An Internet Request Server Architecture for Telecommanding the CONTROLAB AGV through Real Time Data and Image

G.H.M.B.Carneiro
NCE/UFRJ and IME/RJ
Brazil

E.P.L.Aude, H.Serdeira, J.T.C.Silveira, M.F.Martins
NCE/UFRJ
Federal University of Rio de Janeiro
P.O. Box 2324 - Rio de Janeiro - RJ
20001-970 - Brazil

E.P.Lopes
IM/UFRJ
Brazil

***Abstract-* The CONTROLAB AGV moves autonomously through any type of office environment, avoiding obstacles detected by vision and sonar sensors in real-time, and follows requests issued by any client station connected to the office network. This paper presents the CONTROLAB client-server architecture using the Internet, the techniques employed for wireless communication between the server and the AGV and the adopted approaches for compressing image information sent in real-time by the AGV to the server, for transmitting it to the clients and for decompressing it before displaying.**

I. Introduction

The CONTROLAB Autonomous Guided Vehicle (AGV) [1], shown in Fig. 1, can move autonomously within any office environment avoiding obstacles detected in real time and following instructions issued by client stations connected to the office network. These requests are managed and sent to the AGV by the Request Server using wireless communication. While operating, the CONTROLAB AGV sends compressed image information to the Request Server to allow remote monitoring of its operation by the client stations. This paper describes the client-server architecture, the techniques employed for wireless communication between the AGV and the Request Server and the adopted approaches for compressing and decompressing image information within the CONTROLAB AGV system.

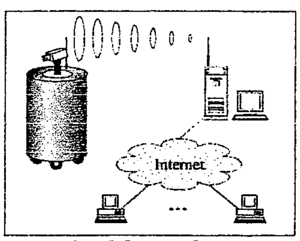

Fig. 1. The CONTROLAB AGV

Section 2 of this paper presents the techniques used for communication between the AGV and the Request Server and between the server and the clients. This section also discusses the techniques used for compressing and decompressing images. Section 3 describes the client-server system operation. The paper conclusions and directions for future work are presented in Section 4.

II. Communication Techniques

This section describes the adopted techniques for client-server communication on the Internet and for wireless communication between the AGV and the Request Server.

A. Client-Server Communication Using the Internet

The client-server communication on the Internet uses TCP/IP (Transport Control Protocol/Internet Protocol) for reliable messages and UDP/IP (User Datagram Protocol/Internet Protocol) for unreliable messages. The system implementation uses *sockets* [2]. All unreliable messages (UDP) are transmitted to the group of clients using the multicast protocol [3, 4]. This protocol is used for transmitting information in real-time. For instance, after receiving compressed image information from the AGV, the server distributes this information to the clients using the multicast protocol. With this protocol, a single copy of the data is sent to the group of clients and the sender does not need to wait for acknowledgment messages.

B. Wireless Communication between the Server and the AGV

The wireless communication system has been implemented with the use of the RPC-433-A - Radiometrix radio equipment which is able to transmit information at 40 kbps half-duplex in packets consisting of 1 to 27 bytes. Due to the codification used by the RPC, every 27-byte packet is transmitted in 13.8 ms. Therefore, the effective data transmission rate is limited to 15652 bps.

From the seven OSI-ISO (*Open Systems Interconnection - International Standards Organizations*) reference model layers, only four have been used in the development of the wireless communication system: the *Physical* layer, implemented by the available hardware; the DLC - *Data Link Control* layer, in which the *Medium Access Control* is implemented by the RPC and the data link itself deals with the algorithms to provide reliable communication between two computers; the *Presentation* layer, which deals with the compression of images captured by the CONTROLAB AGV; and, finally, the *Application* layer, which deals with high-level information transfer between the AGV and the Request Server. Fig. 2 shows the implemented layers in the adopted wireless communication protocol.

