Analyzing the behavior of a real-time telerobotic system on IEEE 802.11b wireless networks[1]

Carlos del Puerto Riquelme, R. García and M.P. Malumbres
Technical University of Valencia
Camino de Vera 17,
46071, Valencia, Spain
{cdpuerto,roman,mperez}@disca.upv.es

Abstract - *A real-time telerobotic system is an example of application where temporal constraints are imposed on media streams delivery (typically video and navigation control). The communication network has to offer an appropriate transport service, making this kind of applications successful. So, we will analyze the feasibility of a real-time navigation system on IEEE 802.11b wireless networks. For that purpose, we have developed a prototype that consists of one autonomous robot equipped with a webcam and an 802.11b network adapter. The control station sends navigation commands to the robot and, at the same time, it is receiving a video stream that shows to the user a frontal view of actual robot location. In this framework, we analyze the end-to-end delays in both the video and navigation data flows under different network conditions. Also, we measure the impact of several network parameters (mobility, traffic, link quality, etc.) to determine the feasibility of this kind of applications in WLANs.*

Keywords: *Telerobotics, client-server applications, real-time communications, wireless networks, performance evaluation.*

1. Introduction

IEEE's 802.11b [1] standard is being increasingly used throughout corporations worldwide due to its good balance of cost, range, bandwidth and flexibility. The bandwidths set by the standard range from 1 to 11 Mbps. It offers two operation modes named Point Coordination Function (PCF) and Distributed Coordination Function (DCF). PCF is used in infrastructure mode, where Access Points are responsible for coordinating the transmissions from nodes. DCF, on the other hand, is a distributed mechanism where each node has the responsibility of sensing the medium, to avoid and react to collisions.

The new standard 802.11g aims at higher bandwidths maintaining the frequency of operation (2.4 GHz) and, along with 802.11e [2] that supplies QoS at MAC layer form a good base to support multimedia traffic.

Although the main application area of wireless networks is related with user access to existing wired network infrastructure, other applications, like telerobotics can benefit from WLANs technology. The development of teleoperated systems has gained considerable attention in recent years due to the new potential applications, such as remote production monitoring [3], remote exploration and manipulation in inhospitable environments [4][5][6], tele-surgery [7][8], and remote training. Important issues concerning communication channels, random propagation delays, bandwidth limitations, fault-tolerance, synchronization, tele-presence, and the stability of the robotic systems involving human operators have all been taken into account in different works across the literature [9][10][11][12]. Most of them consider Internet as the interconnection network between telecontrolled system(s) and control station(s)[13].

The use of wireless channels for teleoperating mobile and autonomous robots is not new. Most of the existing wireless-controlled robots use a specific (non-standard) radio-modem for communications between control station and robot. However, this wireless link use to be a point-to-point dedicated wireless link which usually works under good signal quality environments. Therefore, in that case, the wireless link does not represent a drawback in terms of performance of the overall system.

Other recent works describe implementations of teleoperation systems with standard wireless devices. In [14] authors use a PDA with an IEEE 802.11 wireless interface for telecontrolling a robot, and in [15] the same was done through a WAP connection. However, they do not analyze the behavior and performance of wireless link for the correct operation of this kind of applications.

IEEE 802.11b networks (WLANs) are becoming a very popular network technology, and their future is very promising. When working in infrastructure mode, the access points (base stations) provide the network connectivity between mobile and wired hosts. However, WLAN nodes share the medium through a CSMA-CA protocol, so network latency may be significant and dependent of current traffic load. The time-varying

[1] This work was funded by the *Ministry of Science and Technology* of Spain under grant TIC/2003/00339, by the *Junta de Comunidades de Castilla La-Mancha* under grant PBC/03/001 and by the *Spanish Agency of International Cooperation* (MAE-AECI).

conditions of the wireless channel, roaming processes and certain node mobility patterns, may disturb established communications, increasing the packet lost ratio and the average delay and jitter, up to intolerant levels for certain kind of multimedia applications.

