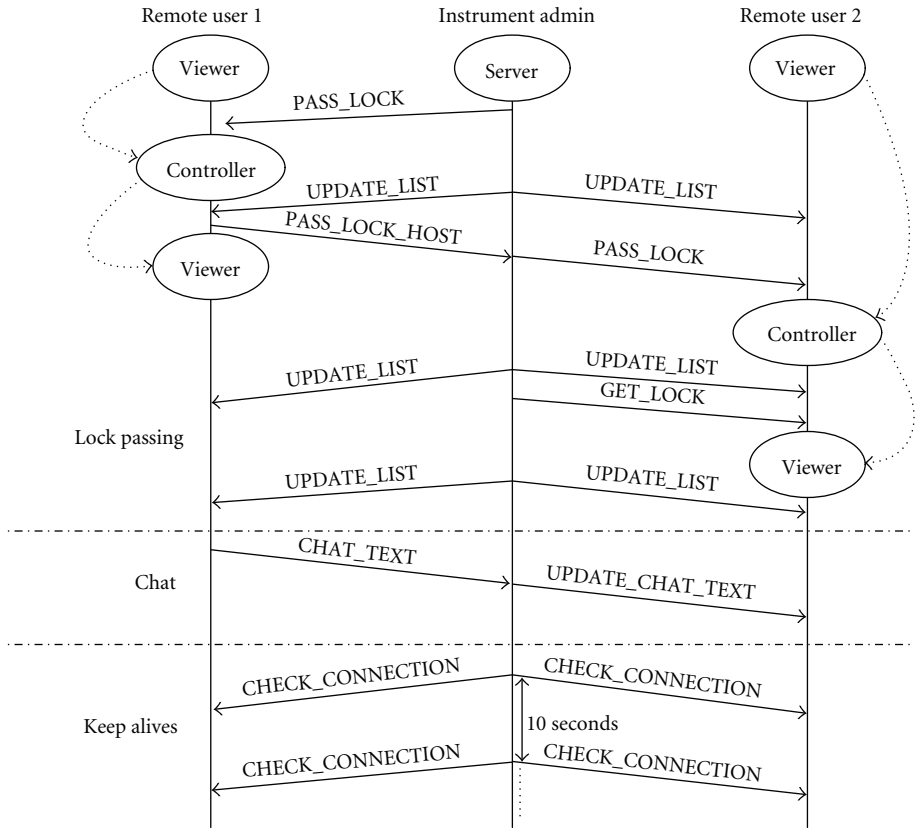


Figure 20: RICE's Session Signaling Protocol in-session messages.

Figures 19(a) and 19(b) show the start session and end session messages, respectively, for a RICE session involving two remote users. To start a session, a RICE client sends a START_REQ message to the RICE server. If a RICE server accepts the connection, it sends a START_ACK message back to the RICE client. Similarly, if a RICE client wants to terminate a session, it sends an END_REQ message to the RICE server, which in turn sends an END_ACK message to the RICE client. The UPDATE_LIST message maintains the latest presence information of all the users. It is event-driven and is sent from the RICE server to each of the RICE clients whenever there is a change of status in the RICE clients (i.e., whenever RICE clients join/leave the session, and whenever RICE clients switch between "viewer" and "controller" roles).

Figure 20 shows the different messages exchanged during a session when the collaboration tools are being used by the remote users. For the control-lock passing, we require the instrument administrator to initially pass the control-lock to any remote user. For this, the PASS_LOCK message is used. As we noted earlier, the instrument administrator always has a copy of the control-lock, and can retrieve the other control-lock from a remote user at any time. For the control-lock retrieval, the GET_LOCK message is used. Once

Table 1: List of RICE's session signaling protocol messages.

Message name	Purpose	Format
START_REQ	Sent from RICE client to RICE server to request a connection establishment	⟨START_REQ, USER_NAME, HOST_IP⟩
START_ACK	Sent from RICE server to RICE client to accept a connection request	⟨START_ACK, ADMIN_NAME⟩
UPDATE_LIST	Sent from RICE server to all connected RICE clients to update changes in user names/status	⟨UPDATE_LIST, USERS_STATUS[][]⟩
PASS_LOCK	Sent from RICE server to a RICE client to grant instrument control privileges	⟨PASS_LOCK⟩
PASS_LOCK_HOST	Sent from a RICE client to RICE server to pass control privileges to another RICE client	⟨PASS_LOCK, HOST_IP⟩
GET_LOCK	Sent from RICE server to a RICE client to revoke instrument control privileges	⟨GET_LOCK⟩
CHAT_TEXT	Sent from a RICE client to RICE server to send a chat text message to other connected RICE clients	⟨CHAT_TEXT, TEXT_INFO⟩
UPDATE_CHAT_TEXT	Sent from RICE server to all connected RICE clients to relay a new chat text message	⟨UPDATE_CHAT_TEXT, TEXT_INFO⟩
CHECK_CONNECTION	Sent from RICE server to all connected RICE clients to check connection validity	⟨CHECK_CONNECTION⟩
END_REQ	Sent from RICE client or RICE server to terminate a connection	⟨END_REQ⟩
END_ACK	Sent from RICE client or RICE server to acknowledge termination of a connection	⟨END_ACK⟩

a control-lock is given to a remote user, we also support the direct control-lock passing to another remote user, without the intervention of the instrument administrator. For this, the RICE client that has the control-lock sends a PASS_LOCK_HOST to the RICE server, which in turn sends a PASS_LOCK to the appropriate RICE client instantly. A similar relaying of chat text between remote users is performed by the RICE server. Here, the chat text from a RICE client is sent to the RICE server using the CHAT_TEXT message, which in turn sends an UPDATE_CHAT_TEXT message to all the other RICE clients. We remark that the relaying of VoIP traffic between the RICE clients is handled by the OpenMCU software using the standard ITU-T H.323 signaling protocols.

In case of any exception conditions, we use a time-out period of 10 seconds. Exception conditions correspond to cases where the RICE server is not running/accessible, or if one or more RICE clients are disconnected due to a network partition during an active session. To detect the latter case, CHECK_CONNECTION messages are sent every 10 seconds from the RICE server to each of the RICE clients. Table 1 summarizes the purpose and format of the various session signaling protocol messages in RICE.

5.3. Applications for research and training

In this section, we describe a few applications of RICE to foster research and training activities involving remote instrumentation. In the context of research, RICE allows multiple experts to jointly collaborate in investigations of samples loaded in scientific instruments. Each of the remote experts can use the quality adjustment slider to easily configure the appropriate frame rates of the image feeds from the instrument to match their respective last-mile network links. During a remote instrumentation session, the self-contained collaboration tools such as VoIP and chat in RICE enable the experts to efficiently communicate with each other. The control-lock passing feature ensures that there are no conflicts in remote operation of the instrument amongst the experts. Using RICE, each expert after completing his/her set of tasks on the instrument can instantly transfer the instrument-control to any other expert who wants to perform another set of tasks on the instrument. Both the control-lock passing and network health monitoring in RICE prevent inadvertent damages to the instrument by the remote users.

In the context of training, all of the above benefits of RICE mentioned for research are equally applicable, except the roles of remote users which are different. A number of students can use RICE to join and participate in a remote instrumentation session that is controlled by an instructor, who also could be remote using RICE. After the training session, the instructor can assign time-slots on the instrument for the students to complete their lab assignments. During the beginning and end of the lab time-slots of the students, the local instrument administrator can pass and retrieve the control-lock amongst the students, respectively.

RICE can be customized for integrating remote instrumentation with scientific web-portals that help in organization and archival of images and datasets acquired from instruments. Specifically, web-services and middleware defined in works such as [9, 29] can be implemented within RICE to automatically upload images and datasets

to the web-portal storage along with basic metadata such as instrument type and date and time stamps. Remote users can then login to the web-portal using their RICE account credentials and supply additional tags and annotations to their respective image and datasets for search and archival functions. If necessary, the users can even submit batch jobs of image processing filters or data analytics to compute clusters via the same web-portal interface. Thus, RICE can be an essential component in cyberinfrastructure deployments that aim at integrating instruments with networking, computing and storage resources for catering to the research and training needs of scientific user-communities.

6. CONCLUSION

In this paper, we modeled and characterized the complex interplay between the user control behavior and video image transfer performance in remote instrumentation sessions. Our remote instrumentation session model borrowed demand and supply terminology from traditional economics and identified the various system states (Idle, Stable, Unstable, Breakdown, and Recovery) and their transition conditions. The transition conditions were found to be primarily driven by time-varying user behavior and connection quality of the network path between the user and the instrument. Analyzing subjective and objective measurements of remote microscopy sessions involving actual users on LAN and WAN network paths, we found that (a) user QoE is highly sensitive to network health fluctuations caused by network congestion, (b) network health impacts both the user's control traffic throughput and microscope's video image transfer rates, and network congestion can hamper user productivity and cause the system to enter into unproductive states (i.e., Unstable and Breakdown states), and (c) the real-time control and video image transfer traffic is extremely bandwidth intensive for achieving "at-the-microscope" QoE. Thus, we demonstrated that remote instrumentation is a demanding network-based immersive multimedia application and is comparable to other applications of its class such as online-gaming, and high-definition videoconferencing.

We also described our RICE software functionalities that leverage our user and network interplay studies to cope with network bottlenecks, and cater to the requirements of remote observation and remote operation use-cases. In particular, we described in detail two main RICE functionalities that facilitated reliable and efficient remote instrumentation. The first functionality corresponded to the network health monitoring coupled with network performance anomaly detection using a "plateau-detector algorithm" that warns and blocks user's control-actions during network congestion periods. The second functionality corresponded to the "session-signaling protocol" we developed to enable multiuser collaboration in RICE using VoIP, chat, presence, and control-lock passing. Finally, we presented potential applications of RICE for research and training purposes that require remote instrumentation capabilities.

ACKNOWLEDGMENTS

This work has been supported in part by The Ohio Board of Regents and The Ohio State University's CAMM VIM program. The authors thank Dr. Peter Collins, Daniel Huber, Robert Williams, and Professor Hamish Fraser of OSU's CAMM for their participation in the remote microscopy testbed, RICE development feedback, and for providing microscope sample graphics used in this paper. They also thank Gordon Renkes (Department of Chemistry, The Ohio State University) for RICE testing, development feedback, and screenshots of RICE in active sessions. Further, they thank Dr. Dave Hudak, Neil Ludban, Dr. Eylem Ekici, Dr. Dong Xuan, and Dr. Steve Gordon for their useful suggestions during the course of the study. A preliminary version of this paper has appeared in the proceedings of ICST/ACM Immersive Telecommunications Conference (IMMERSCOM), 2007 [30].

REFERENCES

[1] "Chemistry Research Instrumentation and Facilities: Departmental Multi-User Instrumentation (CRIF:MU)," Program Solicitation NSF 07-552, National Science Foundation, Arlington, Va, USA, 2007.

[2] T. Richardson, Q. Stafford-Fraser, K. R. Wood, and A. Hopper, "Virtual network computing," *IEEE Internet Computing*, vol. 2, no. 1, pp. 33–38, 1998.

[3] K. W. Hodapp, J. B. Jensen, E. M. Irwin, et al., "The Gemini Near-Infrared Imager (NIRI)," *Publications of the Astronomical Society of the Pacific*, vol. 115, no. 814, pp. 1388–1406, 2003.

[4] K. Jeffay, T. Hudson, and M. Parris, "Beyond audio and video: multimedia networking support for distributed, immersive virtual environments," in *Proceedings of the 27th Euromicro Conference (EUROMICRO '01)*, pp. 300–307, Warsaw, Poland, September 2001.

[5] K. Furuya, M. Tanaka, K. Mitsuishi, et al., "Public opened internet electron microscopy in education field," in *Proceedings of the Microscopy and Microanalysis Conference*, vol. 10, pp. 1566–1567, Savannah, Ga, USA, August 2004.

[6] J. Kao, D. Troxel, and S. Kittipiyakul, "Internet remote microscope," in *Telemanipulator and Telepresence Technologies III*, vol. 2901 of *Proceedings of SPIE*, pp. 90–100, Boston, Mass, USA, November 1996.

[7] M. A. O'Keefe, B. Parvin, D. Owen, et al., "Automation for on-line remote-control in-situ electron microscopy," in *Proceedings of the Pfefferkorn Conference on Electron Image and Signal Processing*, Silver Bay, NY, USA, May 1996.

[8] M. Hadida, Y. Kadobayashi, S. Lamont, et al., "Advanced networking for telemicroscopy," in *Proceedings of the of the 10th Annual Internet Society Conference on Advanced Networking for Telemicroscopy (INET '00)*, Yokohama, Japan, July 2000.

[9] T. E. Molina, G. Yang, A. W. Lin, S. T. Peltier, and M. H. Ellisman, "A generalized service-oriented architecture for remote control of scientific imaging instruments," in *Proceedings of the 1st International Conference on e-Science and Grid Computing*, vol. 2005, pp. 550–556, Melbourne, Australia, December 2005.

[10] Ultra VNC Open-source Remote Access Solution, http://www.uvnc.com/.

[11] Real VNC Open-source Remote Access Solution, http://www.realvnc.com/.

[12] Thinklogical KVMoIP Remote Access Solution, http://www.thinklogical.com/.

[13] Avocent KVMoIP Remote Access Solution, http://www.avocent.com/.

[14] Adder KVMoIP Remote Access Solution, http://www.adder.com/.

[15] M. C. Wright, C. R. Hubbard, R. Lenarduzzi, and J. Rome, "Internet-based remote collaboration at the neutron residual stress facility at HFIR," in *Proceedings of the Workshop on New Opportunities for Better User Group Software (NOBUGS '02)*, Gaithersburg, Md, USA, November 2002.

[16] R. H. Cockrum, D. L. Clark, and S. T. Kelly, "Remote internet instrumentation for monitoring ocean data," in *Proceedings of the MTS/IEEE Conference and Exhibition (OCEANS '01)*, vol. 2, pp. 1176–1182, Honolulu, Hawaii, USA, November 2001.

[17] The Ohio State University CAMM VIM Program, http://www.camm.ohio-state.edu/.

[18] S. Winkler, *Digital Video Quality: Vision Models and Metrics*, John Wiley & Sons, New York, NY, USA, 2005.

[19] P. Calyam, D. Krymskiy, M. Sridharan, and P. Schopis, "TBI: end-to-end network performance measurement testbed for empirical bottleneck detection," in *Proceddings of the 1st International Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities (Tridentcom '05)*, pp. 290–298, Trento, Italy, February 2005.

[20] S. Shalunov and B. Teitelbaum, "One-way Active Measurement Protocol (OWAMP) Requirements," *IETF RFC 3763*, 2004.