0-7803-5491-5/99/$10.00

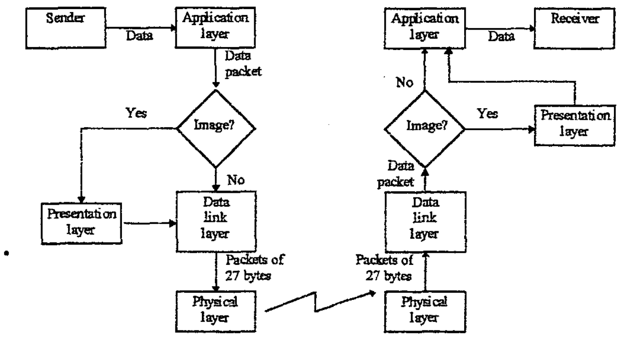

Fig. 2. Wireless Communication Protocol Layers

The main task of the Data Link Control layer is to ensure that the packets are delivered to the destination free of errors and in the correct order. To attain this degree of reliability, the ARQ (*Automatic Repeat Request*) [5] technique, which retransmits the packets for which no acknowledgment message is received, has been used. This technique can be implemented in different ways. The simplest one, *stop-and-wait*, tries to ensure that a packet has been correctly received before sending the next one. This goal can be achieved by numbering each packet which is sent. This way, the sender, when sending packet 0 (SN=0, Sequence Number), waits for its confirmation (RN=1, Request Number) for some time (*time-out interval*). If the confirmation is received, packet 1 is sent. Otherwise, packet 0 is retransmitted by the sender.

Fig. 3 shows the format of the data packet used in the Data Link Control layer within the CONTROLAB wireless communication system.

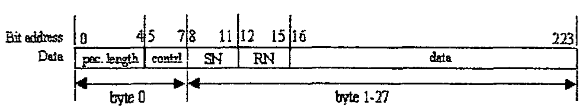

Fig. 3. Data Link Control Layer Data Packet Format

The transmission of some types of information does not need to be as reliable because either they are not essential for the system operation or they present a high enough repetition rate to ensure that if the server loses a message it will receive it again soon after. The unreliable data transmission protocol is simliar to the one previoulsy described. However, the receiver does not send acknowledgment packets and the sender does not wait for them.

Both the AGV and the Request Server need to know the meaning of the data they are transmitting or receiving. Therefore, the first packet to be transmitted is a control packet containing an identifier of the type of the message to be transmitted. Depending on the received packet data type, the application layer directs the packet to the appropriate sub-system to process it. When an image packet sent by the AGV is received by the Request Server, it is firstly sent to the presentation layer to be decompressed. After that, the application layer sends it to the image display.

Whenever a computer sends control data representing the beginning of a message, it has to wait for the arrival of the same data sent by the receiver to indicate that it is ready for communication. At the end of the transmission of a reliable message, an end-of-message control packet has also to be sent and received to indicate that the complete message has been transmitted. The format of the message first packet for both a reliable and an unreliable transmission is shown in Fig. 4. When an unreliable message packet is received, the remaining packets must be received without interruption in intervals of at most 1 second. The total number of packets to be received is defined in the message size field of the first packet, as shown in Fig. 4. If the number of received packets is not the expected one, the receiver discards the received data.

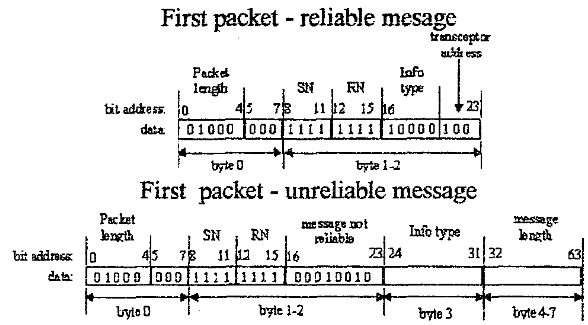

Fig. 4. First Packet Formats

The transmission rates achieved with reliable communication are shown in Table 1. The highest achieved rate, 2406 bps, represents 15% of the maximum rate (15652 bps) that could be reached as previously mentioned.

TABLE 1 - RELIABLE COMMUNICATION TRANSMISSION RATES

Length (bytes)	Transmission time	Transmission rate
1	170 ms	47 bps
26	170 ms	1223 bps
27	270 ms	800 bps
1000	3.46 s	2312 bps
2000	6.65 s	2406 bps

The adopted approach for implementing real time image compression and decompression follows the H.263 specification [6]. The CONTROLAB AGV image transmission system requires all the processing to be performed in real time. Therefore, an image should be captured, compressed and transmitted by the AGV, transmitted by the server to the clients through the Internet using multicast, decompressed and displayed by the clients in real time. A CONTROLAB AGV image is represented by a 242x199 array of pixels with gray levels in between 0 (black) and 255 (white). By using the H.263 specification, several simulation experiments have been performed to compare different coding options such as Arithmetic Coding [7], PB frames, a number of P frames between two I frames, etc. The experiment results, considering the compression of a sequence of 50 frames, are shown in Table 2.