In this paper, we are going to study the feasibility of real-time telerobotic system in standard WLAN networks, taking into account the different impairments that may appear during a teleoperation session. For that purpose, firstly we will analyze the end-to-end delays of both video and navigation data flows in our own system prototype, so we can determine its limitations. Then, we carry out some tests under different network scenarios, measuring network parameters as packet delay, jitter and lost ratios, in order to determine the stability of the system.

The paper is organized as follows: In section 2 we describe the telecontrolled platform that will be used in this work. Section 3 shows a detailed study of end-to-end delays in both video and navigation data flows. In section 4, different scenarios are planned in order to measure the impact of different network conditions in the telecontrolled robot session. Finally, in section 5, some conclusions and future work are drawn.

2. Remote-controlled navigation system

We have developed a basic real-time telerobotic prototype. In Fig. 1, we show our framework, based on an ER1 mobile robot, from Evolution Robotics [16], equipped with a laptop, WLAN network adapter and one Webcam. The control station consists of a PC connected to campus network. We also use one access point, in exclusive access, which connects control station with our robot system through a VPN (Virtual Private Network) tunnel.

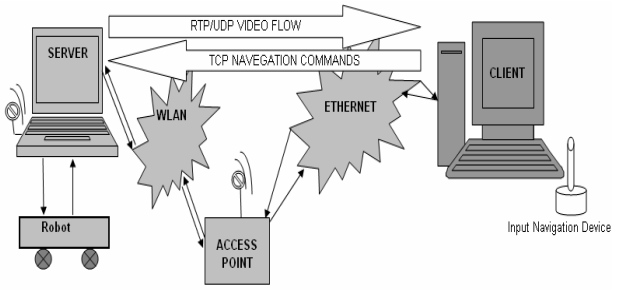

Fig. 1: Overview of the proposed telerobotic system.

The application is based on the client-server model. The client, running at control station, is in charge of requesting a new session to the server which is running at the ER1 robot. This new session consist of two data flows: (a) a video data flow, and (b) a navigation flow.

The first flow is composed of RTP (Real-time Transport Protocol) [17] video packets flowing from server to client, so the user at client station can get a video feedback of actual robot location. When the webcam captures a new video frame, the server performs a packetization process that produces a fixed size packet stream. This non-compressed video packet stream is delivered through RTP protocol. The client has to assemble each frame by arranging all received video packets in one common buffer. When the first packet of the next frame is received, the client displays current frame.

The navigation flow is composed of TCP packets that flow from client to server carrying navigation commands for the robot motion subsystem. When the user requests a navigation command through an input device, the client sends it to the server (very short messages). When this command arrives at server, it is processed and the robot is instructed to execute that command immediately.

3. End-to-End Delay Analysis

First of all, we are going to measure and analyze the end-to-end delay of the control loop, by doing an independent analysis of both video and navigation data flows. Then, we will be able to determine the overall control loop delay of the system under the best network conditions (best case). We will perform this study in three different network scenarios: (a) No network (IP loopback interface), (b) Fast Ethernet (only one unloaded switch between client and server), and (c) WLAN (as shown in Fig. 1) with the robot located just under the access point (maximum signal strength) and no movement.

So, we will analyze the delays in both data flows, in order to identify the different delay sources in our system. Then, we determine the overall control loop delay and its main contributors.

All measurements were taken over 100 consecutive video frames or 100 consecutive robot motions. The video source was setup to 160x120 YUV 4:1:1 at 10 frames per second. All delays are given in milliseconds. The default frame size is 28800 bytes, and packet size is 1KB.

3.1 Video Flow Delay Analysis

V1: It defines the elapsed time between the webcam captures one frame and the server receives that information. This delay depends on the frame size and the webcam hardware. So, we asked to the webcam manufacturer for the hardware capturing process delay. In our case, the capturing delay ranges from 5 to 20 milliseconds. Then, assuming that a 5 ms delay correspond with the lowest video format (160x120) and 20 ms with the largest one (640x420), we lineally determine the capturing delay for each video format.

