

Communication Architecture for Real-Time Telemedicine Systems on the Internet

Fernando Corteggiano¹, Sebastián J. Tosco¹, L. Sebastián Maglione¹ and Juan Fontana¹

¹Facultad de Ingeniería, Universidad Nacional de Río Cuarto, Ruta Nacional 36 Km 601, Río Cuarto, Argentina.

Abstract— The next generations of teleoperation systems pose important challenges related to the performance of communication networks, software technologies and data security. As is known, on the Internet there are several situations that threaten the communication between extremes: delays, jitter variation, packet loss and firewalls, which can be inconvenient when implementing applications of telemedicine in real time, such as telediagnosis, tele-surgery (remote surgery) or other sensitive applications. The main goal of this work is to propose a real-time bilateral teleoperation architecture for telemedicine. It proposes a data transmission architecture based on the double implementation of the WebSocket Application Messaging Protocol (WAMP), to allow the monitoring and bilateral communication of data, the network security, the incorporation of new participants of the network and the real-time transmission, all transparently to network firewalls. The results show that the system is able to meet the requirements of telemedicine implementations in real time over the Internet.

Keywords— teleoperation, WAMP, internet, telemedicine.

Resumen— Las próximas generaciones de sistemas de teleoperación, plantean importantes desafíos relacionados con el desempeño de las redes de comunicaciones, las tecnologías de software y la seguridad de los datos. Como se sabe, en Internet se producen varias situaciones que atentan contra la comunicación entre extremos: retardos, variación del retardo (*jitter*), pérdida de paquetes y bloqueo de puertos por cortafuegos (*firewalls*), las cuales pueden generar inconvenientes cuando se implementan aplicaciones de telemedicina en tiempo real, como telediagnóstico, telecirugía (cirugía remota) u otras aplicaciones sensibles. El objetivo principal de este trabajo es proponer una arquitectura de teleoperación bilateral en tiempo real para telemedicina. Se propone una arquitectura de transmisión de datos basados en la doble implementación del protocolo *WebSocket Application Messaging Protocol* (WAMP), para permitir el monitoreo y la comunicación bilateral de datos, la securización de la red, la incorporación de nuevos participantes de la red y la transmisión en tiempo real (*soft real time*), actuando de manera transparente a los cortafuegos de red. Los resultados obtenidos demuestran que el sistema es capaz de satisfacer los requerimientos de implementaciones de telemedicina en tiempo real sobre internet.

Palabras clave—teleoperación, WAMP, internet, telemedicina.

I. INTRODUCCIÓN

La teleoperación, por definición, permite la ejecución de tareas en ambientes remotos. Algunas aplicaciones típicas son: la telemedicina, la teleoperación de robots, el entrenamiento a distancia, entre otros [8]. Por otra parte, el uso de Internet como canal de comunicación incrementa en gran medida las posibles aplicaciones de los sistemas teleoperados, a costa de introducir inestabilidad y/o pérdida de desempeño debido a los retardos variables característicos [5], [30].

En una plataforma de teleoperación bilateral el canal de comunicación, Internet, debe disponer de por lo menos dos flujos de datos que envían información en tiempo real en sentidos opuestos el dispositivo de mando, llamado dispositivo maestro, ubicado del lado del usuario, es habitualmente un joystick o un teclado, mientras que el dispositivo teleoperado, llamado dispositivo esclavo, puede ser un sistema de teleconsulta, un robot móvil, un equipo de telecirugía, entre otros.

El flujo de datos desde el maestro hacia el esclavo lleva información de comando como por ejemplo posición y/o velocidad sobre los actuadores del esclavo; mientras que el flujo desde el esclavo hacia el maestro, lleva información de estado (haptico, visual u otros) que permiten cerrar el lazo de control. Además, ambos flujos pueden contener información sobre el estado de la red de comunicaciones lo que permite adecuar el algoritmo de control al estado de la red [5], como se muestra en figura 1.

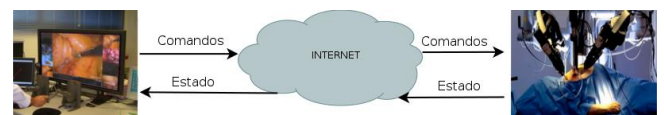


Fig. 1: Esquema general de un sistema teleoperado en internet

El delay, jitter, las pérdidas en la red y la tasa de envío de paquetes son parámetros que tienen un fuerte impacto en el desempeño de un sistema teleoperado. A modo de contraste, un sistema de comunicación de VoIP es común que utilice tramas de voz de 20 a 30 ms. con compresión de señal y una tasa de 30 a 50 paquetes/segundo. En cambio, en sistemas teleoperados se requiere que el delay sea mínimo, con lo cual el procesamiento por bloques no es posible. También la tasa de transmisión de paquetes para

las señales hápticas puede alcanzar los 1000 paquetes/segundo. Cabe aclarar que para disminuir dicha tasa algunos investigadores han propuesto el uso de muestreo adaptivo [8], [13]. En muchas aplicaciones de teleoperación, como los sistemas robóticos de telecirugía sobre Internet, las técnicas de calidad de servicio (QoS) y seguridad son necesarias para garantizar un retardo mínimo ante condiciones cambiantes en la red y evitar ataques de intrusos en el sistema como se muestra en [12]. En general, cualquier sistema de teleoperación tiene particularidades que lo distinguen de otros sistemas en red, como son: el carácter bilateral (envío y retorno de datos), requisito de tiempo real y tamaño de paquete pequeño (20 a 100 octetos) [5]. El aporte del presente trabajo es proponer una arquitectura de teleoperación flexible, de rápido desarrollo, securizada, capaz de soportar los requerimientos para telemedicina basada en el doble uso del subprotocolo *WebSocket Application Messaging Protocol* (WAMP). Esta publicación está organizada en 5 secciones: en la primera se hace una introducción que enmarca la propuesta, mientras que la segunda sección presenta el estado del arte. En la tercera sección se presenta la arquitectura propuesta del sistema de teleoperación bilateral para telemedicina sobre Internet. La cuarta sección muestra resultados experimentales y finalmente, en la última sección se presentan las conclusiones de este trabajo y trabajo a futuro.

II. ANTECEDENTES

Telemedicina se define como “Aportar servicios de salud, donde la distancia es un factor crítico, por cualquier profesional de la salud, usando las nuevas tecnologías de la comunicación para el intercambio válido de información en el diagnóstico, tratamiento y prevención de enfermedades o lesiones, investigación evaluación, y educación continuada de los proveedores de salud, todo con el interés de mejorar la salud de los individuos y sus comunidades” [23]. Dentro de esta línea, se está trabajando ampliamente en diversos campos de investigación y desarrollo (I+D), como por ejemplo: fallas crónicas de corazón [3], [24], cuidado de heridas [9], psiquiatría [18], [19], [31], cirugía [20], soluciones móviles de telemedicina [11], entre otras. Además, existen trabajos previos muy interesantes en el área de vinculación entre las tecnologías de comunicaciones y la medicina [2], [21], [25], [28]. En todos ellos se utilizaron las comunicaciones para transmitir datos de interés hacia un punto remoto o para comunicar varios terminales dentro de un ambiente extenso, como por ejemplo un hospital.