[21] J. Mullin, L. Smallwood, A. Watson, and G. Wilson, "New techniques for assessing audio and video quality in real-time interactive communications," ETNA Project Report, IHM-HCI, Lille, France, 2001.

[22] ITU-T Recommendation G.114, "One-Way Transmission Time," 1996.

[23] P. Calyam, M. Sridharan, W. Mandrawa, and P. Schopis, "Performance measurement and analysis of H. 323 traffic," in *Proceedings of the 5th International Workshop on Passive and Active Network Measurement (PAM '04)*, pp. 137–146, Antibes Juan-les-Pins, France, April 2004.

[24] Open H.323 Project, http://sourceforge.net/projects/openh323.

[25] A. Tirumala, L. Cottrell, and T. Dunigan, "Measuring end-to-end bandwidth with Iperf using Web100," in *Proceedings of the 4th International Workshop on Passive and Active Network Measurement (PAM '03)*, La Jolla, Calif, USA, April 2003.

[26] A. B. Downey, "Using pathchar to estimate Internet link characteristics," in *Proceedings of the ACM SIGMETRICS International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS '99)*, vol. 27, no. 1, pp. 222–223, Atlata, Ga, USA, May 1999.

[27] A. McGregor and H.-W. Braoun, "Automated event detection for active measurement systems," in *Proceedings of the Workshop on Passive and Active Measurements (PAM '01)*, Amsterdam, The Netherlands, April 2001.

[28] C. Logg, L. Cottrell, and J. Navratil, "Experiences in traceroute and available bandwidth change analysis," in *Proceedings of the ACM SIGCOMM Workshop on Network Troubleshooting: Research, Theory and Operations Practice Meet Malfunctioning Reality*, pp. 247–252, Portland, Ore, USA, August-September 2004.

[29] I. Atkinson, D. du Boulay, C. Chee, et al., "Developing CIMA-based cyberinfrastructure for remote access to scientific instruments and collaborative e-research," in *Proceedings of the 5th Australasian Symposium on ACSW Frontiers*, vol. 249 of *ACM International Conference Series*, pp. 3–10, Ballarat, Australia, January-February 2007.

[30] P. Calyam, N. Howes, A. Kalash, and M. Haffner, "User and network interplay in internet telemicroscopy," in *Proceedings of the 1st ICST/ACM International Conference on Immersive Telecommunications (IMMERSCOM '07)*, Verona, Italy, October 2007.

Hindawi Publishing Corporation
Advances in Multimedia
Volume 2008, Article ID 569848, 11 pages
doi:10.1155/2008/569848

Research Article

Sensor Network-Based Localization for Continuous Tracking Applications: Implementation and Performance Evaluation

Livio Denegri, Sandro Zappatore, and Franco Davoli

Department of Communications, Computer and Systems Science, University of Genoa, 13 via all'Opera Pia, 16145 Genoa, Italy

Correspondence should be addressed to Livio Denegri, denegri@dist.unige.it

Received 7 March 2008; Accepted 17 June 2008

Recommended by Mohammed Ghanbari

The increasing interest in systems able to provide users with immersive services (e.g., domotics, context-aware applications, and immersive distance learning tools) has encouraged the development of cheap and effective platforms aimed at tracking objects and people within a certain space. In this context, wireless sensor networks (WSNs) can play a very important role, since specialized sensors can be fruitfully exploited in order to generate/receive signals by means of which the WSN can derive the position of nodes joined to the objects to be tracked. The paper presents an original localization platform that exploits a single-hop WSN, based on a Microchip MCU and a Cypress RF device, to track its moving nodes. Specifically, the nodes of the network are divided into three sets: the first set consists of anchor nodes that, according to the commands from the sink (the central node of the WSN), generate ultrasonic pulses. These pulses are received by the second set of (moving) nodes, which estimate the pulse time trip and communicate it to the sink. Finally, the last set is constituted by general purpose nodes that collect any kind of data from the surrounding field. The sink gathers all the data, computes the position of moving nodes, and transfers information to external users on the Internet. The algorithms adopted to manage the network and to localize moving nodes are discussed. A working prototype based upon the hardware platform, software, and protocol described in this paper has been deployed and tested, and some results are shown. Simulation results of the localization system are presented to show system scalability.

1. INTRODUCTION

Nowadays, more and more systems have localization as a key element to speed up work or to provide advanced services to users: in the so-called context-aware systems, for example, to have knowledge of the position is a basic feature around which all services are developed.

Moreover, localization functionalities are a fundamental component in immersive communications, at least in all situations involving user mobility; in some cases, users are represented by robots moving within a certain zone [1–3]. As pointed out, for example in [4], there are four common tasks in immersive visualization, namely, localization, orientation, navigation, and representation. Actually, many works on virtual immersive environments focus on the task of having a user localize a specific target (e.g., an audio source) [4–6]. However, equally important in such environments is often the symmetric problem of localization of the user on the part of the surrounding ambient technology (see, e.g., [7, 8]).

The problem of localization has been commonly solved by using different approaches, which rely upon the environment where the system has to operate. When the application is deployed in a wide outdoor environment, GPS is the most ordinary approach: it works well when receivers are in wide areas, but it can be useless in a more complex environment, such as narrow city streets or indoor spaces [9].

Other solutions have been developed to overcome these deficiencies: radio power maps and theoretical, or empirical, power-decrease laws are often used to estimate the distance from known fixed positions, for example by measuring received power from IEEE802.11 access points or cellular base stations. For indoor applications, also the approaches based on received power do not provide enough accuracy, owing to environmental complexity. Improvements are obtained when other techniques are adopted: solutions based on ultrasonics give better results than radio approaches and turn out to be cheaper too.

When high precision is the goal, systems are usually based on high-end technological devices [10] or very complex

infrastructures, which make the deployment expensive and difficult, especially indoor. Wiring takes most of the effort: solutions based on wireless communications can easily overcome this trouble and therefore can speed up system deployment.

Wireless sensor networks (WSNs) embody the idea of flexibility and easiness. They are composed of several simple devices, the nodes, which communicate by radio and cooperate to reach a goal. WSNs are commonly used to monitor very wide areas: nodes measure some quantity, for example temperature, and send data toward a specific node, called sink, which is the interface between the WSN and other networks. Specific protocols have to be developed to configure the network, by flooding topological information, and to route data from sensors to the sink. To overcome long distances, protocols must allow multihop transmissions. Nodes send packets to the sink by means of other nodes, but they must also be as simple as possible to save batteries and to cope with limited computational and storage capabilities of the nodes. WSNs can be organized as meshes, by exploiting multihop protocols, but also clusters or hierarchical structures can be adopted during network design.

Localization, in an indoor environment, can be well faced by using WSNs [11]: a subset of the nodes forms the fixed infrastructure, while the others are attached to what has to be localized. The availability of cheap single-chip computers and miniaturized radio-transceivers makes easy to design small-dimension nodes, which can be installed everywhere without many difficulties.

The approach presented in the paper mixes two different technologies, namely, radio and ultrasonics, which are jointly used in a novel integrated system of localization measurement and data collection. Time-of-flight measurements are performed by means of ultrasonic devices, which provide better precision for this task with respect to radio propagation. A wireless packet network based on polling is adopted for measurement (and other sensor data) collection. Decoupling the two tasks allows a more effective system design and the adoption of the best technology for each of them.

The paper is organized as follows. Section 2 describes the structure of the WSN, the nodes, and the algorithms implemented. Measurements done on the communication channel and localization precision results are shown in Section 3, while Section 4 shows the results of simulations performed with the localization system. Finally, in the last section, conclusions are drawn and future improvements to the system designed are proposed.

2. THE SENSOR NETWORK

Figure 1 sketches the devised sensor network, whose peripheral elements, the nodes, are the components of a platform devoted to the tracking application. Nodes can be divided in three main sets, according to the role they play within the network. The first set consists of the nodes, called anchor nodes (AN), that must be placed in fixed, suitable (and known) positions; the second set is represented by the moving nodes, here referred to as mobile nodes (MN),

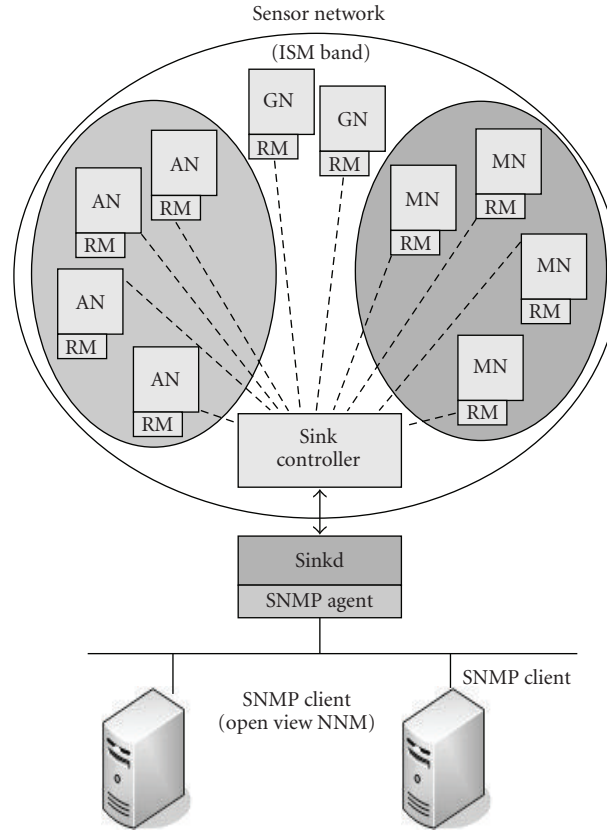


Figure 1: Overall structure of the proposed sensor network, publishing data by means of an SNMP agent. Data are accessed through SNMP clients (e.g., the commercial product OpenView Network Node Manager (OV NNM)).

which have to be tracked within a certain building or room. Eventually, the last set groups general-purpose nodes (GN), whose goal might be, for instance, the monitoring of some environmental parameter, as temperature. All these nodes directly communicate, via a radio modem (RM), with a central node, called sink, which gathers the data acquired by the peripheral nodes, synchronizes network operations, and acts as a "gateway" to an IP infrastructure.

It should be highlighted that, in our implementation, the hardware platform has been designed with off-the-shelf elements: this allows building very cheap nodes, characterized by good computational and communication capabilities. To this aim, during the design phase, much attention was paid in order to (i) adopt commercial components, (ii) choose integrated circuits providing PDIP (plastic dual inline package) to render any possible part replacement quite simple, (iii) exploit an RF transceiver operating in the free ISM band (2.4 GHz) able to internally spread and serialize/deserialize data to be transmitted, and (iv) keep the size of the node board as small as possible, thus permitting an easy deployment of the WSN.

The anchor, mobile, and general-purpose nodes are based upon the same hardware platform, consisting of a main board, on which all the electronic components are

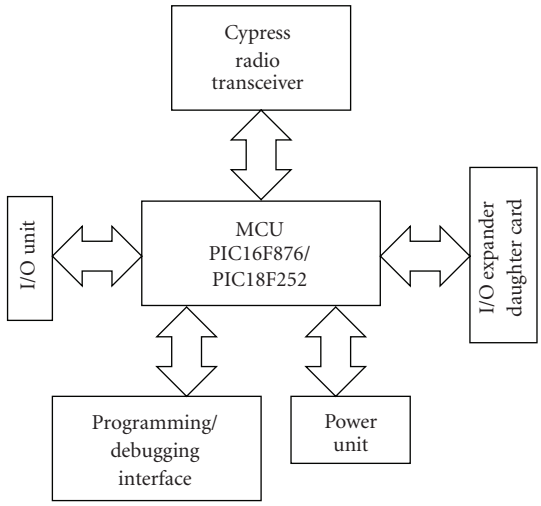


Figure 2: Functional blocks of a WSN node.

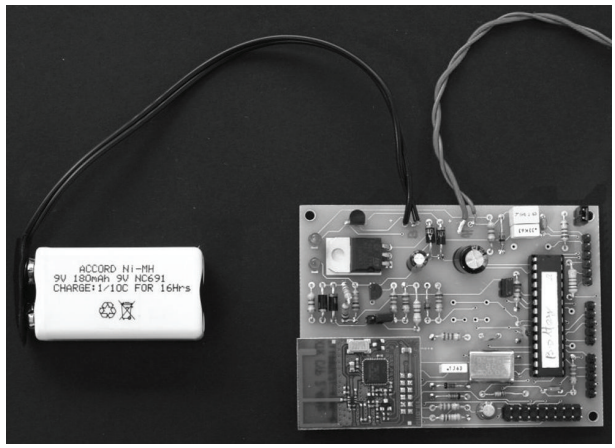


Figure 3: Main board of the implemented wireless sensor.

placed. The different "identities" of the nodes are provided by the specific daughter card plugged into the main board. At the moment, two daughter cards have been designed: one, used in the ANs, includes an amplifier and an ultrasonic emission device [12]. The other, employed in the MNs, includes an ultrasonic microphone [12], and a band pass front-end amplifier. The GNs require no daughter cards, at least for the most common data acquisitions.

The functional blocks of a main board are depicted in Figure 2. Its core component is represented by the MCU, a PIC 16f876 [13] (or PIC 18f252, i.e., pin-to-pin compatible with the previous one): the latter communicates, via an SPI (serial peripheral interface), with a radio transceiver, based on the Cypress CYWUSB6935 chip [14]. In our implementation, we adopted the Cypress-based "high-speed multichannel transceiver," produced by Aurel S.p.A., an Italian medium enterprise, specialized in RF modules' design [15].

All the analog inputs of the MCU, SPI, and PWM (pulse width modulator) lines and other MCU general-purpose signals are available at the connector, into which a daughter card can be plugged. In this manner, it is possible to control and acquire a greater number of signals, as well as to handle special purpose cards, consisting of an ad hoc electronic circuitry, as in the case of the MNs and ANs.

The programming/debugging interface provides an ICD (in-circuit debugging) tap, which allows the user to upload and debug the firmware directly on the MCU. The power unit (PU) monitors the battery status and, if an external power supply is available, the PU controls and regulates the battery charge. The I/O unit is represented by a very simple interface, which permits the MCU to acquire only a very limited number of signals from the field. Although the MCU can handle 8-digital (namely, on-off) inputs, 8-digital outputs, and 4 analog input channels, the on-board circuitry permits to manage only 1 digital input, 1 digital output, and 1 analog input channel: to achieve more I/O capacity a daughter card must be plugged in the main board.

This design choice is motivated by the fact that, in general, it is convenient to decouple the network operation facilities and functionalities, which reside on the main board of a node, from specific capabilities that involve the use of ad hoc components. Furthermore, it should be noted that, in the case of a GN, only a limited number of I/O lines are generally needed to meet data acquisition requirements.