V2: It measures the required time to send a whole video frame. This time corresponds to the packetization process and network delivery delay. It defines the elapsed time since the first video packet of one frame is delivered until the last one is received. We measure this delay by means of the Ethereal tool [15] running on a passive node that captures all packets traversing the wireless link. In table 1 we show the values of this delay with different video frame sizes in an Ethernet network connection. Table 2 shows the video frame delivery delay for different network connections (video frame size is 28800). Also, we measure the overhead of the RTP software library [20], being on average around 0.1 ms.

Table 1. Video frame delay with different frame sizes.

Size(bytes)	Av	Max	Min.	Std Dev
28800	26,84	30,53	21,61	3,25
57600	44,38	49,06	38,58	3,99
115200	81,09	91,19	75,7	4,69
153600	113,77	131,18	100,7	5,48
230400	168,99	194,24	149,65	6,3

Table 2: Video frame delivery delay on different networks.

Network	Av	Max	Min	Std Dev
Loopback	26,53	29,62	21,19	2,74
Ethernet	26,84	30,53	21,61	3,25
802.11b	27,51	32,44	21,92	4,46

V3: Frame reassembling and playing delay. It is the elapsed time since client receives the last packet of current video frame until the frame is showed to the user. This delay is very small and mainly depends on the OS drawing APIs. So, we consider it insignificant for our purposes.

3.2 Navigation Flow Delay Analysis

N1: The elapsed time since the user gives a navigation order and the navigation packet is sent. This delay is near zero, so we consider it insignificant in our study.

N2: It corresponds with the elapsed time between the client sends one navigation packet and the server receives it. We follow the same measurement procedures used to calculate V2 delay. Table 3 shows the obtained results.

Table 3: Navigation packet delay on different networks.

Network	Av	Max	Min	Std Dev
Localhost	0,11	0,33	<0,01	0,19
Ethernet	0,23	1,34	0,03	0,42
802.11b	0,47	2,92	0,09	0,97

N3: The elapsed time since a navigation command is received and the corresponding action is ordered to the robot. In that case we perform very little processing to receiving command, being this delay negligible.

N4: Measures the elapsed time since server sends the navigation order to the robot until it starts to executes it. To calculate this delay, we have to poll robot status to know the moment at which the robot starts to move. Depending on the specified moving action, the delays may change. So, we perform measures for different kinds of movement. The results are shown in Table 4.

Table 4: Motion delays for different kinds of movements.

Action	Av	Max	Min	Std Dev
Front-Back	22,3	24	18	2,2
Stop	12,2	14	9	2,1
Turn	18,3	21	14	2,5

3.3 Overall Delay analysis.

The user participates in the average delay of the whole system control loop of our telerobotic application, by means of the user reaction time. In this work, we do not take into account this delay.

After analyzing the delays of video and navigation flows, we observe two main sources of delay: Network delay (V2 and N3), and the robot reaction time (N4). The results shown in Fig. 2 correspond with the default video frame size (28800 bytes) at 10 frames per second. In order to know the maximum frame rate, we have to respect the V2 delay bound. So, the maximum frame rate will be 30 fps.

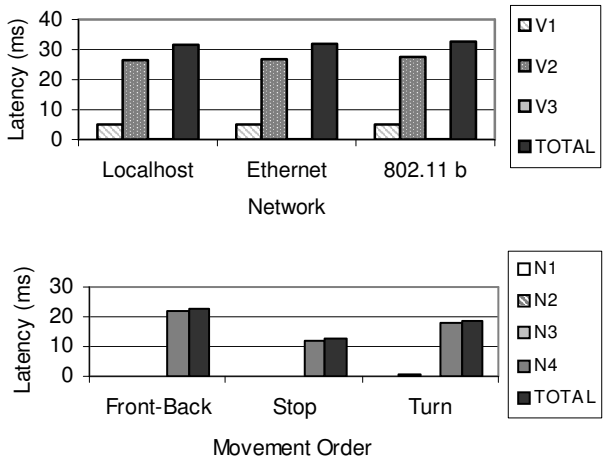

Fig. 2: Video (top) and navigation (bottom) delay results.