TABLE 2 - IMAGE COMPRESSION EXPERIMENTAL RESULTS

	1I+4P	1I+9P	50I	PB	Arithm. coding
Elapsed time (s)	120	129	34	195	120
Compression rate	31.18	44.34	21.64	64.52	30.80
Transmission rate (bps)	4063	2659	20664	1209	4111
Image/s	0.47	0.39	1.47	0.26	0.42

Considering that the wireless communication in use transmits data at 40 kbps, a compression technique based only on I (Intracoded) frames is the one that produces by far the best result for real time applications. Its compression rate was over 20 and the time it spent on the compression of a single image was around half a second on a Pentium @ 233 MHz. Therefore, this compression approach has been adopted within the CONTROLAB AGV system.

At the reception end of the image transmission, a variable length buffer is used to store the compressed image since its size is not known a priori.

III. THE CLIENT-SERVER SYSTEM

The client-server system has been implemented according to the object-oriented paradigm proposed by Coad/Yourdon [8]. The goal of the system is to allow the communication between the clients and the server and between the server and the AGV. A simplified object diagram of the system is represented in Fig. 5.

The client/server object can be either a client or a server. The objects in this class are the only ones which exchange messages with the object in the form class, which is the screen with text boxes, image boxes, buttons and menus.

The wireless connection server implements the protocol for message communication with the AGV. The reliable connection server manages the connections to each client (each connection is an object of the reliable connection class that is not shown in Fig. 5). The unreliable connection server manages the connections which use the message transmission to multicast groups. This object manages 4 other objects, each one representing a connection to one multicast group: IP+Port group, image group, AGV position group and user/request list group. The request manager accepts the addition of new requests and the elimination of requests which have not started to be executed by the AGV yet. Within the client, the request manager is simpler because it only needs to store requests sent by the client. For the client to cancel a request, it must select the request and send this information to the server. The server checks if that request is under execution by the AGV and cancels it if not. The information on the cancellation of the request is sent as a reliable message to the client and the updated list of requests is sent through an unreliable message to the user/request list multicast group.

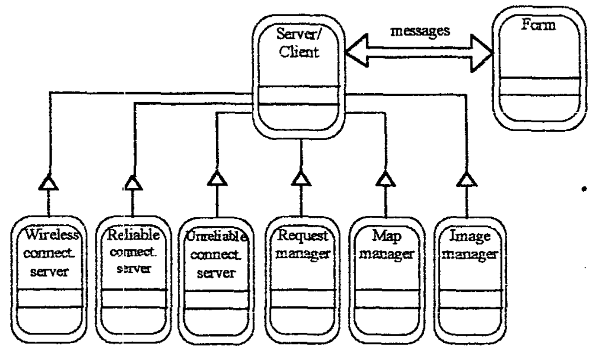

Fig. 5. Client-Server System Object Diagram

The map manager can read a map sent by the CONTROLAB Architect, a tool for describing the AGV working environment floorplan, and represent on it the current AGV position. The image manager is activated when the client/server receives a compressed image. It performs its decodification in order to display it on the screen.

All data sent by the Internet or by wireless communication need an identifier to describe the packet data type. When the Internet is used, the first 8 bits of the message represent the identifier as shown in Fig. 6. Depending on the identifier code, the client/server directs the data to the suitable object to process them. This message analysis is performed by the message manager, which is an object belonging to a reliable or unreliable connection object.

message_ID (8 bits)	data	EOF

Fig. 6. Message Format in the Internet

Fig. 7 summarizes the types of messages which are exchanged between the systems. Each unreliable message communication represents a different multicast group.