Figure 3 presents the main board of a sensor developed within our research activity. The various protocol layers of the WSN, as well as the driver handling the RF transceiver, the software portions managing the power unit and the signal acquisition and signal conditioning are implemented on the MCU.

As already mentioned, the central element of the network is the sink, which includes two main elements: a network sink controller (NSC), and a sink manager (SM). The hardware and firmware of the former are similar to those of a common sensor node, while the latter consists of a single board ARM (Advanced RISC Machine) computer (booting Linux), produced by Technologic Systems [16].

The NSC actually manages the sensor network and provides a proper synchronization, thus permitting the MNs to estimate their distance from the ANs (see Section 2.4). The SM gathers the MNs' data related to distances (from ANs), and processes them according to a triangulation algorithm. Finally, the SM publishes MNs' positions by exploiting the facilities offered by an SNMP agent or by a Web Service.

To this aim, the SM continuously communicates with the NSC in order to get any information acquired from the MNs, and to send commands to the ANs, so that they are triggered to generate an ultrasonic pulse. Moreover, the SM collects data acquired from the GNs. The SM stores information from the nodes in an internal real-time database, which is also accessed by the processes involved in the triangulation procedure and in the publishing of data gathered by the GNs. The functional blocks of the sink are depicted in Figure 4.

2.1. Anchor node

In Figure 5, blocks forming the ultrasonic transmission electronics are depicted. As can be seen, the design has been done as simple as possible to reduce power consumption

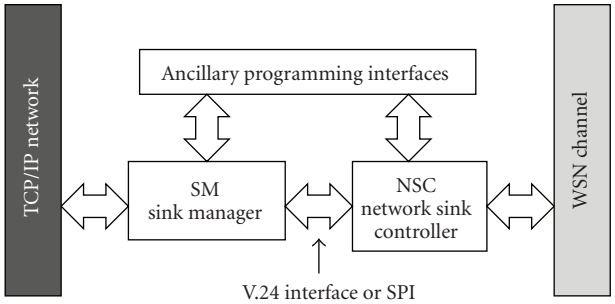


Figure 4: Functional blocks of the sink.

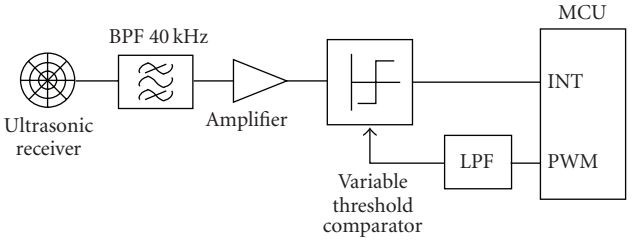


Figure 6: Functional blocks of the ultrasonic receiver.

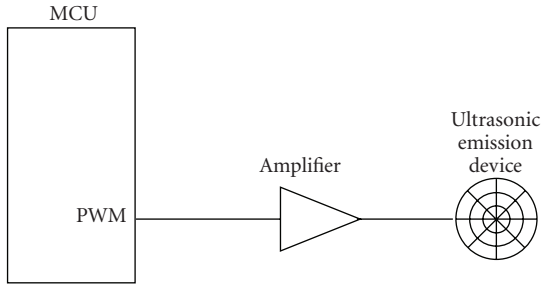


Figure 5: Functional blocks of the ultrasonic transmitter.

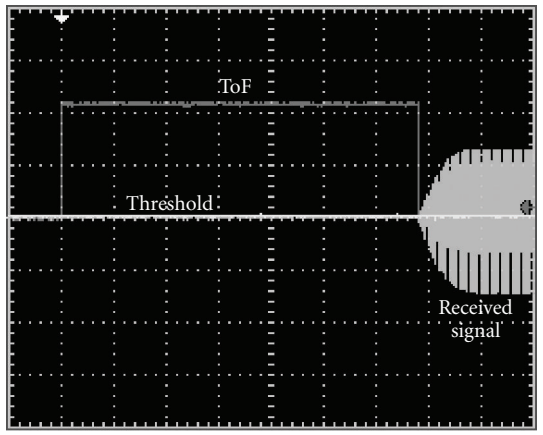


Figure 7: Screenshot showing the received signal, threshold level, and the measured ToF.

and dimension. All operations are controlled by the MCU of the main board. Ultrasonic pulses, at the frequency of 40 kHz, are generated by means of the PWM (pulse width modulation) signal produced by the PIC which drives an amplifier/voltage buffer connected directly to the ultrasonic emission device [12].

2.2. Mobile node

Mobile nodes ultrasonic electronics is composed by the blocks shown in Figure 6. As for anchor nodes, pulse detection is maintained as simply as possible.

The ultrasonic received signal is first filtered with a band pass filter (BPF) at 40 kHz and then amplified 1000 times to get voltages of about 2 volts. Pulse detection is performed by exploiting the MCU external interrupt (INT) facility. When the received signal exceeds a threshold, the interrupt is generated and the delay of the pulse is measured (see Figure 7). The threshold is generated by filtering the PWM signal provided by the MCU: by varying the duty cycle of the PWM, the mean value extracted by the LPF (low pass filter) can be changed. The threshold is determined before an MN begins its operations: its value is chosen above the maximum noise level during a listening window. Proceeding in this way, delay measurement errors are minimized and the system can adapt to different noise levels.

2.3. Communication protocol

The access protocol implemented is quite simple and can be considered a modified version of a polling algorithm. The basic idea divides the network operation in two different phases. During the former one, called polling phase (PP), the sink cyclically sends ANs the command to force the emission of an ultrasonic pulse train. Then, the sink polls the MNs, in order to receive, from each MN, the time spent by the pulse to reach it. In order to send/receive commands/data to an AN/MN, the sink transmits a POLL-REQuest to an addressed anchor/mobile node, which, in turn, must respond with a POLL-RESponse packet within a certain time interval.

After sending the commands to all the ANs and polling each MN, the second phase, named Aloha phase (AP), starts. The sink periodically broadcasts a beacon (BEAC) and listens to the channel for a certain amount of time. Upon receiving a beacon, a bound GN node may notify (by means of a TX-REQuest) the sink that it has new data acquired from the field, or an unbound node may ask to enter the network by sending a BIND-REQuest packet.

In the former case, the sink polls the GN, thus enabling it to transmit the information acquired; in the latter case, the sink binds the new node and sends it a Bind-RESponse packet, containing a unique identifier, by means of which the sink will poll the sensor in all the following data exchanges. Figure 8 diagrammatically represents different transaction types between sink and sensor nodes.

The various packet types used within the WSN are reported in Figure 9. The packet labeled (a) is transmitted

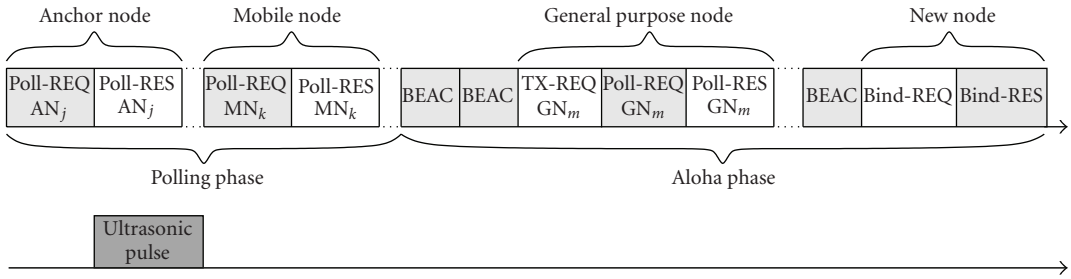


Figure 8: Time line showing: jth AN and kth MN polling (polling phase); mth GN request to send and new node binding (Aloha phase). Packets transmitted by the sink are shown in light grey, white packets are transmitted by nodes, and the ultrasonic pulse is shown in dark grey.

by the sink during the AP; packets (b) and (c) are exchanged by the sink and an unbound node that wants to enter the network. Packets (d) and (e) are used during the polling phases to communicate with ANs, MNs, and GNs; while the last packet (f) is transmitted by a GN to inform the sink it has data to send.

2.4. Synchronization, measure, and data gathering

When the network is set up, mobile nodes' localization can start: Figure 8 shows the operations taking place. The sink node begins by broadcasting a packet, commanding anchor node j to send an ultrasonic pulse. Because this packet is broadcast, also mobile nodes receive it: so they use this packet as a time reference to measure time-of-flight (ToF) of the next ultrasonic pulse. ToF measurement of each mobile node ends when it detects the ultrasonic pulse, as shown in Figure 10, or after a hard-coded timeout. Ultrasonic detection is performed by using the hardware described in Section 2.2. Since radio propagation delays can be accounted as zero due to the short node-to-node distances of a singlehop network, only firmware and hardware delays have to be canceled out from the ToF to get a precise distance measurement. These delays are caused by radio packet processing and electronics governing ultrasonic generation [17].

At a fixed time after command transmission, the sink begins the polling phase to collect measurements from every mobile node. Each polled node transmits its own ToF measurement to the sink; the latter records the measurement: when a mobile node has a set made up of at least three measurements, its position can be fixed. The algorithm used to estimate mobile node position is based on trilateration and will be described in Section 2.5.

The polling phase is the most time-consuming one, especially if a great number of mobile nodes have to be localized, because all mobile nodes' measurements have to be gathered. To improve this operation, a position-based approach has been introduced. When the system starts localization, the polling phase is executed by polling all the nodes associated with a sink in an impartial way. However, when nodes' positions are known, polling can be done by using a cleverer approach. Mobile nodes can only detect ultrasonic pulses coming from anchor nodes placed in a small area above them, so that when the sink commands anchor AN$_j$ to transmit the ultrasonic pulse, it subsequently polls only mobile nodes which were before localized near AN$_j$. Adopting this technique, which works under the hypothesis of slow movements, only few nodes are polled: therefore the duration of the polling phase is reduced.

2.5. Position fixing

When at least three time-of-flight measurements of the same mobile node are collected, a trilateration algorithm can be used to estimate node position [18].

Assuming the speed of sound equal to 344 m/s, distances of the mobile node from the anchor nodes are calculated. Heights from the ground are considered fixed, that is, each mobile node's height is known by the system and does not change during time. This hypothesis is easily fulfilled, for example, when mobile nodes are attached to racks, trolleys, or forklift trucks: during network start-up, mobile nodes can transmit their heights to the sink. Further improvements to the implemented algorithm will allow to remove this hypothesis.

The first step of the algorithm is to calculate the intersection points of the three time-of-flight-radius spheres: by projecting everything onto the plane where the mobile node lies (the height from the ground of anchor nodes and of the specific mobile node is used now) we will consider circles instead of spheres. The radius of a circle is tightly related to time-of-flight-radius, so we will use the same notation. Due to errors or inaccuracies during ToF calculation, this first step can give rise to three cases: (i) all the circles intersect, so there are six intersection points, (ii) not all the circles intersect, so there are four intersection points, and (iii) only two circles intersect.

In the latter case, localization is not possible, while in the first two cases, intersections points are processed in the same way during step two of the algorithm. First, the set of the three or two closest points belonging to different intersections is created. Second, the estimated position is evaluated as the center of mass of the three or two points. Figure 11 exactly shows the result when all the time-of-flight-radius spheres intersect. The grey triangle is created by the

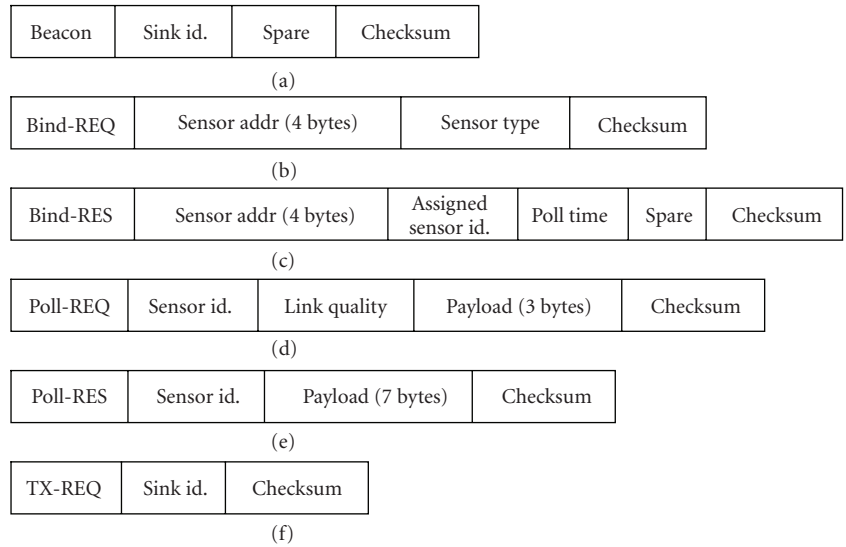


Figure 9: Packet types used within the WSN: (a) beacon packet, (b)-(c) binding packets, (d)-(e) polling packets, (f) data transmission request packet.

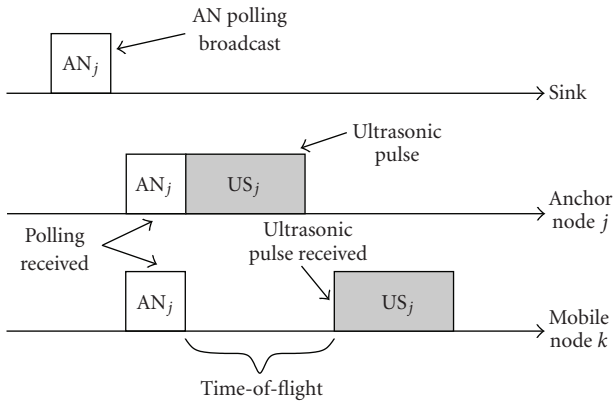


Figure 10: Polling packet sent by the sink to the jth AN and used by the kth MN as a reference time to calculate ToF.

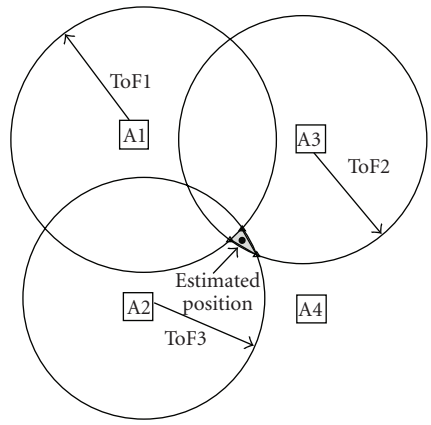


Figure 11: Example of trilateration by using three anchor nodes (AN$_j$). Position is fixed (black circle) in the center of mass of the triangle determined by the intersection points (small triangles) of the time-of-flight-radius (ToF$_j$) circles.

three closest points of the intersecting circles (marked with triangles) and the estimated position is its center of mass (black dot).