En cuanto a las arquitecturas de comunicación para sistemas de teleoperación bilateral sobre internet, existen varios enfoques. Uno de ellos es el que se plantea en [12] donde se propone utilizar un protocolo de comunicación llamado HoIP que corre sobre UDP y está pensado para dispositivos hápticos sobre redes IP. Este protocolo está escrito en C++ y usa una aproximación multihilo para minimizar las demoras de procesamiento. En el peor caso se lograron retardos de procesamiento (en un sentido) del orden de los 0.6 ms. Otro enfoque se desarrolla ampliamente en [29], proponiendo un modelo de controladores PD-Like en canales de comunicación con retardo variable sobre el protocolo UDP utilizando el joystick Novint Falcon, software MatLab y simulador VREP.

En [14] se propone el protocolo trinomial sobre UDP como algoritmo de control de flujo, observándose una mejora en el delay y el jitter (comparado con TCP), logrando así una transmisión más suave. En el mismo sentido, [35] propone un protocolo de transporte bidireccional (BTP) sobre UDP que provee una técnica de control de congestión novedosa basada en ajustar la tasa de envío de paquetes acorde al estado de congestión de la red.

En una línea semejante, [12] propone un sistema de teleoperación bilateral para telerobótica teniendo en cuenta QoS (calidad de servicio), seguridad y confidencialidad de la información, mejorando (en comparación con TCP) la latencia, la tasa de transmisión y la pérdida de paquetes. Cabe aclarar que el protocolo propuesto en dicha publicación se llama QIPSec y corre sobre UDP.

Por otro lado, en [27] se propone IoT (Internet de las Cosas) como un concepto renovador que permite la interacción entre distintos dispositivos físicos y aplicaciones de software. Este paradigma permite aplicaciones en diversos campos tales como: el hogar, la automatización industrial, los sistemas de soporte médico, la atención móvil de la salud, la asistencia remota de ancianos, la gestión inteligente de la energía, las redes inteligentes, la gestión del tráfico automovilístico, entre muchos otros. Además, dicho trabajo realiza una categorización de las plataformas IoT existentes basada en su diseño, por ejemplo: orientado a eventos, orientado a servicios, basado en máquinas virtuales, etc. Cabe destacar que este tipo de arquitecturas consideran imprescindible poder comunicar cientos de nodos participantes en la comunicación, algunos como sensores, otros como actuadores y otros con un comportamiento más complejo.

En otro sentido, existen trabajos previos donde se plantea la utilización de una arquitectura basada en mensajes dentro de un esquema de teleoperación a través de internet [32]-[34]. Considerando particularidades tomadas de las arquitecturas IoT, y tomando en cuenta los requerimientos de los sistemas teleoperados bilaterales, el presente trabajo propone una arquitectura de comunicación para telemedicina sobre internet basada en el subprotocolo WAMP [15].

III. ARQUITECTURA DE TELEMEDICINA PROPUESTA

A. Visión General

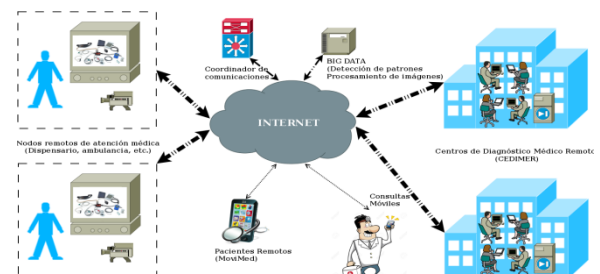


Fig. 2: Esquema general de un sistema de telemedicina.

En la figura 2 se expone un diagrama general de la arquitectura propuesta. Como puede evidenciarse en el diagrama, el sistema permite que diferentes nodos remotos de atención médica (por ejemplo, dispensarios, centro de salud e incluso ambulancias) accedan a diversas instancias de consultas médicas especializadas. Además, es posible vincular el sistema de comunicaciones con servicios de Big

Data y Cluster de Computación, los cuales pueden brindar servicios especializados para diagnóstico automatizado y registro de eventos.

Por otro lado, la idea propuesta puede implementarse a distintas escalas:

1) En el ámbito intra-hospitalario: Hospitales y clínicas cuyos profesionales, dependencias y pacientes estén interconectados.

2) En el ámbito de la medicina ambulatoria: Con la correcta selección de equipamiento de comunicaciones es posible brindar asistencia a los paramédicos para un mejor cuidado del paciente durante el traslado.

3) En el ámbito interhospitalario: Es posible con esta arquitectura vincular varios centros de atención médica (CEDIMER) para realizar juntas médicas a distancia.

4) En el ámbito privado: es posible pensar en servicios de monitorización y control de pacientes a través de la red celular (MoviMed).

Una vez visualizado el sistema de forma global, se procede a explicar los conceptos de funcionamiento y las tecnologías involucradas.

B. Tecnologías y funcionamiento

1) Comunicaciones Tradicionales en Internet HTTP, TCP, UDP e IP: El protocolo *Internet Protocol* (IP) provee un mecanismo de comunicación entre computadores o equipos [26]. IP distingue a cada equipo en la red por un único número denominado dirección IP. Los protocolos de transporte TCP y UDP son encapsulados habitualmente en los paquetes IP.

El protocolo *Transmission Control Protocol* (TCP) [4], es el protocolo más utilizado en la capa de transmisión en Internet, provee una comunicación bidireccional confiable, cuya tasa de transferencia se ajusta al estado de congestión de la red.

El protocolo de *User Datagram Protocol* (UDP) [4] es otro de los protocolos de transporte usado en Internet. Comparado con TCP, UDP es menos complejo y confiable, cuya tasa de transferencia lo define el equipo que envía, sea cual fuere el estado de congestión de la red.

HTTP es un protocolo de intercambio de contenido ubicado en la capa de aplicación basado en el modelo petición/respuesta [6]. A HTTP se le asigna el puerto 80 sobre TCP o UDP, sin embargo la mayoría de las comunicaciones en Internet actualmente se realizan sobre TCP, lo que permite que las comunicaciones sobre este protocolo puedan atravesar la Internet sin problemas.