Finally, taking into account the worst case (WLAN network and front-back motion delay), the overall control loop delay is of 70 ms. This means that the proposed system requires 70 ms. to perform a teleoperation action.

4. Experimental tests

In this section, we define several network scenarios to determine the influence of other network-dependent factors like, background traffic, wireless signal strength,

and robot mobility patterns. Also, we analyze the interference degree between video and navigation data flows when competing for network access. The scenarios are the following:

(a) The robot is stopped at several places with different signal strengths.

(b) The robot is moving at constant velocity (10 cm/s) from high to low signal strength locations.

(c) The same scenario than (b) but with a discontinuous motion pattern with periodic (0.5 ms.) stop-ahead navigation command series.

(d) A scenario where the robot moves on a restricted area with moderate good signal strength (60-80%), with all kind of navigation commands and 1 Mbps background traffic. It represents a "real" scenario.

We configure our system with a video format of 160x120 YUV 4:1:1 (frame size of 28800 bytes) at 10 frames per second. The resulting video bitrate is around 2.2 Mbps (no compression). As performance metrics we will employ the ones supplied by RTP/RTCP report messages: Packet lost ratio and jitter; also we will use the average packet delay obtained through the Ethereal tool. The statistics are gathered every second. We have used again the Ethereal tool to capture packets and calculate the above metrics for TCP navigation packets. In order to inject traffic load into the WLAN, we used IP-Tools [19].

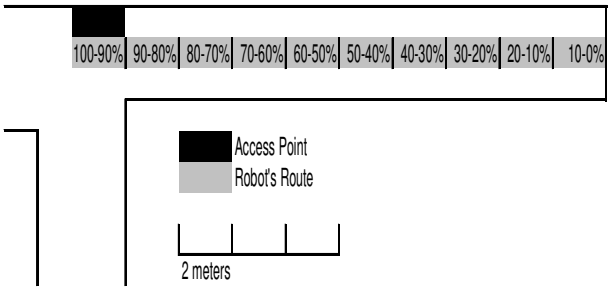

Fig. 3: Robot navigation path

In Fig. 3, we show the robot path that starts at the access point location and ends at a very low signal strength position. We have tagged the path with the observed signal strength values at each place.

4.1 Impact of node mobility

In order to determine the influence of mobility we use scenarios (a) and (b) with no background traffic. In Fig. 4, we show the packet lost ratio and jitter for video packet stream. As it can be seen, packet lost ratio lineally increase as signal strength decreases, reaching up to a 10% of lost packets at very low signal strength locations. Although not shown, the same occurs with average delay.

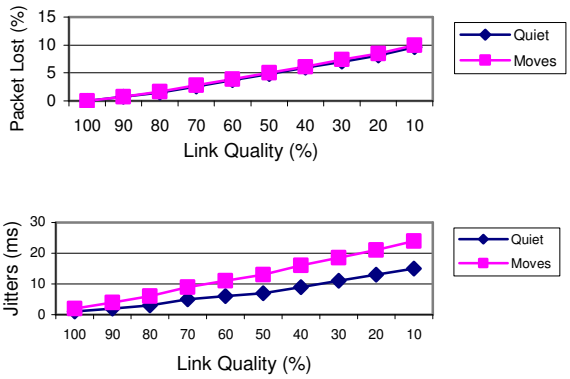

Fig. 4: Lost packets (top) and jitter (bottom).

However, jitter is significant in magnitude, as expected, but we find clear differences between both scenarios. In this case, the robot mobility increases jitter up to a 75% at locations with low signal strength.