3. EXPERIMENTAL RESULTS

The test activity had two main objectives. The first one aims at evaluating the actual characteristics of the radio-modems adopted for our WSN. Specifically, the goal was to better characterize the coverage area and interference immunity offered by radio devices under different operating conditions. The second group of tests was carried out in order to estimate the accuracy in determining MNs' position achievable with the adopted localization algorithm. Concerning the coverage area and interference immunity, a number of tests were performed on a floor of a building, with several offices and laboratories.

To estimate the coverage area of the radio-transceiver, a set of measurements has been collected on a floor of the building housing our department. The floor is formed by two rows of contiguous offices and laboratories, separated by a passageway. In particular, walls between rooms are built with reinforced concrete, while walls between the passageway and rooms are glass panels: the environment can be considered as quasiopen space. The sink was placed in the middle point of the passageway.

When packets are transmitted by using the maximum power (15 dBm), the coverage areaextends to about 25 m in every direction from the node, and allows to cover the entire floor using a single sink. It has been measured that 95% of the floor area is covered also by transmitting at 6 dBm, which allows to save batteries. Furthermore, long-distance measurements, performed under a quasifree space condition, highlighted a maximum transmission range of about 500 m. This proves that, within an open space (e.g., a storeroom,

Table 1: Received packet percentage and incorrect packet percentage versus different power levels for two devices in the same room. Total packets transmitted: 5000.

Power level [dBm]	−14	−10	−6	−1	6	10	13	15
Received	97.80%	99.44%	98.75%	98.84%	100%	100%	100%	99.96%
Incorrect	4.67%	2.51%	3.31%	0.96%	0.08%	0.00%	0.25%	0.51%

Table 2: Received packet percentage and incorrect packet percentage versus different power levels for two devices in near rooms. Total packets transmitted: 5000.

Power level [dBm]	−14	−10	−6	−1	6	10	13	15
Received	27.02%	54.52%	62.76%	72.80%	76.76%	99.67%	89.70%	77.93%
Incorrect	25.05%	32.65%	14.74%	9.00%	3.43%	1.69%	0.52%	0.33%

a hangar), a node may be about 150–200 m away from the sink, therefore, as long as the sink is suitably placed, the WSN may cover an area of about 25000 m^2.

From the percentages reported in Tables 1, 2, and 3, it can be noted that for some high power levels there is a decrease in the number of received packets: this effect is caused by interferences owing to stronger reflective and scattering effects of walls and objects. The above-considered multipath and shadowing effects are the only ones that generate interference. Increasing the number of nodes has no effects on the number of correctly received packets, since the multiple access scheme is based on polling.

The interference immunity was evaluated by studying the capability of rejecting cochannel and adjacent-channels interferences. The tests were carried out with the help of another radio-transceiver, suitably programmed in order to generate signals at the same frequency or on an adjacent channel. Cochannel tests were performed using different pseudocodes, while during adjacent-channel tests, the main and the interfering signal exploit the same pseudocode. In both cases, a good interference immunity has been proved. Table 4 reports the number of correctly received packets when an interfering signal is present on the same channel, but with a different pseudonoise spreading sequence. Moreover, the effects of an interfering signal on an adjacent channel, with the same pseudonoise spreading sequence, are summarized in Table 5. Both sets of results confirm a good level of immunity to interferences.

Finally, to evaluate the performance of the ultrasonic localization subsystem, a reduced testbed has been adopted. Four ANs have been attached to the ceiling of our laboratory and one MN has been moved inside the area below them. Figure 12 shows the position estimation of a still MN; Figure 13 shows instead the tracking of an MN moving along a path. The position of a motionless node falls in an area of about 1 cm^2: this is due to random ToF measurements errors. The tracking of an MN moving along a path shows greater errors, which are however always less than 20 cm.

Owing to the narrow beams of the ultrasonic emission devices and receivers, ceiling height and the height of the MNs affect the ultrasonic coverage area of each AN. Specifically, with a ceiling height of about 3 m, the coverage area of each installed AN is approximately a circle of 2.5 m radius.

4. SIMULATION RESULTS

To evaluate the performance of the localization system in a working WSN based on the architecture described in the previous sections, some simulations have been conducted. The simulation program has been written in Matlab and the most important parameters involved in simulations are listed in Table 6. Two different strategies can be used to poll ANs.

In the "raster scan" strategy all ANs are polled at every cycle using a raster scheme (i.e., scanning them row by row) before switching to the polling phase dedicated to collect data from mobiles. With the "smart scan" strategy, only those ANs are polled that are in the neighborhood of the previously estimated positions of the MNs. All other anchor nodes are not polled, so the time elapsed between two localizations is reduced. This approach works well when in the WSN field area there are few MNs or when they cluster in small regions. The improvement of the "smart scan" approach vanishes when MNs are equally distributed within the controlled area. To take into account new MN nodes that enter the system, a "raster scan" polling is periodically performed to discover the positions of new nodes.

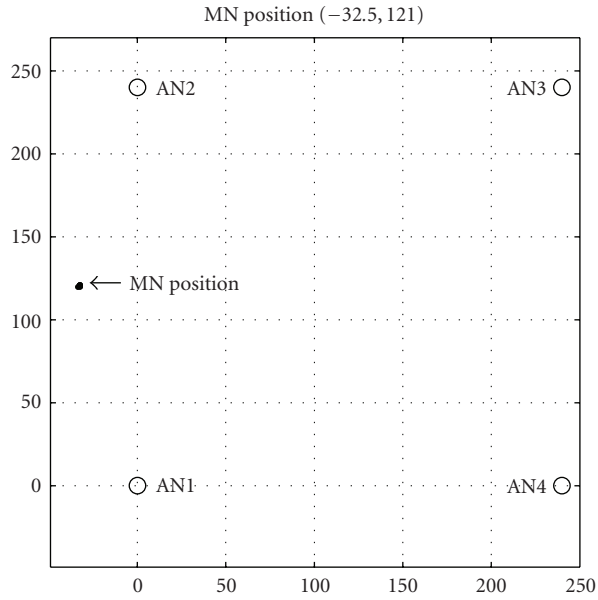
Figure 14 shows the tracking of 6 MNs moving at different speeds within the area covered by the localization system.

Figures 15, 16, 17, and 18 and Tables 7–10 show the tracking errors and their statistics for two kinds of simulations: 6 MNs all moving at 1 m/s and 6 MNs moving at {0.1 0.5 0.9 1.25 1.6 2} m/s. As can be seen in the figures, the error is always less than 0.2 m (only two points go over this value).

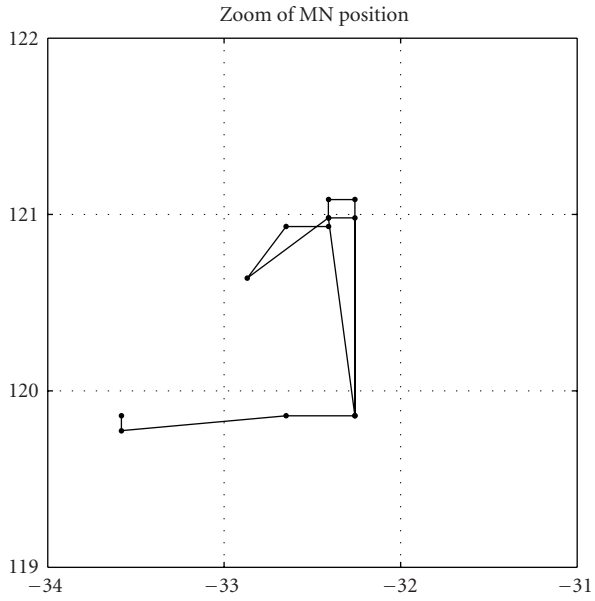
When a simulation with all nodes moving at equal speed is considered, Tables 7 and 8 show that using the "smart scan" strategy errors are a bit lower than when the "raster scan" strategy is used. Furthermore, within the same simulation time, we get about 60 localizations by using the "smart scan" strategy, while 33 localizations are performed by using only the "raster scan" strategy. The advantage of the "smart scan" polling is that the system can perform a more precise tracking

Table 3: Received packet percentage and incorrect packet percentage versus different power levels for two devices in far-away rooms. Total packets transmitted: 5000.

Power level [dBm]	−14	−10	−6	−1	6	10	13	15
Received	12.16%	23.70%	62.50%	60.30%	79.31%	87.93%	96.64%	99.18%
Incorrect	21.32%	16.54%	3.10%	33.17%	7.20%	4.86%	8.30%	1.35%



(a)



(b)

Figure 12: Localization of a still MN. Zoom shows that estimated positions are concentrated in an area of almost 1 cm^2. All measures are expressed in centimetres.

Table 4: Received packet percentage and incorrect packet percentage versus different power levels for two devices in the same room communicating on the same channel with different pseudonoise codes. Total packets transmitted: 5000.

Power level [dBm]	−14	6	15
Received	98.03%	98.71%	99.63%
Incorrect	12.63%	10.35%	5.29%

Table 5: Received packet percentage and incorrect packet percentage versus different power levels for two devices in the same room communicating on adjacent channels with the same pseudonoise code. Total packets transmitted: 5000.

Power level [dBm]	−14	6	15
Received	91.76%	99.78%	99.25%
Incorrect	6.40%	2.38%	5.16%

Table 6: Most relevant parameters involved in simulations.

Parameter	Description
WSN field area	467 m^2
Number of cells	Variable
Number of MNs	Variable
Speed of MNs	Variable
Type of movement	Random way-point
Polling strategy	Raster or smart scan
Noise	Only MN movement is considered
Simulation time	60 seconds

Table 7: Error statistics for 6 MNs moving at 1 m/s using "raster scan" polling.

MN #	Mean [m]	Variance [m^2]
1	0.0426	$8.9156e-4$
2	0.0349	$6.1449e-4$
3	0.0282	$3.4996e-4$
4	0.0395	$7.2375e-4$
5	0.0389	$1.0e-3$
6	0.0401	$6.7607e-4$

of MNs, because it spends less time during the AN polling phase.

Considering now the simulation with MNs moving at different speed, we can see that we get very small errors for

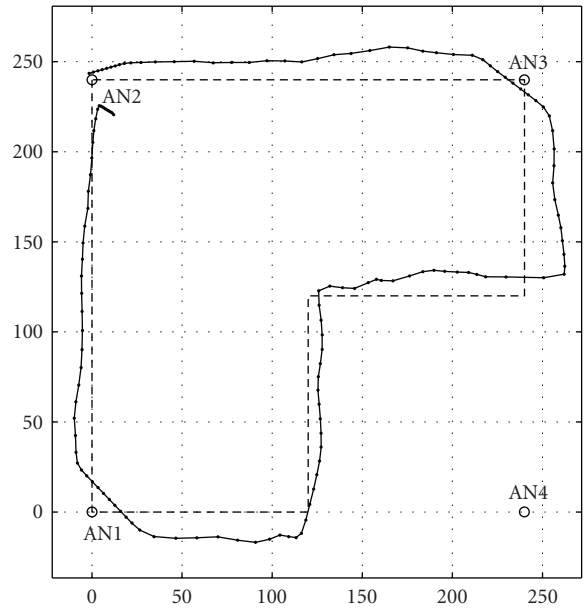


Figure 13: Tracking of an MN (solid line) following a path (dashed line). Maximum error is about 20 cm. All measures are expressed in centimeters.

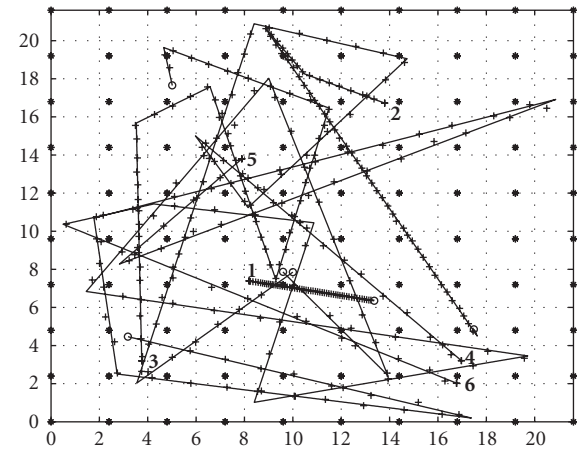


Figure 14: Example of tracking: 6 MNs moving at {0.1 0.5 0.9 1.25 1.6 2} m/s using "smart scan" polling. Each trajectory is labeled with MN number. Stars (∗) are the ANs, crosses (+) are the MNs' estimated positions. The errors of this simulation are shown in Figure 18 and Table 10.

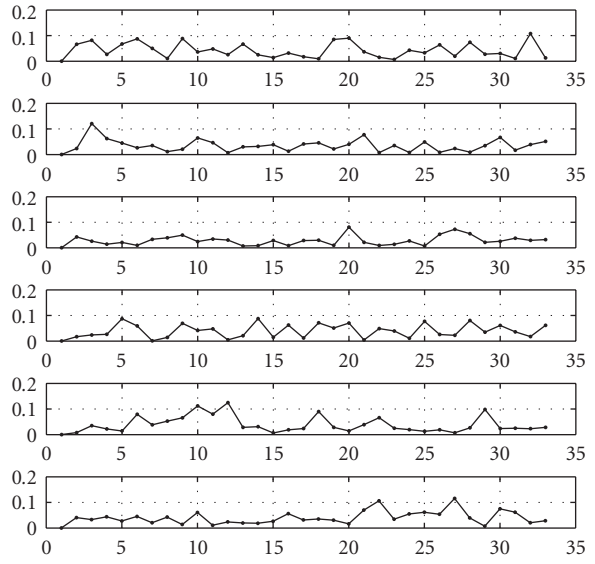


Figure 15: Tracking errors [m] versus MN estimated positions: 6 MNs moving at 1 m/s using "raster scan" polling. The statistics of this simulation are listed in Table 7.

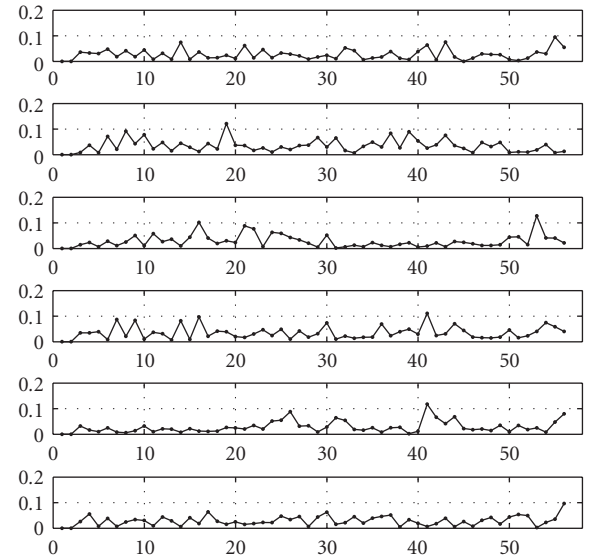


Figure 16: Tracking errors [m] versus MN estimated positions: 6 MNs moving at 1 m/s using "smart scan" polling. The statistics of this simulation are listed in Table 8.