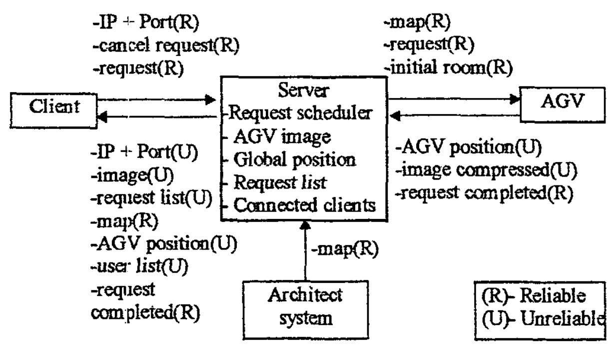

Fig. 7. Message Exchange Between the Systems

In order to send a request to the AGV, the client selects on the map shown on its screen (Fig. 8) the final position to be reached by the AGV as a result of the request. This event causes a system message to be sent to the client object which, then, sends a system message to the request manager and to the reliable connection objects. As a result, the request manager adds the request to its list and the reliable connection object sends the request through the Internet to the server. Then, the server object sends a system message to the reliable connection server, which reads the message from the Internet, checks that it is a request and sends a system message back to the server. As a consequence, the server sends system messages to: the map manager, which represents the new request on the map; the request manager, which inserts the new request in the queue; the unreliable connection manager, which sends the request to the user/request list multicast group; and the wireless connection manager (if the request is the first one in the queue), which sends the new request to the AGV. When a message in the user/request list multicast group is detected by the client, a system message is sent to the corresponding unreliable connection manager, which reads the Internet message and sends a system message to the client object informing that it is a request to create a representation on the map. Finally, to complete the whole operation, the client object directs this message to the map manager.

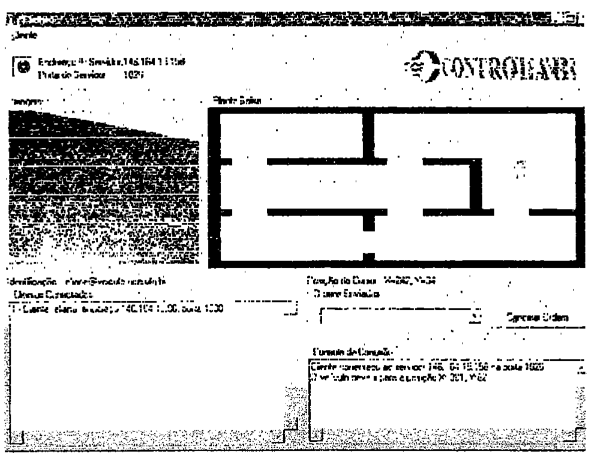

Fig. 8. The Client Screen

During the system initialization phase, the user which is operating the Request Server must start the Internet server to allow the connection of clients. In addition, the user must send to the AGV the environment map and the room in this map where it is initially located. With this information, the AGV starts to send its position to the server which distributes this information to the clients. The AGV also starts to send compressed images captured by its camera. The server, after receiving these images, sends them to the image multicast group. The images are only decompressed by the clients before displaying on the screen. Fig. 8 shows a client screen.

After this initialization phase, the clients start to send requests to the AGV. Each request is stored by the server with the corresponding client identification. The requests are sent to the AGV on a first come first served basis.

IV. Conclusions and Future Work

The implementation of a client-server system on the Internet for the high-level tele-operation of an AGV enables its utilization from any computer with an Internet connection. Therefore, it is possible to isolate the AGV operator from hazardous environments where the AGV may be performing its tasks. The only limitation of the system lies on the fact that some routers do not support the protocols for the implementation of multicast groups.

An improvement of the wireless communication transmission rate is planned as a future work in order to allow the transmission of image information at a higher rate. Another future goal is to design a more efficient task planner aiming at minimizing power consumption by the AGV.

Acknowledgment

The authors would like to thank CNPq/RHAE, Brazil, for the support given to the development of this research work.

References

[1] E.P.L. Aude, G.H.M.B. Carneiro, H. Serdeira, J.T.C. Silveira, M.F. Martins, and E.P. Lopes, "CONTROLAB MUFA: A Multilevel Fusion Architecture for Intelligent Navigation of a Telerobot," *Proc. 1999 IEEE International Conference on Robotics and Automation*, May 1999, Detroit, USA, pp. 465-472

[2] D. Comer, *Internetworking With TCP/IP Vol 1: Principles, Protocols and Architecture*. 3rd ed., Prentice-Hall, Englewood Cliffs, NJ, USA, 1995.

[3] S. Deering et al., *An Architecture for Wide-Area Multicast Routing*, University of Southern California Technical Report 94-565, November 1996.

[4] B. Quinn and D. Shute, *Windows Sockets Network Programming (1.1 and 2.0)*, Addison-Wesley, USA, 1996.

[5] D.P. Bertsekas and R. Gallager, *Data Networks*, 2nd ed., Prentice Hall, USA, 1991.

[6] International Telecommunication Union, *ITU-T Recommendation H.263. Line Transmission of Non-Telephone Signals. Video Coding for Low Bitrate Communication*, 1995.

[7] N. Abramson, *Information Theory and Coding*. McGraw-Hill, New York, USA., 1963.

[8] P. Coad and E. Yourdon, *Object Oriented Analysis*, Prentice Hall, Englewood Cliffs, NJ, USA, 1991.