El modelo cliente/servidor es una de las arquitectura de red para conectar computadoras o aplicaciones, donde el servidor provee recursos y los clientes los solicitan. Usualmente, el modelo cliente/servidor está diseñado como un modelo solicitud/respuesta, donde el servidor no se comunica activamente con los clientes. El problema que presenta HTTP es que para mantener abierta las conexiones se requiere repetidas peticiones de conexión al servidor web, incrementando así la carga en este como así también la cantidad de datos transferidos.

2) *WebSocket*: El protocolo *WebSocket* habilita la comunicación *full-duplex* a través de un simple socket entre el cliente y el servidor. Esta tecnología representa un enorme avance en las comunicaciones en tiempo real y la programación orientada a eventos sobre Internet [16].

Además, este protocolo soporta datos de tipo binario o de texto, sobre conexiones cliente-servidor.

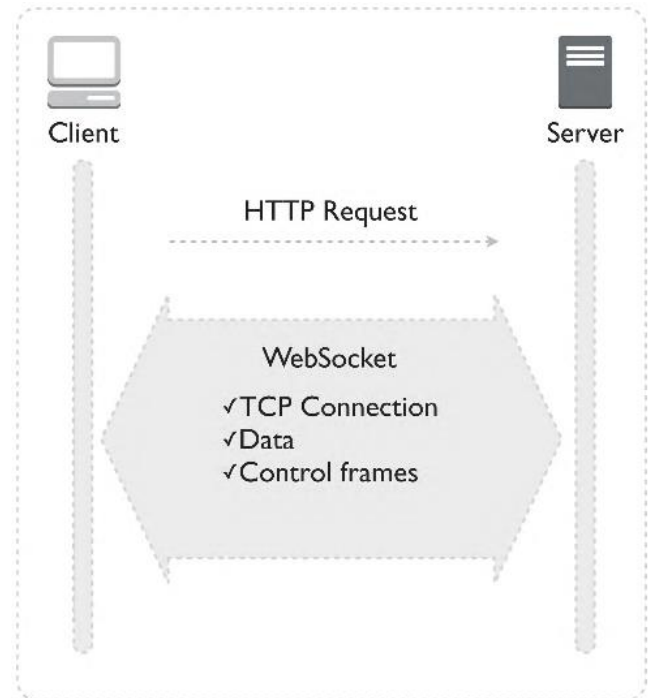


Fig. 3: WebSocket Protocol [10].

La comunicación vía *WebSocket* se inicia con una petición HTTP tras lo cual se puede especificar un subprotocolo (por ejemplo, WAMP) y el resto de los datos comunicados en ambos sentidos se encapsulan sobre TCP, como se muestra en la figura 3. La latencia en las comunicaciones es otro de los beneficios de utilizar *WebSockets*. Como el socket está siempre abierto y escuchando, los datos son enviados inmediatamente desde el servidor al cliente, reduciendo el tiempo al mínimo, en comparación con otros paradigmas de comunicación asíncronos, donde hay que realizar una petición, procesar la respuesta y luego devolver la misma.

3) Subprotocolo WAMP: *WebSocket Application Messaging Protocol* (WAMP) es un subprotocolo de *WebSocket*, el cual ofrece dos patrones de mensajería de alto nivel: Publicador/Subscriber (Pub/Sub) y Llamada a Procedimiento Remoto (RPC). WAMP suplementa al protocolo *WebSocket*, sin embargo no crea ningún standard extra, pues sus bases son estándares web bien conocidos: *WebSocket*, JSON and URI. *WebSocket* es el canal de transporte estándar donde se establece la comunicación WAMP asumiendo confiabilidad, ordenamiento y un canal full-duplex para los mensajes. Su arquitectura general se muestra en la figura 4.

Debido a que las conexiones TCP comunes sobre puertos diferentes al 80 son habitualmente bloqueadas por los administradores de redes, el uso de esta tecnología proporcionaría una solución a este tipo de limitaciones proveyendo una funcionalidad similar a la apertura de varias conexiones en distintos puertos, pero multiplexando diferentes servicios *WebSocket* sobre un único puerto TCP (a costa de una pequeña sobrecarga del protocolo).

Varias publicaciones han propuesto el uso de protocolos de control de congestión sobre UDP como por ejemplo Trinomial [14], HoIp [8], entre otros [5], [12], [35] los cuales han mostrado una eficiencia adecuada para la

teleoperación bilateral. En el presente trabajo se propone utilizar como protocolo de transporte a TCP, dado que éste permite establecer conexiones entre puntos remotos minimizando la posibilidad de bloqueos en Internet por la presencia de Cortafuegos. Otra razón para la antedicha elección es que los estándares actuales han incluido el desarrollo de tecnologías de comunicación que utilizan ampliamente estándares web, los cuales están siendo adoptados en las sucesivas actualizaciones de los navegadores web.

Por otra parte, hoy en día la evolución de los sistemas teleoperados requiere la participación cooperativa de varios esclavos comandados por un maestro. En vista de esto, resultaron oportunos los conceptos integradores de la línea IoT para definir un esquema de comunicación tipo Publicador-Subscriptor, el cual permite de forma relativamente sencilla agregar o quitar participantes en la comunicación. Dentro de los protocolos de comunicación en tiempo real que están actualmente en pleno desarrollo, se encuentra *websocket* definido en [17], quien propone un subprotocolo llamado WAMP [22], el cual provee dos modelos de envío de mensajes en un protocolo unificado (RPC y Publicador-Subscriptor) que permite una implementación liviana y una portabilidad suave [15]. Lo innovador de *websocket* es que permite comunicación bilateral simultánea (*full-duplex*) utilizando un único socket y un único inicio de sesión http. También proporciona seguridad y tiene compatibilidad con los navegadores mod Como se muestra en figura 4, WAMP, a diferencia de *websocket*, necesita un router WAMP, el cual puede estar en cualquier parte de la nube, e incluso puede ser usado dentro de tecnologías de cluster. Para este trabajo se escogió el router CrossbarIO, el cual está desarrollado en Python y usa tecnologías de programación asíncrona, que logran más de 20K peticiones/seg. sobre redes 10GbE [1]. Existen implementaciones de este router sobre plataformas de *cloud computing* como AZURE (Microsoft), EC2 (Amazon) y Open-Stack [15].

En vista de lo antes mencionado, se propone un sistema para teleoperación bilateral para telemedicina a través de internet utilizando como protocolo de comunicación a WAMP.

C. Tele-WAMP

Sobre la base de lo ya mencionado, se propone una especificación a nivel de capa de aplicación, denominada Tele-WAMP, que implica generar un canal de comunicaciones bilateral utilizando una doble implantación del protocolo WAMP. De esta manera, se hace posible la realimentación en tiempo real del estado de la red y del estado de los nodos teleoperados (esclavos), pudiendo así ajustar las tasas máximas de transferencia de datos, para lograr un mejor desempeño del sistema. Como caso de estudio se implementó un modelo de teleoperación bilateral en donde cada nodo publica información de estado y se suscribe para recibir las órdenes de comando (Figura 5).