Table 8: Error statistics for 6 MNs moving at 1 m/s using "smart scan" polling.

MN #	Mean [m]	Variance [m^2]
1	0.0270	$4.3186e-4$
2	0.0351	$6.5857e-4$
3	0.0289	$6.5972e-4$
4	0.0355	$6.3438e-4$
5	0.0279	$5.3628e-4$
6	0.0289	$3.6381e-4$

nodes moving slowly, while errors slightly increase for nodes moving faster.

Considering both simulations and real experiments it must be highlighted that refresh rate limits are due to the physical system used to perform localization (i.e., ultrasounds) and not to the communication or channel access protocol. Anchors cannot be polled faster than the time needed by the ultrasonic pulse to travel from the AN to the farthest point of its coverage area. Performance can be probably improved further by polling ANs in smarter ways

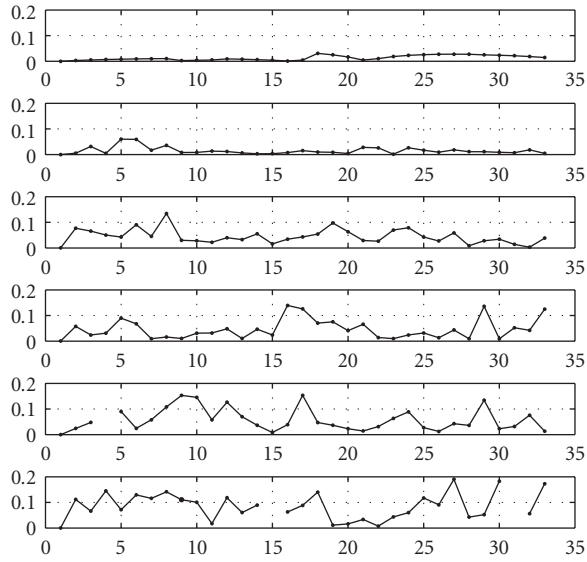


Figure 17: Tracking errors [m] versus MN estimated positions: 6 MNs moving at {0.1 0.5 0.9 1.25 1.6 2} m/s using "raster scan" polling. The statistics of this simulation are listed in Table 9.

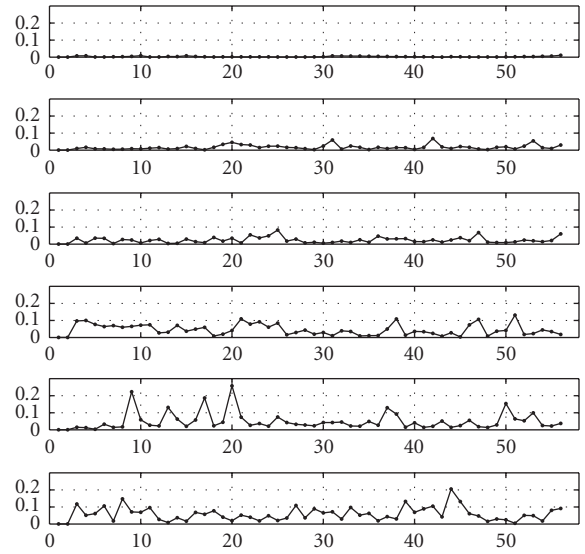


Figure 18: Tracking errors [m] versus MN estimated positions: 6 MNs moving at {0.1 0.5 0.9 1.25 1.6 2} m/s using "smart scan" polling. The statistics of this simulation are listed in Table 10.

Table 9: Error statistics for 6 MNs moving at {0.1 0.5 0.9 1.25 1.6 2} m/s using "raster scan" polling.

MN #	Mean [m]	Variance [m^2]
1	0.0131	$9.0023e-5$
2	0.0151	$2.0972e-4$
3	0.0448	$8.2210e-4$
4	0.0462	$1.5e-3$
5	0.0576	$2.0e-4$
6	0.0853	$2.8e-4$

Table 10: Error statistics for 6 MNs moving at {0.1 0.5 0.9 1.25 1.6 2} m/s using "smart scan" polling.

MN #	Mean [m]	Variance [m^2]
1	0.0029	$6.1304e-006$
2	0.0165	$1.9648e-004$
3	0.0228	$2.9222e-004$
4	0.0452	$1.1e-3$
5	0.0498	$2.8e-4$
6	0.0578	$1.7e-4$

(e.g., more than one, but not adjacent, AN at the same time), but a lower limit to the polling rate always exists.

5. CONCLUSIONS

The paper presented an original WSN, designed and implemented for facing localization needs in an indoor environment. The network operates according to a single-hop transmission scheme: a sink coordinates all the anchor nodes and mobile nodes to localize the latter; furthermore, other nodes with different "identities" (e.g., to measure temperature) are handled by the network.

Measurements have proved that, by using the hardware and the algorithms described here, a sink can control an open-space area of about 25000 m^2; moreover, the precision achieved by the ultrasonic localization subsystem is about 2 cm for still MNs and 15 cm for mobile ones.

The precision obtained by measurements is very close to the one provided by simulation: the mean errors along specific paths are between 2 and 5 cm, while the absolute maximum error is limited to about 20 cm. Furthermore, simulations proved that the precision of the system can be slightly improved by adopting a smarter polling strategy. A lower limit for polling rate always exists and is caused by the propagation speed of ultrasonic pulses.

A possible evolution we are considering consists of using a multihop protocol to organize the network in a hierarchical framework. In this way, data coming from sinks are collected by a higher level where nodes act as data concentrators. This approach can be repeated in a pyramidal way to deliver data to a unique control point.

ACKNOWLEDGMENTS

This work was partially supported by the Italian Ministry of Education and Research (MIUR), within the framework of the AIBER project. The paper is an extended version of an invited contribution presented at IMMERSCOM 2007, Bussolengo (Verona), Italy.

REFERENCES

[1] M. Y. Ibrahim and A. Fernandes, "Study on mobile robot navigation techniques," in *Proceedings of the IEEE International Conference on Industrial Technology (ICIT '04)*, vol. 1, pp. 230–236, Hammamet, Tunisia, December 2004.

[2] D. P. Stormont and A. Kutiyanawala, "Localization using triangulation in swarms of autonomous rescue robots," in *Proceedings of the IEEE International Workshop on Safety, Security and Rescue Robotics (SSRR '07)*, pp. 1–6, Rome, Italy, September 2007.

[3] H. H.-H. Lin, C. C.-C. Tsai, and J. J.-C. Hsu, "Ultrasonic localization and pose tracking of an autonomous mobile robot via fuzzy adaptive extended information filtering," to appear in *IEEE Transactions on Instrumentation and Measurement*.

[4] M. Gröhn, T. Lokki, and L. Savioja, "Using binaural hearing for localization in multimodal virtual environments," in *Proceedings of the 17th International Congress on Acoustics (ICA '01)*, vol. 4, Rome, Italy, September 2001.

[5] J. Breebaart, J. Herre, L. Villemoes, et al., "Multi-channel goes mobile: MPEG surround binaural rendering," in *Proceedings of the 29th International Audio Engineering Society Conference (AES '06)*, pp. 1–13, Seoul, Korea, September 2006.

[6] M. Gröhn, T. Lokki, L. Savioja, and T. Takala, "Some aspects of role of audio in immersive visualization," in *Visual Data Exploration and Analysis VIII*, vol. 4302 of *Proceedings of SPIE*, pp. 13–22, San Jose, Calif, USA, January 2001.

[7] Y. Nakazato, M. Kanbara, and N. Yokoya, "Discreet markers for user localization," in *Proceedings of the 8th IEEE International Symposium on Wearable Computers (ISWC '04)*, vol. 1, pp. 172–173, Arlington, Va, USA, October-November 2004.

[8] R. Singh, L. Macchi, and C. S. Regazzoni, "A statistical modeling versus geometrical determination location approach for static positioning in indoor environment," in *Proceedings of the International Symposium on Wireless Personal Multimedia Communications (WPMC '05)*, Aalborg, Denmark, September 2005.

[9] E. Elnahrawy, L. Xiaoyan, and R. P. Martin, "The limits of localization using signal strength: a comparative study," in *Proceedings of the 1st Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (SECON '04)*, pp. 406–414, Santa Clara, Calif, USA, October 2004.

[10] J. Xu, M. Ma, and C. L. Law, "Position estimation using ultra-wideband time difference of arrival measurements," *IET Science, Measurement & Technology*, vol. 2, no. 1, pp. 53–58, 2008.

[11] A. Boukerche, H. A. B. de Oliveira, E. F. Nakamura, and A. A. F. Loureiro, "Towards an integrated solution for node localization and data routing in sensor networks," in *Proceedings of the 12th IEEE Symposium on Computers and Communications (ISCC '07)*, pp. 449–454, Aveiro, Portugal, July 2007.

[12] http://www.alldatasheet.com/datasheet-pdf/pdf/109543/ETC/400ST160.html.

[13] http://www.microchip.com/.

[14] Cypress, "CYWUSB6935: WirelessUSB TM LS 2.4 GHz DSSS Radio SoC," Datasheet, http://www.cypress.com.

[15] http://www.aurelwireless.com/.

[16] http://www.embeddedarm.com/.

[17] N. Priyantha, A. Chakraborty, and H. Balakrishnan, "The Cricket location-support system," in *Proceedings of the 6th ACM/IEEE International Conference on Mobile Computing and Networking*, pp. 32–43, Boston, Mass, USA, August 2000.

[18] A. Mandal, C. V. Lopes, T. Givargis, A. Haghighat, R. Jurdak, and P. Baldi, "Beep: 3D indoor positioning using audible sound," in *Proceedings of the 2nd IEEE Consumer Communications and Networking Conference (CCNC '05)*, pp. 348–353, Las Vegas, Nev, USA, January 2005.

Hindawi Publishing Corporation
Advances in Multimedia
Volume 2008, Article ID 426981, 11 pages
doi:10.1155/2008/426981

Research Article

Remote Laboratory Experiments in a Virtual Immersive Learning Environment

Luca Berruti,[1] Franco Davoli,[2] Sandro Zappatore,[2] Gianluca Massei,[3] and Amedeo Scarpiello[3]

[1] *Communications Laboratory, University of Genoa Research Unit, University of Genoa, National Inter-University Consortium for Telecommunications (CNIT), Savona Multimedia, Via Armando Magliotto 2, 17100 Savona, Italy*
[2] *Department of Communications, Computer and Systems Science (DIST), University of Genoa Research Unit, National Inter-University Consortium for Telecommunications (CNIT), Via Opera Pia 13, 16154 Genova, Italy*
[3] *National Laboratory for Multimedia Communications, Complesso Universitario Monte Sant'Angelo, National Inter-University Consortium for Telecommunications (CNIT), Via Cinthia, Edificio Centri Comuni, 80126 Napoli, Italy*

Correspondence should be addressed to Gianluca Massei, gianluca.massei@cnit.it

Received 7 March 2008; Accepted 3 November 2008

Recommended by Ghassan Alregib

The *Virtual Immersive Learning* (VIL) test bench implements a virtual collaborative immersive environment, capable of integrating natural contexts and typical gestures, which may occur during traditional lectures, enhanced with advanced experimental sessions. The system architecture is described, along with the motivations, and the most significant choices, both hardware and software, adopted for its implementation. The novelty of the approach essentially relies on its capability of embedding functionalities that stem from various research results (mainly carried out within the VICOM national project), and "putting the pieces together" in a well-integrated framework. These features, along with its high portability, good flexibility, and, above all, low cost, make this approach appropriate for educational and training purposes, mainly concerning measurements on telecommunication systems, at universities and research centers, as well as enterprises. Moreover, the methodology can be employed for remote access to and sharing of costly measurement equipment in many different activities. The immersive characteristics of the framework are illustrated, along with performance measurements related to a specific application.

1. INTRODUCTION

Virtual Immersive Communications (VICOM) is a national project funded by the Italian Ministry of Education, University and Research (MIUR), started in November 2002 and ended in May 2006 (http://www.vicom-project.it). The project goal has been the design of a communication system's architecture able to provide mobile virtual immersive services. The architectural framework and its functionalities have been demonstrated with two service test benches, denoted as *Mobility in Immersive Environment* (MIE) and *Virtual Immersive Learning* (VIL), respectively. In particular, the VIL test bench implements a virtual collaborative immersive environment, capable of integrating natural contexts and typical gestures, which may occur during traditional lectures, enhanced with advanced experimental sessions. Two training courses have been realized: the first one was oriented to virtual restoration of paintings, whereas the second one concerned e-measurement applications, enabling students to remotely control real devices and instrumentation, located at the National Laboratory for Multimedia Communications in Naples, Italy, and at WiLab in Bologna, Italy, respectively.

A 3D virtual reality application allows the real-time interaction between a lecturer or instructor and students, who are not physically present in the same classroom. Students were grouped inside a number of well-equipped classrooms, interconnected through an IP network.

Traditional approaches to Virtual Reality (VR) are based on complex and relatively expensive devices, such as head-mounted displays (HMDs), data gloves, and CAVE systems [1]. Instead, the proposed approach to realize the VIL test bench has leveraged results that were the output of research activities related to specific work packages of the VICOM project. In particular, VIL exploits audio and video processing algorithms to realize an immersive interaction with the virtual class, a specific database to share and

manage all context information, a multimedia board and an embedded haptic interface to show different approaches to virtual reality applications, hardware/software architectures specifically designed and realized to control real measurement instruments and devices (which may also be placed in different laboratories), and virtual restoration tools to improve the quality of digital reproductions of paintings.

Accessing remote laboratory instrumentation and performing experiments, either individually or under the supervision of an instructor, have become key elements in distance learning and training, not only in technical disciplines. So, the layout and the output of a demo laboratory session on a telecommunication measurement experiment (interference generation and control over a wireless LAN) are also described.

The paper is organized as follows. In Section 2 the required hardware components are illustrated, while in Section 3 the software system architecture is presented. Section 4 summarizes some performance results of the e-measurement software architecture, also in comparison with commercial solutions. Finally, Sections 5 and 6 discuss an operative example and user mobility issues, respectively, while in the last section conclusions are drawn.