4.2 Evaluation under background traffic

We repeat the previous experiments in scenarios (a) and (b) with background traffic in the WLAN network. For that purpose, we use a third node that injects traffic to the WLAN using the IP-Tools package. We have established different load levels: 0.5, 1 and 2 Mbps.

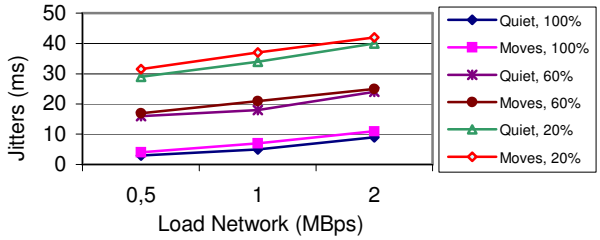

Fig. 5: Average jitter at different network loads.

In Fig. 5, we show the average jitter plots. The packet lost ratio and average packet delay exhibit a simple and lineal behaviour with respect background network load. So, packet lost increased up to a 6% from low to high background traffic levels. This increment is lineal and independent from link quality and robot mobility. Something similar happens with the average packet delay, which exhibits a lineal increase that does not depend on link quality. In the case of jitter, we found that without background load, robot mobility significantly increases the jitter. However, when there is background traffic, the differences in terms of mobility are very short.

4.3 Interference of navigation and video flows

Now, we proceed with another experiment in order to determine the interference degree between video and

navigation flows. For that purpose, we will use the scenario (c) assuring that no background traffic is present.

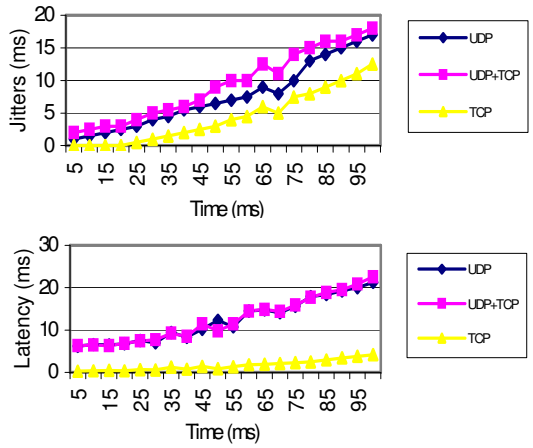

Fig. 6: Average jitter (top) and delay (bottom) of video and navigation packets.

In Fig. 6, we show the average packet delay and jitter. The video-only curves (UDP label) correspond to the results obtained on scenario (a), as reference. As it can be seen, there is a significant interference between video and navigation flows, increasing jitter, although the average packet delay is near the same. Notice that in scenario (c), 2 navigation packets per second are injected during the session (plus the associated ACK packets) representing a minimum fraction of the total traffic. Also, we observe that the average navigation packets delay is relatively low when comparing with average video packet delay. However, the latency/jitter relation is high, proving that navigation packets suffer more jitter than the video ones.

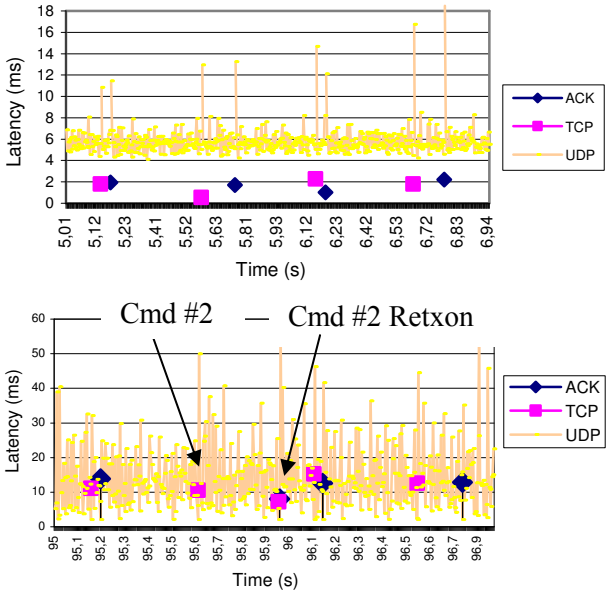

Fig. 7: Packet delay of TCP (data+ack) and UDP flows: 2 seconds view from the beginning (top) and the end (bottom) of session.