Para implementar dichos nodos se utilizó el framework de tiempo real de código abierto llamado Autobahn, desarrollado sobre Twisted, el cual es un framework de red para programación dirigida por eventos escrito en Python.

El patrón de diseño de Twisted [36] se basa en el modelo de reactor, el cual es un lazo que reacciona ante la ocurrencia de eventos, los cuales son atendidos por funciones de callback. Esta tecnología permite generar

eventos cada 1 [ms] y enviarlo a la red a través del protocolo Wamp, el cual usa la capa de transporte TCP. El framework *Autobahn* tiene la ventaja de poseer múltiples subproyectos que permiten su uso en diferentes lenguajes: C/C++, Javascript, Android y Python, siendo este último el elegido para este trabajo. Además, para acceder a información del estado del protocolo TCP en tiempo real, se utilizó el módulo para kernel linux llamado TCPProbe, del cual se obtienen parámetros como: SRTT, tamaño de ventana de congestión, umbral de congestión, tamaño de paquete, entre otros.

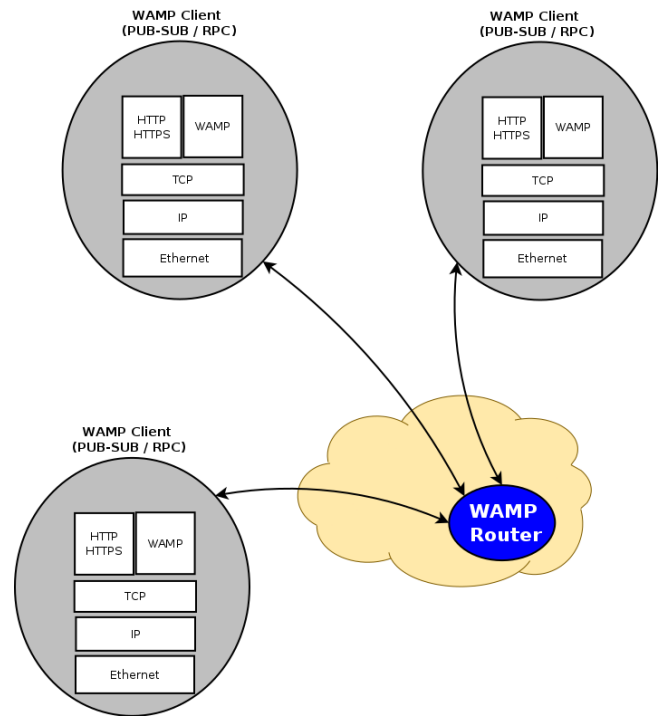


Fig. 4: Arquitectura Wamp

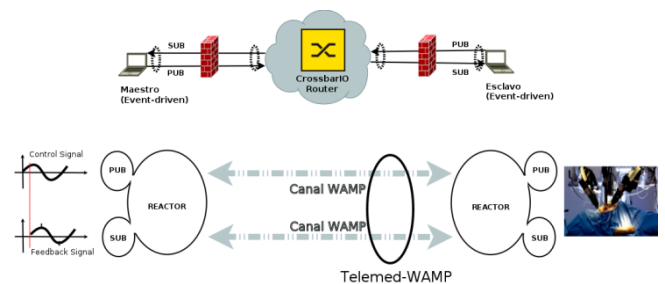


Fig. 5: Tele-WAMP

D. Análisis del canal de comunicaciones

El nodo implementado que representa al maestro en la figura 5 genera una señal variable siguiendo un patrón sinusoidal de baja frecuencia la cual es publicada en el enrutador para que el/los nodo/s esclavo/s se suscriba/n a ella. El nodo esclavo recibe la señal de comando y publica otra señal que representa su información de estado (por ejemplo una fuerza) que sirve de realimentación al nodo maestro, el cual se suscribe a ella. De esta manera se forma un lazo de control cerrado a través de la red (Internet). Además, como se aprecia en la figura 5, este modelo tiene la capacidad de ser transparente a los cortafuegos y también permite la segurización de la información [22].

La diferencia de nuestro modelo con [5], [14] es que se propone utilizar TCP. Asumir esto implica aceptar dos fuentes de latencia: la generada por el tamaño de la ventana de congestión y la que deviene de las retransmisiones (ya que TCP no permite pérdidas) [7]. En la figura 6 se observa que, luego de sintonizar el kernel del sistema operativo para minimizar la latencia en TCP, la ventana de congestión se limita al envío de un solo paquete pequeño (en el ejemplo 119 bytes). Esta estrategia es totalmente implementable, dado que los ancho de banda de acceso a Internet que se puede contratar superan aproximadamente en un orden los requerimientos mínimos. Por ejemplo, si se desea transmitir cada 2 ms. paquetes de unos 200 bytes se necesita alrededor de 1Mbps; en muchos países es común disponer de conexiones 20/5 Mbps (Uplink/Downlink).

Cabe aclarar que se utilizó Python3 con Twisted dado que este *framework*, cuando se comunica con TCP, utiliza el modo Push (PSH), el cual indica al protocolo que procure no agrupar paquetes. Además, pueden configurarse otros parámetros de TCP como por ejemplo: *tcp low latency*, *tcp congestion control*, *tcp rmem* y *tcp wmem*, entre otros. Para las mediciones realizadas, se dejaron las opciones por defecto de Debian Jessie y Ubuntu 14.04, para el maestro y el esclavo respectivamente.

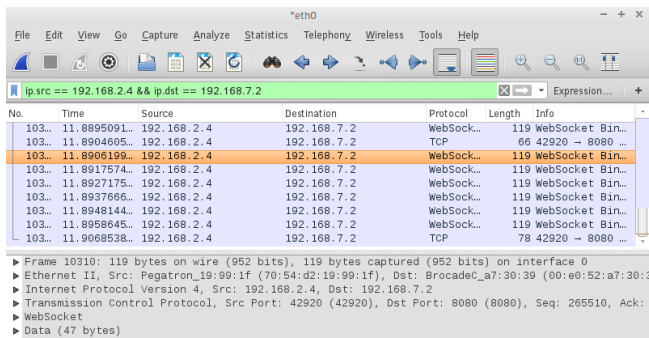


Fig. 6: Captura de tráfico.

IV. EXPERIMENTACIÓN

A continuación, se plantean tres escenarios de pruebas sobre el canal de comunicaciones usando Tele-WAMP:

A. Modelo de laboratorio, donde se emula la red WAN utilizando la herramienta *netem*, y la red de acceso se presume limitada mediante un filtro *token bucket*.