2. HARDWARE COMPONENTS

In the VIL scenario, the generic user reaches a VIL real classroom and logs in to the system through an accounting phase, to define the user's profile and know the seat reserved. Then, the lecturer and students enter the virtual classroom, where they are represented by their avatars, and reach their own virtual workspace. So, the real-time lecture takes place in a virtual context-aware environment, where interactions occur in a natural way, by means of scene analysis systems and immersive input devices. Finally, lectures are complemented with experimental laboratory sessions, oriented to supervised telerestoration and cooperative telemeasurements, exploiting specialized virtual laboratory software.

The proposed scenario has been realized in order to be compliant with the economical resources of the VICOM project. To this aim, all useful research results from project work packages have been embedded into the system, rather than relying on very expensive hardware available on the market for data acquisition and visualization in immersive environments.

The fulfillment of the VIL goals has required the specification and the acquisition of the equipment of some enhanced classrooms, through which lecturers and students can take part in the immersive lecture. These classrooms were interconnected through the CNIT national network, mainly based on a satellite platform (DVB-RCS-like [2]), allowing the bidirectional interconnection of a large part of CNIT research units and laboratories. The network, provided by Eutelsat, operated in Skyplex technology over the Ka band (HotBird6 Satellite) [3], by providing an overall satellite bandwidth of 2 Mbps, shared among the active earth stations. In particular, such network connected some CNIT and CNR (National Research Council) laboratories in Naples, Bologna, Florence, Genoa, and Pisa, (Italy) which have taken part in the development of the VIL test bench.

Since different types of enhanced classrooms are possible, each center can choose the specific test bench components to highlight. A fully equipped classroom would include the hardware components explained in the following, to list all significant functionalities.

(i) *Video rendering systems*. For the students' class, we have selected a visualization system composed by a projection screen, two linear polarization filters, two XGA projectors, and passive glasses (see Figure 1). An autostereoscopic display is used for the lecturer. Both systems must be equipped with a professional graphics workstation.

(ii) *Audio rendering systems.* For the students' class we have chosen wireless headphones, while normal loudspeakers are sufficient for the lecturer.

(iii) *Input devices* (see Figure 2). Any user can interact with the Graphical User Interface (GUI) through input devices providing different immersion sensations. The user can choose a simple mouse, a 3D mouse with six degrees of freedom, a haptic interface (provided by the PERCRO laboratories of Pontedera, Italy), or a multimedia board (provided by the CNIT research unit at the University of Florence, Italy).

(iv) *Contribution devices*. During the lectures or the laboratory experiments, audio and video interaction of any user must be allowed. For the students' class we have selected a Pan-Tilt-Zoom (PTZ) dome camera (whose control is allowed via VISCA commands) and omnidirectional microphones, while simple commercial devices are sufficient for the lecturer. Video System Control Architecture (VISCA) is a network protocol designed to interface a wide variety of video devices to a computer.

(v) *Scene analysis systems.* These systems allow the acquisition and analysis of context information. They need an accurate tuning to overcome the environment problems (room size, light, noise level, reverberation, etc.). In particular, the *Audi location system*, provided by the research unit at the Technical University of Milan [4], allows locating the position of the speaker making a reservation, through the phase processing of the acquired audio signals (it includes an array of microphones, audio mixer, computer for the processing, and deadening panels), while the *Request Identification System*, provided by the CNIT research unit at the University of Genoa [5], allows making a reservation for a question or intervention simply by raising a hand, by means of video processing techniques (it includes dome camera and a computer for processing). Finally, a specific application, developed by the CNIT research unit at the University of Cagliari, is able to control the PTZ dome camera to transmit the video of the student making a reservation.

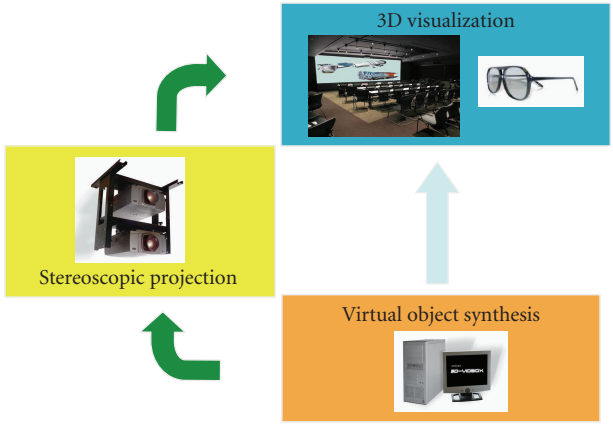


FIGURE 1: Video rendering system for the students' class.

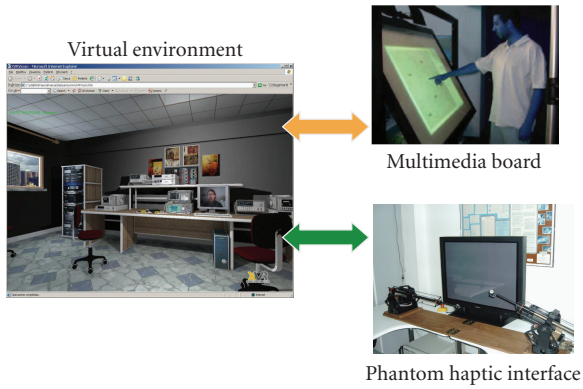


FIGURE 2: Input devices.

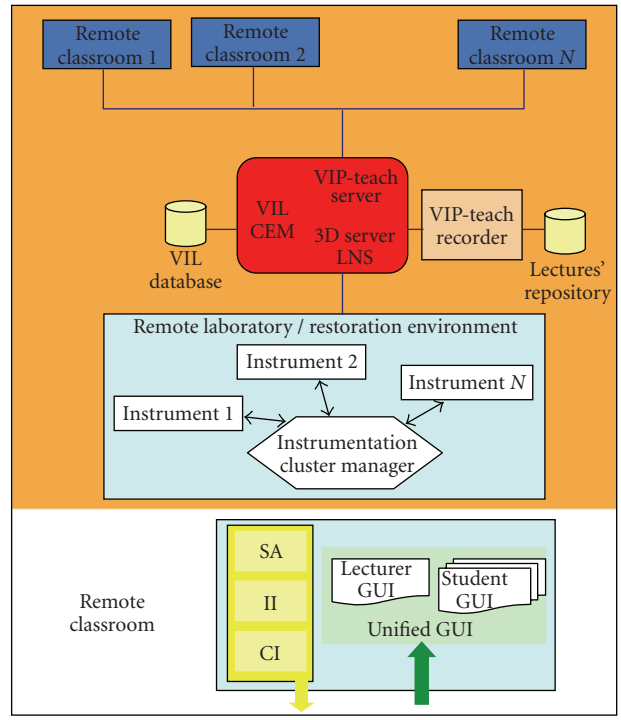


FIGURE 3: Main blocks of the software architecture.

3. SOFTWARE ARCHITECTURE

The software architecture is illustrated in Figure 3. The *common experience manager* (CEM) is certainly the main block of such architecture, as it manages both e-learning and experimental laboratory sessions. Context is captured and analyzed by the *scene analysis* (SA) module, through arrays of microphones and cameras. Such information is stored in the *VIL database* and managed by a Java interface.

Any student can select a synchronous or asynchronous instruction course. In the former case, the CEM manages the interaction between students and lecturer through a token-based mechanism: the lecturer is able to entirely release or to share its privileges, communicating with the CEM through an immersive Graphical User Interface (GUI). *Interactive inputs* (II) allow interaction with the virtual environment, while *contribution inputs* (CI) permit to ask questions during a lecture, after being enabled by the lecturer: interventions occur by video and audio streaming. In the latter case, a student can download a previous lecture stored in the *Lectures' Repository* by means of the *video communications over IP (VIP)-teach recorder* and visualize these offline contents by using a specific player.

Finally, the *LabNet server* (LNS) and the *instrumentation cluster manager* (also named *experience manager*) provide the remote control of real laboratory instrumentation, as presented in Section 3.2.

3.1. Graphical user interfaces

A new immersive GUI has been developed to support 3D contents in the synchronous e-learning application *VIP-Teach*, provided by LightComm (http://www.lightcomm.it). The components of this GUI (see Figure 4) are video (MPEG4 codec), chat, ppt presentations, 3D space, and management window (with the list of students online and of those making a reservation). In particular, 3D contents in the lecture session are realized in *Virtual Reality Modeling Language* (VRML) and controlled by Java applications to obtain highly interactive and immersive worlds, whose behavior is modified by user actions [6] in real-time. 3D Studio Max, VIZ and Maya, among others, can be used to generate and export nonelementary environments in VRML files format. They in fact allow navigation in the 3D environment, management of collisions among 3D objects, visualization of the avatars of other users moving in the environment, visualization of reservation events and information about users, search of an avatar by name and selection of a laboratory session.

During the lecture, the lecturer can select a laboratory session, simply by clicking on a virtual door present in the scene. 3D contents in the laboratory sessions (see

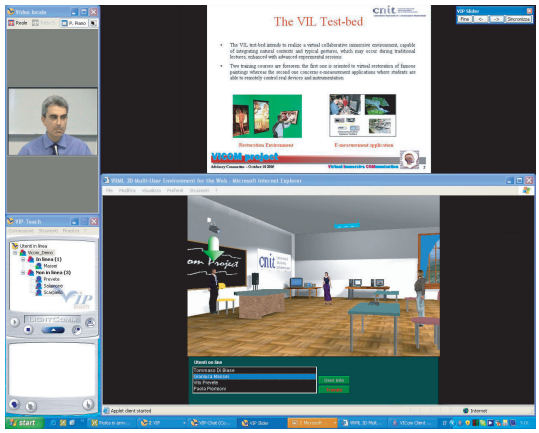


Figure 4: GUI in the lecture session.

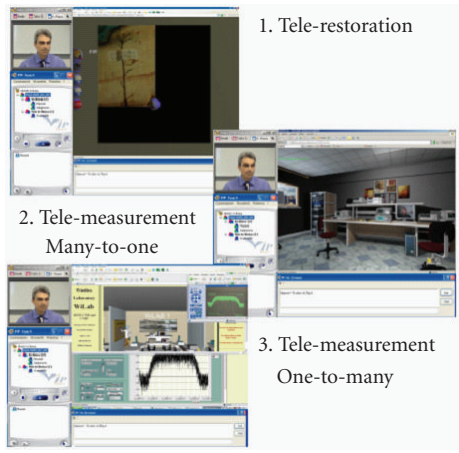


Figure 5: GUI in the laboratory sessions.

Figure 5) are modeled through 3D Studio Max and controlled through *eXtreme Virtual Reality* (XVR) by VRMedia (http://www.vrmedia.it/).

The *telerestoration session*, realized by the CNIT research units at the Universities of Florence and Pisa S. Anna, allows experimenting virtual restoration techniques (such as crack removal and lacuna filling) on high-resolution digital copies of famous paintings [7], while the two *telemeasurement sessions* permit to interact with real instrumentation. In the *many-to-one paradigm*, developed in the CNIT National Laboratory for Multimedia Communications (Naples) [8], the experience is collaborative, namely, the GUI interface allows the lecturer to transfer the experiment's control to the students, while in the *one-to-many paradigm*, realized at the WiLab laboratories (Bologna) [9], it is possible to interact with a "measurement chain," whose instrumentation is geographically distributed in different locations.

As concerns telerestoration, the devised tool aims at obtaining a digital version of the artwork where all damages have been removed; the great advantage is that if a mistake was made, the artwork does not suffer any kind of injury, and the virtual restorer can start again the restoration process. This can be useful for educational aims, in order to look at the artwork as it was in the intent of the artist who made it, and for guidance aims, in order to give the actual restorer the possibility to perform some useful tests before choosing the best restoration technique. The telerestoration session [10] permits to download high-quality digital images in bitmap format, to zoom in the images, and to restore a crack and a lacuna according to the techniques actually used during restorations. Indeed, cracks and lacunas are two of the main problems a painting or a fresco can be affected by. They deteriorate the artworks more or less significantly depending on their number and their severity.

The telerestoration session is able to remove cracks in a semiautomatic way, as it requires the aid of a human user, who has to select one of the pixels belonging to the crack; the reason for this is that only an observer can decide if a dark line is a crack or it belongs to the texture or the subject of the artwork. So suitably initialized, the restoration automatic procedure is able to recover the whole crack by means of an interpolation technique.

Lacunas occur when some parts of the artwork collapse and fall down, resulting in a lack of paint. The telerestoration session operates by repainting the parts that have collapsed according to some restoration methods, such as *chromatic selection, chromatic abstraction, rigatino*, and *pointellism*. Their aim is to fill in the lacuna, so as to recover the coarse uniformity of the artwork and avoid the presence of annoying holes in the whole image.

As regards the telemeasurement system, the virtual instruments in the many-to-one paradigm represent the laboratory "active elements," in the sense that knobs, buttons, and displays present on their front panels can be dynamically controlled by the users or updated on the basis of measurement results. These active elements are handled by Java applets (running within the framework of an XVR application), which communicate with the server-side infrastructure in order to exchange commands, data, and results to/from the real remote instrumentation.

XVR, by means of which all laboratory sessions have been represented, is an integrated environment for the rapid development of Virtual Reality applications. XVR is structured in two main modules: the ActiveX control module, which hosts the very basic components of the technology (like the versioning check and the plugin interfaces), and the XVR Virtual Machine (VM) module, which contains the core of the technology (such as the 3D graphics engine, the multimedia engine, and all the software modules managing the other built-in XVR features).

XVR features include: client plugin as an ActiveX control for Internet Explorer, import of models from 3DSMax 4.0 or higher, advanced OpenGL rendering engine, dedicated script language (S3D), vertex and pixel shaders' support, supplied byte-code compiler, run-time expandable module capabilities, HTML pages interaction using JavaScript or VBScript, video textures supporting AVI, import of FLASH images as 3D textures. Supported audio formats include WAV, MIDI, MP3, and WMA; other features are positional 3D audio support, input devices' management, remote connections support (TCP and UDP management).

3.2. Server-side architecture

The main components of the CEM are the *VIP-Teach server*, the *LabNet server*, and the *3D server*, as shown in Figure 6.

The *VIP-Teach server* is able to manage users' accounts and permissions, enrol the students in the lectures, and activate the PowerPoint viewer on the remote PCs. It can be followed by a web portal, for the management of the lectures' calendar and for the offline diffusion of ppt presentations, and by a *recorder* that allows recording the lecture.