In Fig. 7, we show a more detailed view of packet delay at two session intervals: At the beginning of session where we have good signal strength from access point and at the end of the session where the signal is too weak.

We can observe that, in the first case, TCP and UDP packets compete for channel access, resulting in notorious delay peaks for UDP packets (most traffic source) when navigation packets require channel access (no packets are lost). At low signal areas, we have a similar behaviour but now packets are lost. In the figure we can observe that the second navigation command was lost, forcing TCP to retransmit it until the acknowledgement arrives. This event, results in a command delay increase of 320 ms. (more than 3 video frames), that limits the operability of the application in low signal areas.

4.4 Overall evaluation

Finally, we show another experiment based on scenario (d), in order to test this application in a real work area. We carry out our tests keeping link quality between 60% and 80%, injecting 1 Mbps background traffic, and sending navigation commands every 4 seconds. The robot executes motion orders at a constant speed of 10 cm/sec.

In Fig. 8, we show that under scenario (d), all metrics are very stable. However, the packet lost ratio, although does not seriously affect to the perceived quality, may be a problem if the application requires larger video frame sizes and higher video quality. We also notice a significant jitter and a near constant average delay.

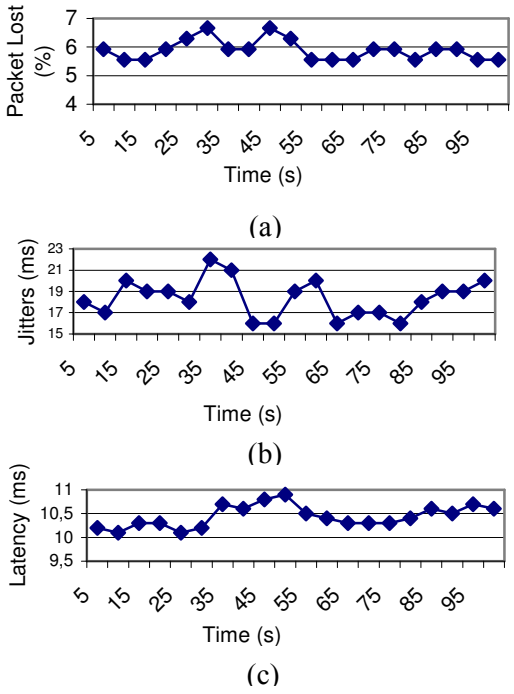

Fig. 8: Packet lost ratio (a), jitter (b) and delay (c).

5. Conclusions and future work

In this paper we study the feasibility of real-time telerobotic applications on standard IEEE 802.11b networks. At first place, we measure the end-to-end delay of the control loop for wired and wireless networks under the "best as possible" conditions. We observed that the overall delay of the system is around 70 ms (worst case) without considering the additional human reaction time.

Also, we have defined several scenarios in our IEEE 802.11b WLAN to evaluate the performance of our system when network conditions change (signal strength, node mobility and background traffic). Also, we have taken into account the consequences of the competition for link access between video and navigation data flows. The results show that, the most degrading factor is the wireless link quality that may increase the packet latency up to 5 times and enhance the drawbacks of navigation and video flows channel competition. Robot mobility only affects to packet jitter growing faster when robot is moving on towards low signal strength locations. With respect background traffic, its influence is lineal with respect to packet delay, jitter and lost ratio.

As future work, we are going to introduce a video compression system in the video flow stream, in order to work with larger video formats keeping as low as possible the required bandwidth. Also, we will deploy an IEEE 802.11e QoS WLAN in order to take profit of its traffic differentiated services for video and navigation flows.