B. Escenario regional dentro de los Estados Unidos representando retardos y jitter típicos donde el router WAMP y el esclavo están en Tulsa (Oklahoma) y el maestro en Houston (Texas) con velocidades Descarga/Carga de 25Mbps/5Mbps.

C. Aplicación que utiliza un kit de laboratorio para medir variables biomédicas, entre ellas ECG, y la conecta al sistema Tele-WAMP.

En los escenarios uno y dos la prueba consiste en enviar una señal senoidal (representando comandos) desde el maestro al esclavo. Éste último recibe dicha señal y devuelve inmediatamente la misma al maestro (representando dato de estado). Cabe aclarar que si bien en las pruebas se establece el doble canal de comunicación utilizando Tele-WAMP, no se hace implementación de un algoritmo de control, lo cual puede considerarse trabajo a futuro.

A. Modelo de laboratorio

En la figura 7 se muestra el diagrama esquemático que representa dicho escenario. La red WAN se emula con 100 ms de delay y 1 ms de jitter, y la velocidad media de la red de acceso es: 1mbps y 400kbps.

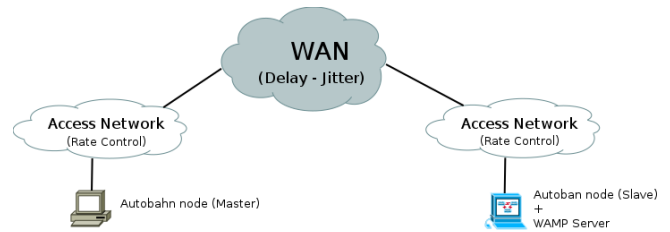


Fig. 7: Diagrama del escenario de medición.

En la figura 8 se observa que el ancho de banda disponible es suficiente para la aplicación (Tiempo de generación entre paquete (Itgt)= 2 ms) por lo que TCP no va a la congestión provocando pérdidas ni agrupamientos de paquetes. Lo antes mencionado se evidencia en que la ventana de congestión permanece constante.

En la figura 9 se observa como al reducirse el ancho de banda y por tanto congestionarse TCP, empiezan tanto las pérdidas como la agrupación de paquetes, degradando así la calidad de la señal en el sistema teleoperado. Sin embargo, esta situación puede subsanarse decrementando el tiempo de intergeneración de paquetes (por ejemplo, 10ms), como se observa en la figura 10. Puede notarse un desempeño similar al de la figura 8.

B. Modelo real

En la figura 11 puede observarse que en la implementación del modelo real no hay pérdida de paquetes, si bien debido al jitter hay cierto movimiento de las muestras. Este buen comportamiento, incluso a una tasa de generación de paquetes de 2ms, muestra la potabilidad de uso del sistema en escenarios de red reales.

Es interesante destacar que la bondad en el comportamiento del sistema se debe principalmente a la disponibilidad de ancho de banda en los sitios escogidos y al buen desempeño de los troncales de redes de fibra.

C. Aplicación en Telemedicina

Una vez que se analizó el comportamiento del canal de comunicaciones utilizando Tele-WAMP se implementó con éxito la arquitectura propuesta para telemedicina mostrada en la figura 5. Como caso particular, se transmitió biometría en tiempo real vía internet y se visualizaron dichos datos tanto en ordenadores como en dispositivos móviles, tanto desde redes cableadas, inalámbricas y móviles.

Cabe destacar que los datos biométricos sensados corresponden en la prueba a un electrocardiograma generado a partir del kit educativo *E-Health Sensor Platform* (de *CookingHacks*). La calidad de los datos resultó apropiada para la evaluación del estado de un paciente en situaciones de emergencia. En la figura 12 se muestra una captura de pantalla del funcionamiento del sistema utilizando una interfaz web creada para tal fin.

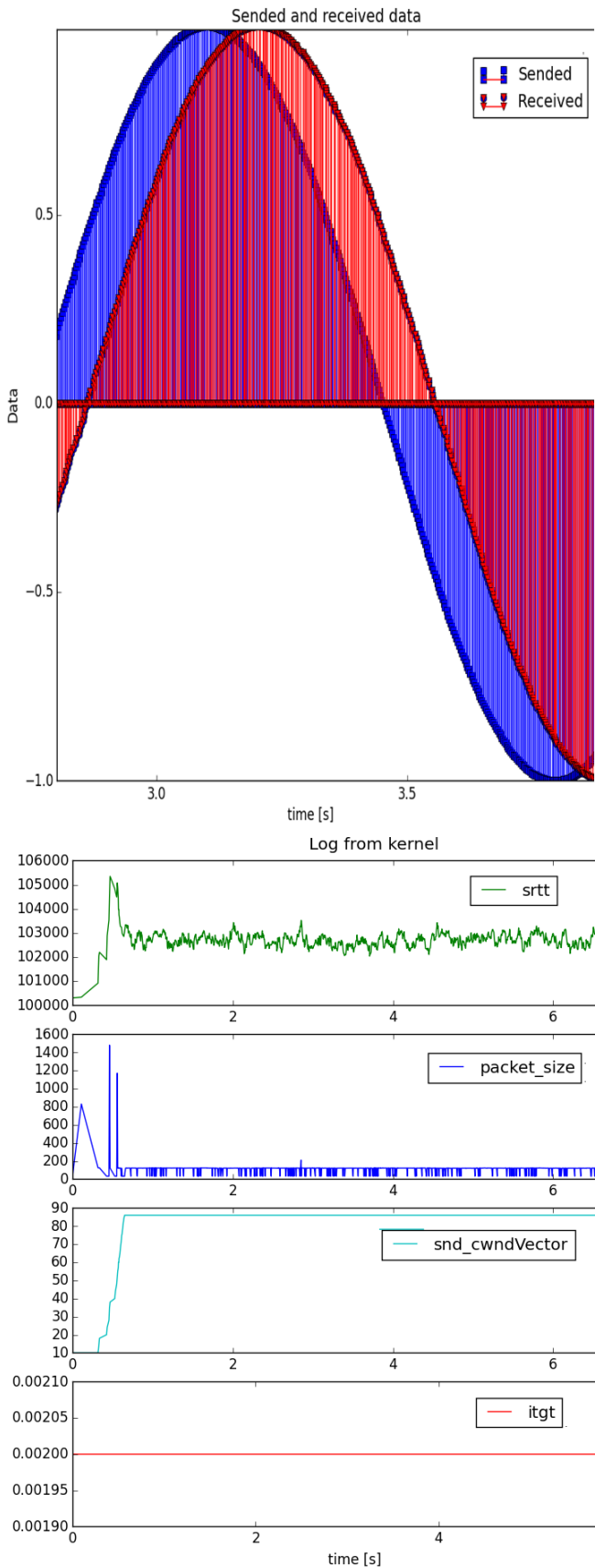


Fig. 8: Señal enviada y recibida con ancho de banda suficiente; no hay agrupamiento de paquetes (1 Mbps, Itgt=2ms)