The *LabNet server* [8], an ad hoc supervising central unit (SCU), manages access to a generic experiment, guaranteeing interoperability and synchronization among users. Particularly, owing to a *control* module, it makes the experience collaborative, allowing a super user (the lecturer) the possibility to pass the instrumentation control (token) to users of inferior level (the students), through the VIP-Teach client interface. Besides, owing to the *data provision* module, the instrument data are distributed to users in multicast fashion, and can be visualized on the 3D interface, via a Java-based adaptation layer.

The *3D manager* (i.e., the main component of the 3D server) is a pure Java application able to manage the *VIL database* and information related to the graphical representation, and to handle authorizations of avatars and the logical structure of the scene.

At the transport layer, the VIP-Teach server adopts UDP for audio/video streams and TCP for session management. TCP is also used by the 3D server. The LabNet server adopts both TCP and UDP, and their use will be specified in more detail below.

The software architecture for e-measurement experiments, developed at the National Laboratory for Multimedia Communications in Naples, is shortly explained in Figure 7, by using a top-down approach. The SW modules involved in the architecture are explained in the following.

(i) The 3D GUI displays the instrument data and communicates with the rest of the architecture via a Java-based interface.

(ii) The LNS (*LabNet server*) manages the access of users to the experiments and distributes the instrument data.

(iii) The *experience manager* manages the allocation of the instruments in the individual experiments, the correspondence among the experiment's variables and actions on the instrumentation drivers.

(iv) The *experience database* contains the experiment table (to list the instruments involved in each one) and instrument table (to define the allocation state).

(v) *Test beds* are the set of instrumentation drivers for e-measurement sessions.

User data communication relies upon UDP, in unicast or multicast fashion. This connectionless communication protocol is light and efficient even on a satellite link, but also unreliable. Therefore, the LNS has to deal with lost

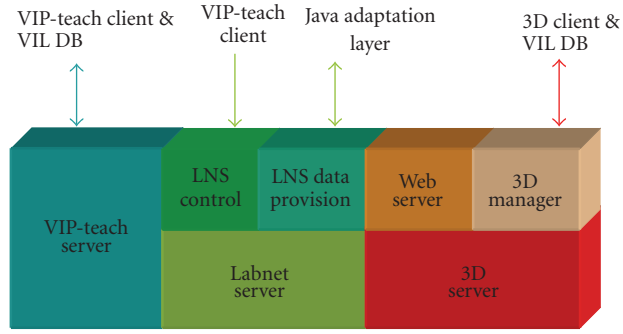


Figure 6: Main components of CEM.

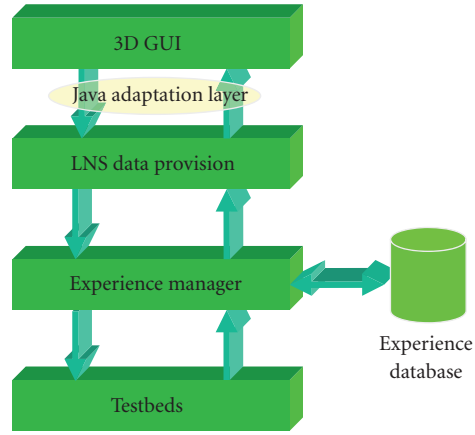


Figure 7: SW architecture for e-measurement session.

packets and quality of service (QoS) problems. Laboratory sessions often involve a large number of user stations, and so multicast transmission should be chosen (wherever it is supported by the network) for a more efficient use of the available bandwidth. On the other hand, for each kind of user, there is a reliable control connection to the server over the TCP communication protocol. It is used both for token exchange and for starting or taking part in an experiment. TCP is heavier than UDP, but it guarantees stability and control of parameters that are critical for the correct working of the system.

The LNS knowledge is limited to the experiments and to their allocation, based on different types of user access, but it does not concern the instruments being used. The experience manager in fact establishes the link between the LNS and the heterogeneous instrumentation world, managing the instruments' allocation and drivers' actions. In particular, to call the driver procedures, the experience manager adopts *remote procedure calls* (RPCs) through *Simple Object Access Protocol* (SOAP), using Extended Markup Language (XML) to encode its calls and HyperText Transfer Protocol (HTTP) as a transport mechanism [11]. The drivers recognize SOAP-RPC messages and translate them into reading/writing commands on the instruments' allocation involved in the experiment.

3.3. Client-side architecture

Figure 8 shows the main components of the remote classroom. In accordance with the GUI, we have considered two main software modules: the *VIP-Teach Client* and the *3D Client*.

The *VIP-Teach Client* provides students and lecturers with the elements needed to actively take part in the lecture; this set of tools includes several audio/video contents and ppt presentations, as well as chat box, management window, and shared board. Furthermore, the *VIP-Teach Client* interacts with VIP-Teach server to manage users' accounts, to receive/transmit the audio and video contents from/to own peers, according to the relative roles, to transfer the information related for token management to the LNS control module, to interact with the VIL database to publish the token holder in the context space, and to extract the reservation data.

The VRML/XVR-based *3D Client* provides context information and creates a 3D immersive representation of the class and instruments involved in the lecture. The VRML/XVR Client interacts with the 3D Manager to log the users and present context information (i.e., user identity, avatar position, students in reservation), with the LNS data-provision module to write and read instrument and painting data via the Java-based adaptation layer, and with the VIL database for data upload/download.

3.4. Context data exchanging

A MySQL DB, named *VIL database* and shown in Figure 9, is used to exchange context data. It consists of 8 tables, regarding both user and environment.

(i) The static tables contain *user* profiles, *authorization, environment settings*, and *experiments' descriptions*.

(ii) The graphical data update is provided by two dynamic tables: *user dialog* (in which any client writes its own data) and *user information* (in which the 3D manager inserts global data to provide the updates to all clients).

(iii) The *Hand UP* table is used by external applications, such as the scene analysis systems and VIP-Teach, to manage the reservation.

(iv) The *location* table is used in order to identify the actual experiment or to change it.

4. PERFORMANCE OF THE E-MEASUREMENT SOFTWARE ARCHITECTURE

The LNS represents the core of the e-measurement software architecture and, in a sense, it can be viewed as middleware providing elements to offer services through a common interface, in order to establish a contact between who asks for a service and who offers it. During its design and implementation, much attention was paid to address several crucial concerns, such as

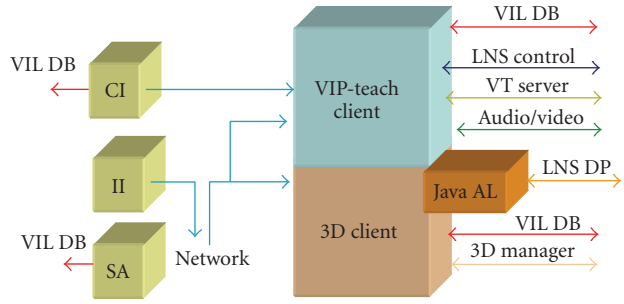


Figure 8: Main components of the remote classroom.

(i) the intrinsic heterogeneity of the application environments and of the instruments;

(ii) the software portability and scalability;

(iii) the level of flexibility (to interact with every kind of equipment in a simple way);

(iv) the capability of multicasting data gathered from the measurement instrumentation for an efficient use of the transmission resources.

All these aspects, although quite relevant, are not sufficiently well focused, and are often neglected, in some products available on the market.

A significant number of tests have been carried out on the LNS, also in comparison with another very popular commercial software package, with two main goals: to evaluate the LNS effectiveness in the presence of channels characterized by high delay-bandwidth products (such as satellite links) and to know the maximum throughput sustainable by the LNS in terms of data dispatching and managing.

4.1. LNS performance on a satellite link

The testing of LNS on a real satellite link [12] aimed at

(i) evaluating the efficiency of the LNS in terms of packet loss and jitter of data packets observed at the receiver end;

(ii) comparing the effectiveness of the proposed software platform with the "data socket server (DSS)" of the LabVIEW suite, a commercial and very popular software package by National Instruments to remotely pilot instrumentation.

The experimental setup that was used for performance evaluation is depicted in Figure 10.

The "variable generator" (VG) plays the role of an experiment manager, producing every D seconds a set of data packets conveying a group of 60 variables (the total net payload amounts to 8400 bytes). Since the variables generated at the VG are the same in both cases, the possible differences in performance can be attributed to the different protocols, data storing, retrieving, and forwarding strategies adopted by the LNS and the DSS. The multicast capability of the LNS was not exploited in these tests, for fairness in

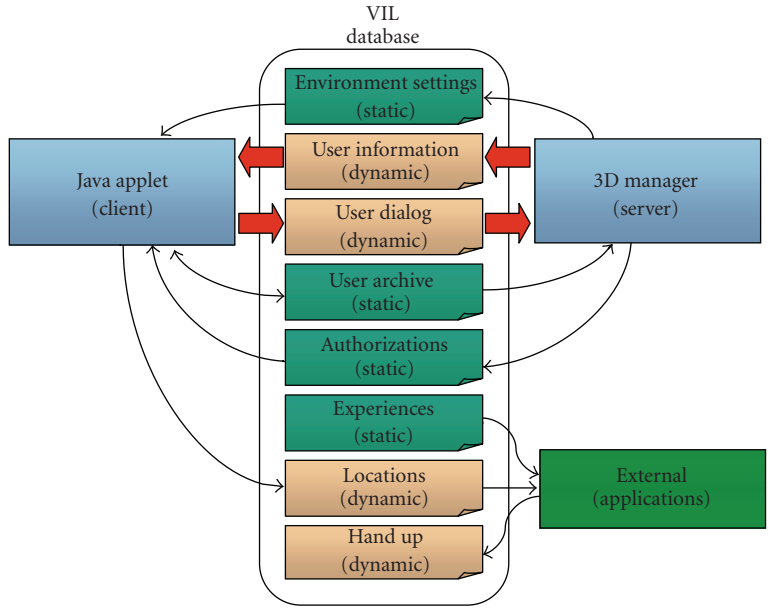


Figure 9: The VIL database.

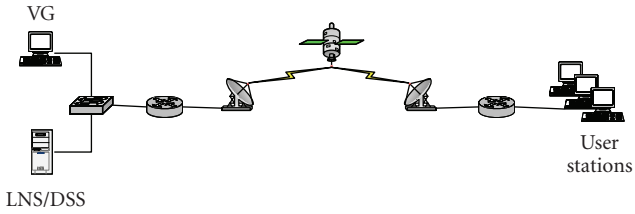


Figure 10: The experimental setup for performance evaluation.

the comparison, as the DSS version used did not support multicast.

Besides the satellite experimental setup, other two quite similar setups have been exploited. In the former, the client stations are connected to the LNS/DSS via a terrestrial link, whose bandwidth amounts to the average capacity measured at the IP layer on the satellite link (1.2 Mbps). In the latter, the client stations are directly connected to the LNS/DSS by means of a high-speed (100 Mbps) LAN, without the presence of routers and satellite links.

Tables 1 and 2 (from [12]) summarize the packet loss and the *root mean square* (RMS) of the delay jitter (i.e., the difference between the expected and the actual variable transit time, namely, the time a variable needs to reach a client since its arrival at the LNS/DSS) versus the timing variable D in LAN, Terrestrial, and satellite scenarios, respectively. The former table shows data related to the LNS while the latter reports data obtained with the DSS. (Whenever the variable losses exceeded 30%, we have preferred to omit the corresponding RMS because there are too few data in order to compute a stable and reliable RMS value and sometimes the DSS itself crashes.)

The results highlight that the LNS performance is almost unaltered in passing from a LAN to a terrestrial link environment, while a satellite link yields higher RMS values. However, also in this latter case, the RMS values never exceed 3% of the timing variable. Moreover, no loss is present for timing variables of 1000, 500, 350 milliseconds. The losses at 300 milliseconds are due to the queue length, inadequate to completely allocate room for the data bursts.

On the contrary, the performance of the DSS dramatically decreases when a satellite link is in use. Comparing the columns of Table 2 related to the terrestrial and satellite links, highlights how the propagation delay, inherent to the satellite link, strongly affects the overall performance of an e-measurement platform centered on the DSS. Furthermore, the DSS appears unable to manage bursts of variables, whose interarrival times are less than 350 milliseconds.

Most likely, the main reason for the different behavior of the LNS and the DSS resides in the transport protocol. The DSS uses TCP as a transport protocol, whose performance may be negatively affected by the presence of a large bandwidth-delay product, whereas the LNS relies on UDP (without any reordering mechanism). However, the adoption at the application level of TCP by the DSS does not assure the absence of loss of variables at the receiver end. This is probably due (the actual DSS working mechanism is undocumented) to the fact that the DSS likely discards the variables arrived too late. On the contrary, although the LNS extensively uses UDP packets to convey information, the UDP lightness and the efficiency of the LNS allow a de facto "reliable" delivery. Obviously, the efficiency drastically increases by enabling the multicast capability owned by the LNS.

4.2. LNS maximum throughput

A second group of tests was carried out aiming at estimating and comparing the maximum throughput sustainable by the

Table 1: Performance results in the presence of the LNS.

Variable time D [ms]	LAN		Terrestrial		Satellite	
	Loss [%]	RMS [μs]	Loss [%]	RMS [μs]	Loss [%]	RMS [μs]
1000	0	70 ± 2	0	139 ± 71	0	16615 ± 70
500	0	72 ± 3	0	170 ± 78	0	15980 ± 240
350	0	75 ± 4	0	258 ± 90	0	11030 ± 530
300	0	71 ± 5	1.6	14141 ± 78	2.3	9120 ± 212

Table 2: Performance results in the presence of the DSS.

Variable time D [ms]	LAN		Terrestrial		Satellite	
	Loss [%]	RMS [μs]	Loss [%]	RMS [μs]	Loss [%]	RMS [μs]
1000	0	16750	0.2	103000	60	—
500	0	14500	25	189000	82	—
350	1.3	24500	60	—	96	—

Table 3: Variables' loss at the user stations versus different loads produced at the VG.

Load produced at the VG	LNS variables' loss	DSS variables' loss
269 kbps	0%	0.2%
340 kbps	0%	9.5%
2100 kbps	0%	33%
15 Mbps	0%	—
20 Mbps	5.24%	—
25 Mbps	26.37%	—

LNS and DSS, by measuring the value of variables' loss at the receiver ends in a simple LAN scenario with 4 client stations. In each row, Table 3 [13] reports the variables' loss observed when the LNS and the DSS are in use, at a specific level of traffic load produced at the VG. Above 2100 kbps, the variable loss introduced by the DSS cannot be measured, as the DSS seems incapable to support such heavy loads; variables' updates are no longer notified to the user stations, and sometimes the DSS itself crashes.

Again, the performance of the LNS appears to be significantly better than that shown by the DSS; furthermore, especially as concerns the packet loss, the performance of the DSS dramatically decreases when heavy loads are produced by the VG.