6. References

[1] I.S.802.11, Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications. The Institute of Electrical and Electronics Engineers, Inc., August 1999.

[2] IEEE P802.11 – TG E, MAC Enhancements for Quality of Service http://grouper.ieee.org/groups/802/11/, 2002.

[3] R. Luo, W. Z. Lee, J. H. Chou, and H. T. Leong, "Tele-Control of Rapid Prototyping Machine Via Internet for Automated Tele-Manufacturing", IEEE Int. Conf. on Robotics and Automation, Vol. 3, pp. 2203-2208, May 1999.

[4] S. E. Everett, R. V. Dubey, "Model-Based Variable Position Mapping for Telerobotic Assistance in a Cylindrical Environment", IEEE Int. Conf. on Robotics and Automation, Vol. 3, pp. 2197-2202, May 1999.

[5] L. Hsu, R. Costa, F. Lizarralde, J. Soares, "Passive Arm Based Dynamic Positioting System for Remotely Operated Underwater Vehicles", IEEE Int. Conf. on Robotics and Automation, Vol. 1, pp. 407-412, May 1999.

[6] G. Hirzinger, B. Brunner, J. Dietrich, and J. Heindl, "Sensor-Based Space Robotics-ROTEX and Its Telerobotic Features", IEEE Trans. on Robotics and Automation, Vol. 9, No. 5, 1993.

[7] D. Kwon, K. Y. Woo, H. S Cho, "Haptic Control of the Master Hand Controller for a Microsurgical Telerobot System", IEEE Int. Conf. on Robotics and Auto., 1999.

[8] M. Tanimoto, F. Arai, T. Fukuda, and M. Negoro, "Force Display Method to Improve Safety in Teleoperation System for Intravascular Neurosurgery", IEEE Int. Conf. on Robotics and Automation, Vol. 3, pp. 1728-1733, May 1999.

[9] K. Brady, T. J. Tarn, "Internet-Based Remote Teleoperation", Proceedings of the 1998 IEEE Int. Conf. on Robotics and Automation, Leuven, Belgium, May 1998.

[10] Imad Elhajj et al, "Supermedia in Internet-Based Telerobotic Operations", IFIP/IEEE Int. Conference on Management of Multimedia Networks and Services,pp..359–372, 2001.

[11] Huosheng Hu, Lixiang Yu, Pui Wo Tsui, Quan Zhou, "Internet-based Robotic Systems for Teleoperation", Int. Journal of Assembly Automation, vol. 21, pp. 143-152. 2001.

[12] D. Andreu, P. Fraisse, V. Roqueta and R. Zapata, "Internet Enhanced Teleoperation: Toward a Remote Supervised Delay Regulator", IEEE International Conference on Industrial Technology , pp. 663-668, April 2003.

[13] K. Goldberg and R. Siegwart, "Beyond webcams: An introduction to online robots", MIT-Press, 2002.

[14] T. Fong et al. "Novel interfaces for remote driving: gesture, haptic and PDA", Autonomous Robots 11:75–85, 2001.

[15] N.M. Sgouros and S. Gerogiannakis. "Robot teleoperation environments featuring WAP-based wireless devices", Journal of Network and Computer Applications, 26:3, pp. 259-271, 2003.

[16] ER1 Personal robot System, Evolution robotics Inc., http://www.evolution.com/er1/

[17] RFC 1889, "RTP: A transport Protocol for Real-Time Applications", 1995.

[18] Greg Morris, Ed Warnicke, and Gilbert Ramirez, "Ethereal packet sniffing", Syngress Publishing, March 2004.

[19] KS-Soft IP-Tools, http://www.ks-soft.net/ip-tools.eng/

[20] Jori Liesenborgs, RTP C++ library, JRTPLIB 2.7. http://lumumba.luc.ac.be/jori/jrtplib