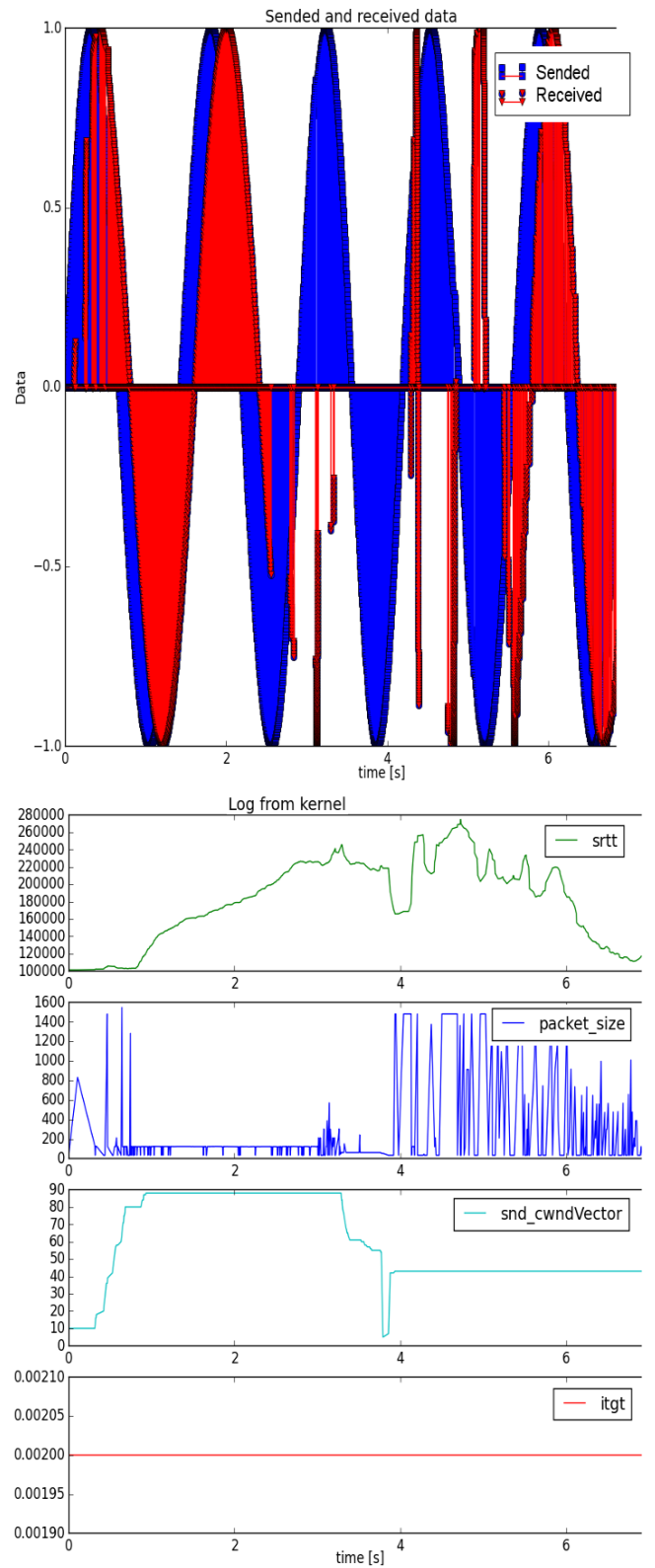


Fig. 9: Señal enviada y recibida (400 Kbps, Itgt=2ms)

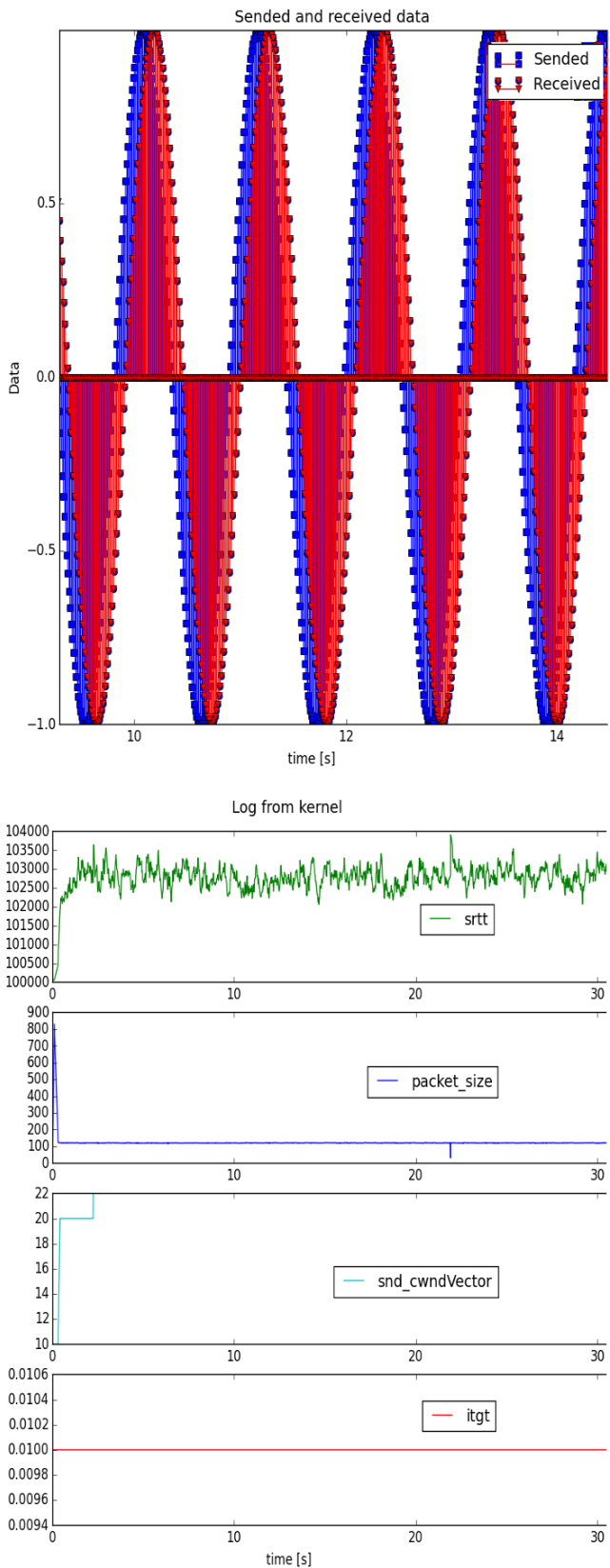


Fig. 10: Señal enviada y recibida (400 Kbps, Itgt=10ms).