5. AN OPERATIVE EXAMPLE

A specific remotely controlled demo has been set up in the many-to-one telemeasurement session by the National Laboratory for Multimedia Communications in Naples. Its goal is to remotely test the operating conditions of a WLAN, in the presence of an adjacent interfering channel, produced by a vector signal generator.

In particular, the qualitative (and, to some extent, quantitative) analysis of the channel throughput is allowed, by observing the quality of a received video sequence and the number of dropped packets and, at the same time, by viewing the resulting waveform on the display of a virtual instrument representing a remotely controlled real spectrum analyzer. The video TX produces a Motion-JPEG encoded stream that feeds the access point (AP) on the right of Figure 11. The RF output of this AP is combined with an interfering signal produced by an *Agilent E-4438C vector signal generator*. The resulting sum traverses a splitter, where the main part of the signal power is directed to the video receiver through a second AP. The decoded video stream is retransmitted over a satellite WAN link or over the Internet (from the National Laboratory for Multimedia Communications in Naples to any remote site) toward the remote observer. Another part of the interfered signal reaches a spectrum analyzer (*Agilent E-4404B*), where the interference phenomenon can be remotely displayed. The GPIB bus (suitably bridged to the laboratory LAN by the E-NET device) disseminates commands and gathers responses from the instruments, thus permitting their complete remote control.

In our experimental setup, the video TX is represented by a VLC application [14], which generates the signal under test (viz the *MotionJPEG*-encoded video), while the interfering traffic consists of a deterministic constant bit rate signal, whose power can be selected by the remote user.

By using the 3D GUI (see Figure 12), it is possible to turn the virtual instrumentation on and off, by clicking on ON/OFF buttons, to see the interfered signal characteristics on a device's display (e.g., a spectrum analyzer), to observe the quality of a received video sequence, to pass and revoke the token to/from a student, to know the statistics of a video transmission and to set the values of experiment variables, by clicking on the instrument's buttons.

For example, when the two transmissions are on nonoverlapping channels (interfering traffic on CH 1 and video one on CH 7) any user can see a very fluent received video, practically no dropped packets, and the classical spectrum of a WLAN transmission. If the interfering signal is shifted on an adjacent channel (CH 6), it is possible to see some dropped packets and a low video quality. If the two

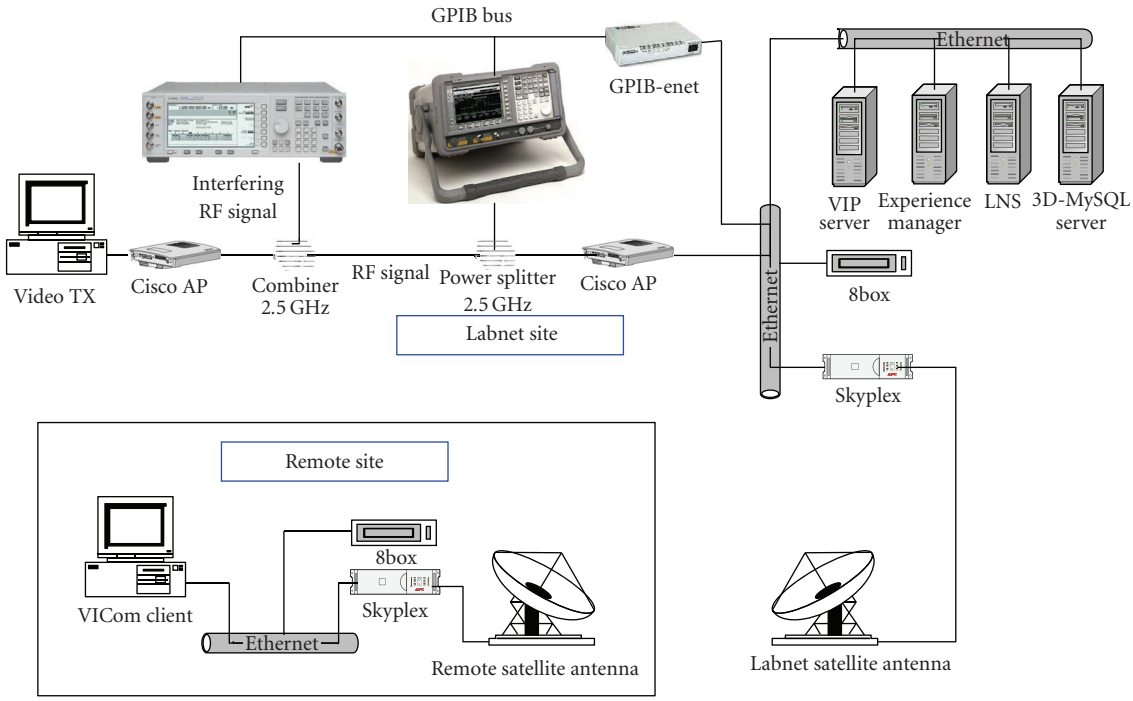


Figure 11: Demo architecture in the many-to-one telemeasurement session.

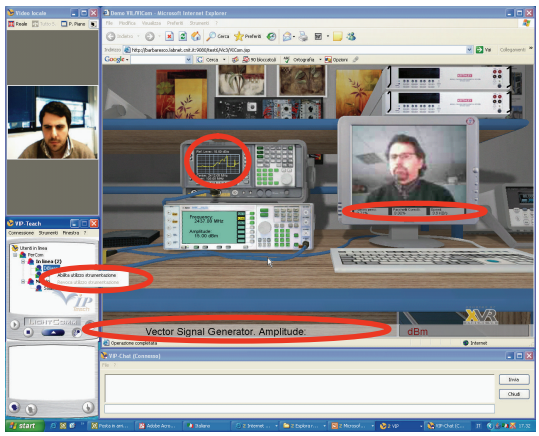


Figure 12: GUI in the many-to-one telemeasurement session.

transmissions are on the same channel (CH 7), the video transmission is completely stopped and it is possible to see a very disturbed spectrum. At this point, if the amplitude of the interfering signal is lowered, the video transmission can start again.

6. USER MOBILITY ISSUES

The VIL test bed does not address mobility issues explicitly. As a matter of fact, the core of the distance learning application does not change, even in case the client used to follow a lecture or access a laboratory session is characterized by a certain degree of mobility. Wireless access, a requisite for mobility, has been indeed considered, since the connection in the example experiment we have described relied upon a satellite link. In this respect, it is worth remembering that the LabNet server protocol, adopted for the management of the client population in the access and control of the measurement devices, has been shown to exhibit a very satisfactory degree of robustness when used over high bandwidth-delay product networks (e.g., satellite or even some types of wireless cellular networks), also in comparison to widespread commercial solutions. Moreover, the full functionalities of the system may be accessed from a wireless network in general, provided that a transmission speed in the range 0.8–1 Mbps is achievable. Problems regarding security should be handled by appropriate authentication and data protection. Possible QoS provisioning mechanisms may be adopted over the wireless link and at the wired/wireless network boundaries.

As regards specifically user mobility, a link with the mechanisms developed within the VICOM project (mobile immersive environment (MIE) testbed) for localization and user guidance may be established. Such mechanisms, based on the use of multiple localization techniques, would facilitate the mobile users in reaching specially equipped classrooms, where they can take advantage of advanced interfaces (e.g., multimedia board, haptic interfaces, or 3D video rendering).

Future developments will regard the establishment of a software interface between LINDA in a mobile environment (LIME) [15], the middleware used for handling the distribution of the context data in the MIE testbed, and the VIL

database, to automatically acquire profiles of mobile users when they enter the classroom.

A final observation regards the adoption of IPv6 at the network layer, especially in conjunction with the need of facing user mobility issues. The VIL test bench has been implemented over IPv4 networks, but it could easily migrate to IPv6. In particular, the Mobile IPv6 (MIPv6) protocol, an IETF standard [16] to provide transparent host mobility within IPv6, should be considered, as it presents several differences to its IPv4 counterpart that provide a simpler, more streamlined protocol (among others, no need for foreign agents, route optimization as standard, integrated support—care of address (COA) and ingress filtering, destination options, COA and multicast routing, use of IPv6 anycast for home agent discovery, etc.).

7. CONCLUSIONS

The paper has presented the design and implementation of the VIL test bed and its main related motivations, as well as critical aspects. The software and hardware strategies, allowing reproduce the context of a real academic classroom in a virtual environment, have been described in some detail.

High portability, good flexibility, and, above all, low cost, make this approach appropriate for educational and training purposes, mainly concerning measurements on telecommunication systems, at universities and research centers, as well as enterprises.

Moreover, the methodology can be employed for remote access to and sharing of costly measurement equipment in many different fields of activity. In fact, the results of a number of tests prove the effectiveness of the proposed solution in terms of both high-sustainable throughput levels and low-delay jitter in comparison with a very popular commercial software package, also in the presence of channels characterized by high delay-bandwidth products (such as satellite links).

As regards in particular the access and management of remote measurement instrumentation and laboratory equipment in general, it is worth mentioning that the LNS platform adopted in the VIL test bench is gradually evolving toward a web services and Grid-based architecture [17], which exploits the functionalities initially developed in the framework of the GRIDCC European project [18]. Specifically, the concept of *instrument element* (IE), developed by GRIDCC, provides a set of services to control and monitor remote physical devices; users view the IE as a set of web services, which provide a common language to the cross-domain collaboration and, at the same time, hide the internal implementation details of accessing specific instruments. The integration of the VIL representation capabilities with Grid-based Remote Instrumentation Services has been addressed in [19].

ACKNOWLEDGMENTS

This work was funded by the Italian Ministry of Education and Scientific Research (MIUR) in the framework of the FIRB_VICOM project. The support of the previous LABNET project in creating the e-measurement framework is also gratefully acknowledged. This work is an extended version of a paper presented at IMMERSCOM 2007, Bussolengo (Verona), Italy.

REFERENCES

[1] D. Tougaw and J. Will, "Visualizing the future of virtual reality," *Computing in Science & Engineering*, vol. 5, no. 4, pp. 8–11, 2003.

[2] Digital Video Broadcasting (DVB), "Interaction Channel for Satellite Distribution Systems," ETSI EN 301 790, 2003.

[3] F. Davoli, G. Nicolai, L. S. Ronga, S. Vignola, S. Zappatore, and A. Zinicola, "A Ka/Ku band integrated satellite network platform for experimental measurements and services: the CNIT experience," in *Proceedings of the 11th Ka and Broadband Utilization Conference*, pp. 715–723, Rome, Italy, September 2005.

[4] F. Antonacci, D. Lonoce, M. Motta, A. Sarti, and S. Tubaro, "Efficient source localization and tracking in reverberant environments using microphone arrays," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '05)*, vol. 4, pp. 1061–1064, Philadelphia, Pa, USA, March 2005.

[5] S. Piva, C. Bonamico, C. Regazzoni, and F. Lavagetto, "A flexible architecture for ambient intelligence systems supporting adaptive multimodal interaction with users," in *Ambient Intelligence*, G. Riva, F. Vatalaro, F. Davide, and M. Alcañiz, Eds., pp. 97–120, IOS Press, Amsterdam, The Netherlands, 2005.

[6] P. Pierleoni, T. Di Biase, G. Cancellieri, F. Fioretti, and S. Pasqualini, "The dynamic construction of multi-user VRML 3D environment for immersive learning on the web," in *Distributed Cooperative Laboratories: Networking, Instrumentation, and Measurements*, F. Davoli, S. Palazzo, and S. Zappatore, Eds., pp. 497–509, Springer, New York, NY, USA, 2006.

[7] V. Cappellini, M. Barni, M. Corsini, A. De Rosa, and A. Piva, "ArtShop: an art-oriented image-processing tool for cultural heritage applications," *The Journal of Visualization and Computer Animation*, vol. 14, no. 3, pp. 149–158, 2003.

[8] F. Davoli, G. Spanò, S. Vignola, and S. Zappatore, "LABNET: towards remote laboratories with unified access," *IEEE Transactions on Instrumentation and Measurement*, vol. 55, no. 5, pp. 1551–1558, 2006.

[9] O. Andrisano, A. Conti, D. Dardari, and A. Roversi, "Telemeasurement and circuit remote configuration through heterogeneous networks: characterization of communications systems," *IEEE Transactions on Instrumentation and Measurement*, vol. 55, no. 3, pp. 744–753, 2006.

[10] F. Bartolini, M. Barni, R. Caldelli, et al., "Research of the image processing and communications lab. of the University of Florence in the cultural heritage field," in *Optical Metrology for Arts and Multimedia*, vol. 5146 of *Proceedings of SPIE*, pp. 116–126, Munich, Germany, June 2003.

[11] A. Vollono and A. Zinicola, "A new perspective in instrumentation interfaces as web services," in *Distributed Cooperative Laboratories: Networking, Instrumentation, and Measurements*, F. Davoli, S. Palazzo, and S. Zappatore, Eds., pp. 451–461, Springer, New York, NY, USA, 2006.

[12] L. Berruti, F. Davoli, S. Vignola, and S. Zappatore, "Interconnection of laboratory equipment via satellite and space links: investigating the performance of software platforms for the

management of measurement instrumentation," in *Satellite Communications and Navigation Systems*, E. Del Re and M. Ruggieri, Eds., pp. 657–666, Springer, New York, NY, USA, 2007.

[13] L. Berruti, S. Vignola, and S. Zappatore, "Investigating the performance of a middleware protocol architecture for tele-measurement," *International Journal of Communication Systems*, vol. 21, no. 5, pp. 509–523, 2008.

[14] VLC—Video LAN Client, http://www.videolan.org/.

[15] G. P. Picco, A. L. Murphy, and G.-C. Roman, "Developing mobile computing applications with LIME," in *Proceedings of the 22th International Conference Software Engineering (ICSE '00)*, M. Jazayeri and A. Wolf, Eds., pp. 766–769, ACM Press, Limerick, Ireland, June 2000.

[16] IETF RFC 3775, Mobility Support in IPv6.

[17] L. Berruti, L. Caviglione, F. Davoli, M. Polizzi, S. Vignola, and S. Zappatore, "On the integration of telecommunication measurement devices within the framework of an instrumentation Grid," in *Grid Enabled Instrumentation and Measurement*, F. Davoli, N. Meyer, R. Pugliese, and S. Zappatore, Eds., Springer, New York, NY, USA, 2008.

[18] GRIDCC project, http://www.gridcc.org/.

[19] L. Berruti, L. Caviglione, F. Davoli, et al., "Joining virtual immersive environments and instrumentation grids for distance learning," in *Proceedings of the IFIP WG 3.6., 3.4. and 3.8. Joint Working Conference on Information Technology for Education and Training (iTET '07)*, pp. 16–28, Prague, Czech Republic, September 2007.