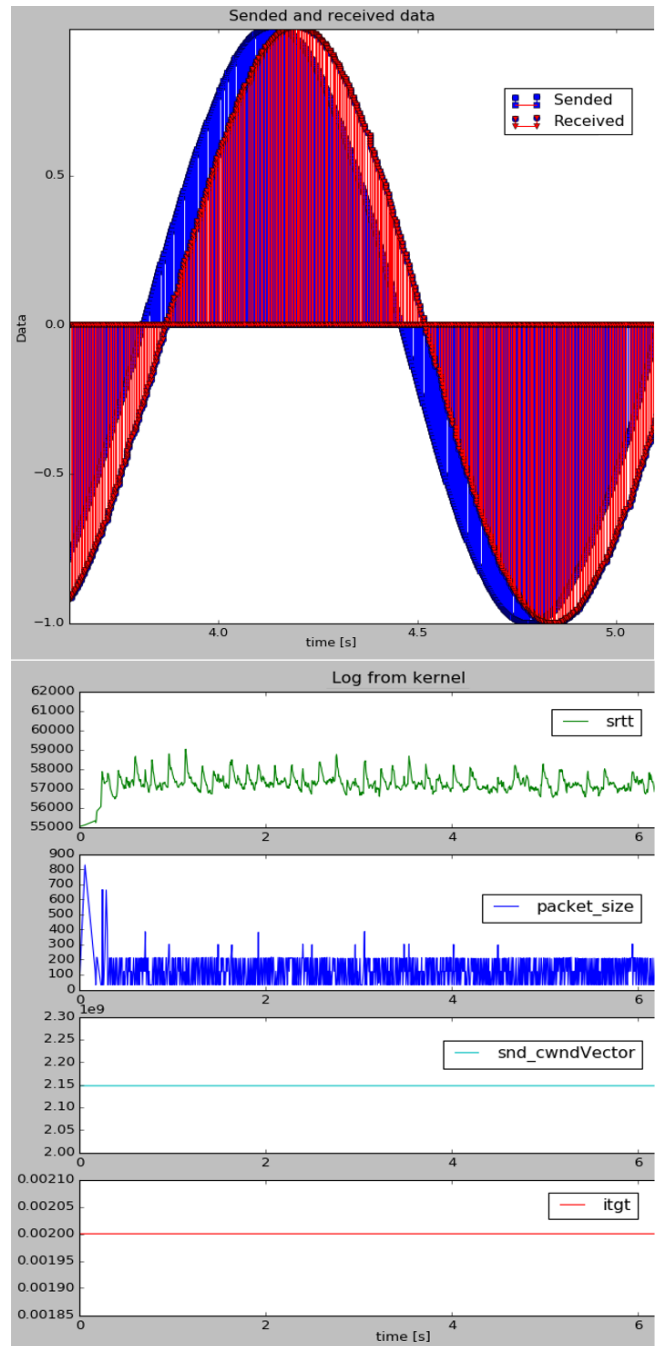


Fig. 11: Señal enviada y recibida escenario real.

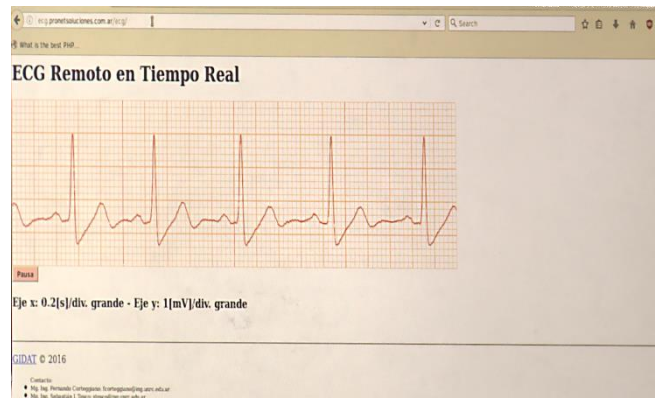


Fig. 12: Prueba ECG.

V. CONCLUSIONES

En este trabajo se ha propuesto una arquitectura de comunicación para teleoperación bilateral sobre internet basada en el subprotocolo WAMP, usando TCP como protocolo de transporte y Python como lenguaje de programación bajo el sistema operativo Linux. En base al paradigma publicador-subscriptor se desarrolló un modelo apropiado para la teleoperación, utilizando técnicas de programación asíncrona en los nodos. En cuanto a la correlación entre las señales transmitidas y las recibidas, puede notarse una buena correlación claramente afectada por el retardo medio de la red, siempre y cuando TCP no sea llevado a la congestión por limitantes de ancho de banda. Cuando esto último ocurre, es posible sintonizar el sistema escogiendo una tasa de generación de paquetes adecuada al ancho de banda disponible. También, se evidenció que es posible recibir parámetros del estado de TCP desde el *kernel* del sistema operativo e incluso configurar el mismo para disminuir latencia.

En virtud a los resultados obtenidos se considera que esta plataforma es una buena opción para implementar sistemas de teleoperación bilateral aplicados a telemedicina, permitiendo que médicos, instituciones y pacientes estén conectados con el fin de lograr una mejor cuidado de la salud. Como trabajo futuro, se puede anexar capacidades de procesamiento de información (*Big Data*) que asistan al médico para un diagnóstico temprano de enfermedades. También, desde el punto de vista del control, se puede seguir investigando sobre la aplicación de filtros bayesianos [13], controlador PD-Like [29], entre otros, sobre Tele-WAMP. Queda como trabajo a futuro implementar dicho tipo de escenarios. Además, sería muy oportuno realizar la sintonización del sistema de forma automática y en tiempo real, utilizando diferentes predictores.

AGRADECIMIENTOS

Este trabajo ha sido realizado parcialmente gracias al apoyo de la Secretaría de Ciencia y Tecnología de la Universidad Nacional de Río Cuarto en el marco de los proyectos PPI2016-2018. También contó con el apoyo del Ministerio de Educación de la Nación en el marco de la 23° Convocatoria “Jorge A. Sabato” de vinculación tecnológica.

REFERENCIAS

- [1] www.crossbar.io. Accessed: 2016-04-29.
- [2] M Bonaccorso, D Lara, C P´arraga, D Navarro, and Jaren J. Teleinformática para prevenir trastornos hipertensivos durante el embarazo. XX Congreso Argentino de Bioingeniería - SABI2015, October 2015.
- [3] Robyn A Clark, Sally C Inglis, Finlay A McAlister, John GF Cleland, and Simon Stewart. Telemonitoring or structured telephone support programmes for patients with chronic heart failure: systematic review and meta-analysis. *Bmj*, 334(7600):942, 2007.
- [4] Douglas E. Comer. *Computer Networks and Internets*. Prentice Hall Press, Upper Saddle River, NJ, USA, 5th edition, 2008.
- [5] M. Diaz-Cacho, A. Barreiro, and M.G. Rivera. Bidireccionalidad y eficiencia en el transporte de datos de teleoperación a través de redes ip. *Revista Iberoamericana de Automática e Informática Industrial IRIAI*, 7(2):99 – 110, 2010.
- [6] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, and T. Berners-Lee. Hypertext transfer protocol – http/1.1, 1999.
- [7] Ashvin Goel, Charles Krasic, and Jonathan Walpole. Low-latency adaptive streaming over tcp. *ACM Trans. Multimedia Comput. Commun. Appl.*, 4(3):20:1–20:20, September 2008.
- [8] V. Gokhale, O. Dabeer, and S. Chaudhuri. Hoip: Haptics over internet protocol. In 2013 IEEE International Symposium on Haptic

- Audio Visual Environments and Games (HAVE), pages 45–50, Oct 2013.
- [9] R Hofmann-Wellenhof, W Salmhofer, B Binder, A Okcu, H Kerl, and HP Soyer. Feasibility and acceptance of telemedicine for wound care in patients with chronic leg ulcers. *Journal of telemedicine and telecare*, 12(suppl 1):15–17, 2006.
- [10] Fuguo Huang. Web technologies for the internet of things. Abstract of Master thesis, July 2013.
- [11] Study Group 2 ITU-D. *Mobile eHealth Solutions for Developing Countries*, 2011.
- [12] A. Abou El Kalam, A. Ferreira, and F. Kratz. Bilateral teleoperation system using qos and secure communication networks for telemedicine applications. *IEEE Systems Journal*, PP(99):1–12, 2015.
- [13] Jae-young Lee and Shahram Payandeh. Bayesian framework for bilateral teleoperation systems over unreliable network. *Robotica*, FirstView:1–16, 4 2016.
- [14] P. X. Liu, M. Q. H. Meng, P. R. Liu, and S. X. Yang. An end-to-end transmission architecture for the remote control of robots over ip networks. *IEEE/ASME Transactions on Mechatronics*, 10(5):560–570, Oct 2005.
- [15] F. Longo, D. Bruneo, S. Distefano, G. Merlino, and A. Puliafito. Stack4things: An openstack-based framework for iot. In *Future Internet of Things and Cloud (FiCloud)*, 2015 3rd International Conference on, pages 204–211, Aug 2015.
- [16] P. Lubbers and F. Greco. Html5 web sockets: A quantum leap in scalability for the web. *SOA World Magazine*, March 2010.
- [17] Alexey Melnikov and Ian Fette. The WebSocket Protocol. RFC 6455, October 2015.
- [18] D Mucic. Telepsychiatry in denmark: Mental health care in rural and remote areas. *J. e-Health Technol. Appl*, 5(3), 2007.
- [19] D Mucic. International telepsychiatry, patient acceptability. *Global telemedicine/e-health updates: Knowledge resources*, 1:383–384, 2008.
- [20] M. Nayeemuddin, M. A. Majeed, A. Muneer, and A. Misra. An outpatient survey of plastic surgery patients. *Med-e-Tel*, 2007. Luxembourg, 2009.
- [21] C. Oberdan Rolim, F. L. Koch, M. Assuncao, and C. Becker Westphall. Towards a grid of sensors for telemedicine. 2006.
- [22] Tobias G. Oberstein and Alexander Goedde. The Web Application Messaging Protocol. Internet-Draft draft-oberstet-hybi-tavendo-wamp-02, Internet Engineering Task Force, April 2016. Work in Progress.
- [23] WHO (World Health Organization). *Telemedicine. Opportunities and developments in member states. Report on the second global survey on eHealth. Global Observatory for eHealth series. Volume 2*, 2010.
- [24] Guy Par´e, Mirou Jaana, and Claude Sicotte. Systematic review of home telemonitoring for chronic diseases: the evidence base. *Journal of the American Medical Informatics Association*, 14(3):269–277, 2007.
- [25] David Piccinini, Nicolás Andino, Sergio Ponce, Martín Roberto, and Natalia López. Sistema vestible de adquisición y monitoreo de señales biológicas., 2015.
- [26] Jonathan B. Postel, Carl A. Sunshine, and Danny Cohen. The ARPA internet protocol. *Computer Networks*, 5:261–271, 1981.
- [27] M. A. Razzaque, M. Milojevic-Jevric, A. Palade, and S. Clarke. Middleware for internet of things: A survey. *IEEE Internet of Things Journal*, 3(1):70–95, Feb 2016.
- [28] M. Sanchez-Ocampo, B. Segura-Giraldo, R. Florez-Hurtado, and C. Cortés-Aguirre. Diseño del prototipo de un módulo paciente para asistencia remota de personas con deterioro cognitivo en domicilio. Oct 2015.
- [29] E. Slawiński, V. Mut, and D. Santiago. Pd-like controller for delayed bilateral teleoperation of wheeled robots. *International Journal of Control*, 0(0):1–10, 2016.
- [30] E. Slawiński, J. F. Postigo, and Vicente Mut. Bilateral teleoperation through the internet. *Robotics and Autonomous Systems*, 55(3):205–215, 2007.
- [31] M Stojakovic, M Jordanova, and F Lievens. Compare psychiatric consultation face-to-face and telepsychiatry. *Global telemedicine/ellealth Updates: Knowledge Resources*, 4:130–134, 2011.
- [32] S. Tosco and F. Corteggiano. Evaluación de una plataforma basada en mensajes para la operación remota de procesos. *Mecánica Computacional. San Carlos de Bariloche: Graciela Bertolino, Mariano Cantero, Mario Storti y Federico Teruel (Eds.)*, 33:2595–2618, 23-26 Setiembre 2014. issn 1666-6070.
- [33] S. Tosco, F. Corteggiano, and M. Broll. Implementación de un esquema de teleoperación utilizando el sistema operativo ros en el

- contexto de un laboratorio remoto. *Mecánica Computacional*, Vol XXXI:3741–3749, 2012. Salta, 2012.
- [34] S. Tosco, F. Corteggiano, M. Broll, and H. Magnago. Teleoperación de un laboratorio remoto para la enseñanza a distancia de la ingeniería. V Reunión de Trabajo en Procesamiento de la Información y Control, 2013.
- [35] R. Wirz, R. Marin, M. Ferre, J. Barrio, J. M. Claver, and J. Ortego. Bidirectional transport protocol for teleoperated robots. *IEEE Transactions on Industrial Electronics*, 56(9):3772–3781, Sept 2009.
- [36] R. Wysocki and W. Zabierowski. Twisted framework on game server example. *CAD Systems in Microelectronics (CADSM)*, 2011 11th International Conference. The Experience of Designing and Application of, 2011.