Design Considerations for Stand-alone Haptic Interfaces Communicating via UDP Protocol

Ryan M. Traylor,[1] Danny Wilhelm,[1] Bernard D. Adelstein[2] and Hong Z. Tan[1]
[1] *Haptic Interface Research Laboratory, Purdue University, West Lafayette, IN 47906*
{traylorr, dwilhelm, hongtan}@purdue.edu
[2] *NASA Ames Research Center, Moffett Field, CA 94035*
bda@eos.arc.nasa.gov

Abstract

This work was motivated by the need for high speed communication between a stand-alone haptic interface and a computer running haptic rendering algorithms. This paper describes our recent work on the use of User Datagram Protocol (UDP) over Ethernet as a communication channel between a remote computer and a custom embedded controller built for a 3 DOF force-feedback haptic interface (the 3-DOF ministick). The results and observations from three experiments designed to test the use of detailed timing diagrams representing theoretical models are contrasted with experimental data in order to isolate potential problems and verify predicted models. Details of the hardware that enabled a haptic update rate of 3800 Hz are presented. Important information learned through the testing process is presented to the reader in order to help reduce development time for other systems in which Ethernet is desired as a communication channel.

1. Introduction

As haptics technology becomes more mature, there has been much work in recent years toward developing stand-alone and plug-and-play haptic interfaces. Instead of using proprietary ISA or PCI bus cards to interface a haptic interface to a PC, the trend is to use the parallel or USB 2.0 port for inter-device communication. Recently, we have experimented with User Datagram Protocol (UDP) over Ethernet for a 3-DOF force-feedback hardware platform, and succeeded in a haptic update rate of 3800 Hz.

The advantages of UDP include high bandwidth, design simplicity, scalability, low communication overhead (no need for packet acknowledgment) and fixed short (eight byte) header length.

2. Hardware platform

The embedded controller is based on an 8-bit ATMEGA128 [4] microcontroller running at 16 MHz. It is wired to a Packet Whacker [5] 10 Mbps half-duplex Ethernet controller card through the microcontroller's common communication bus. An extremely streamlined UDP/IP Ethernet stack is implemented in the firmware stored onboard the microcontroller. The Ethernet stack is loosely based on Fred Eady's full TCP/IP stack [6], with modifications made for the use of static IP headers, but without checksum calculations or TCP protocol management.

The haptic interface (Fig. 1) housing the embedded controller is a point-contact joystick that couples three rotationally actuated five-bar loops to 3-DOF endpoint force and motion. The design is based on a unique fully-parallel, 10 rigid link, 12 revolute joint spatial architecture [7] that affords the structural stiffness typically associated with parallel mechanisms, but with the range of motion approaching that of serial linkages. Both the forward and inverse kinematics for this device can be expressed and solved in closed form [8]. In the current finger-scale implementations [9-10], the parallelogram's upper and lower arm links are each 50.8 mm (2 in) in length. The device's usable, interference-free workspace occupies a hemispherical region roughly 9 cm by 9 cm by 6 cm.

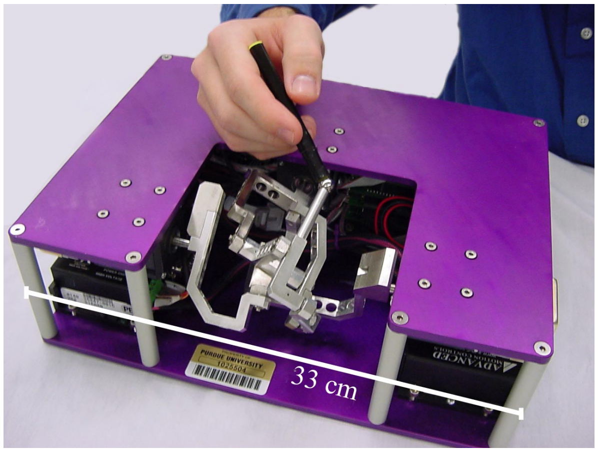

Figure 1: The 3-DOF ministick.

3. Experiments with UDP[1]

3.1. Client mode

In this experiment, the embedded controller's firmware is modified such that it waits to receive a packet of actuator commands from a remote computer, processes the commands, and then sends a packet of position readings back to the computer. Since the embedded controller in this case relies on the computer to initiate and sustain communication, this mode of operation is called the *client mode*. For this and all subsequent experiments, the remote computer is equipped with a 2.8 GHz Pentium processor and 3Com 3C940 Gigabit Ethernet card and running the Windows XP Operating System.

We found that it is theoretically possible to push the update rates of the system to around 3800 Hz under certain conditions, and that the embedded controller is the limiting factor in determining the maximum update rate due to its processor running close to full capacity. A typical timing diagram is shown in Fig. 2.

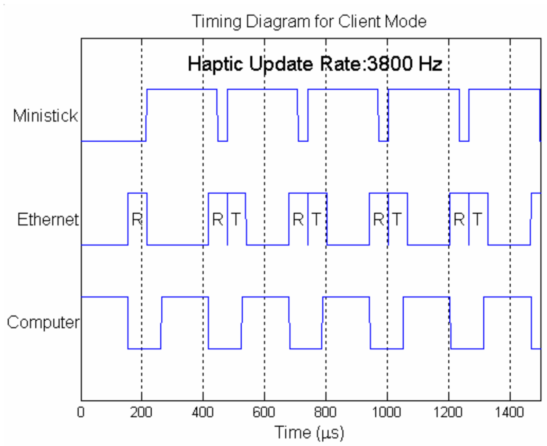

Figure 2: Timing diagram for client mode operation running at 3800 Hz. The "R" and "T" labels respectively indicate from the embedded controller's perspective whether the packet is being received or transmitted.

3.2. Server mode

In this experiment, the embedded controller's firmware transmits successive position packets at a fixed interval, then processes actuator command packets sent in response from the remote computer as they arrive. Since the embedded controller initiates and sustains communication with the remote computer, this mode of operation is called the *server mode*. A maximum update rate of 3800 Hz was achieved.

3.3. Distributed mode

Although the embedded controller has proven to be the limiting factor in determining the maximum update rate of this system, it is feasible that in the future a different embedded system equipped with a much faster processor and Ethernet interface could actually cause the computer to be the limiting factor. It is also conceivable that a very complicated virtual environment could cause the computer to take much longer than the 155 μs allotted in both the client-mode and server-mode experiments thereby causing the computer to become the limiting factor. A novel way to alleviate these potential problems is by distributing the workload of processing the Ethernet packets among more than one computer using the scalability afforded to the system by Ethernet communication protocol. Due to the fact that in this configuration the packets are distributed for processing over more than one computer, this mode of operation is called the *distributed mode*. Again, an update rate of 3800 Hz was achievable in the distributed mode.

4. Conclusion

UDP protocol over Ethernet has been shown to be a viable method of communication between haptic devices and remote computers. A haptic update rate of 3800 Hz was shown to be an approximate upper bound for reliable operation in our system. This rate could be easily increased by using a faster microcontroller, a 100 Mbps Ethernet controller with full-duplex capabilities, and a more streamlined Ethernet stack.

Acknowledgments

This work was supported in part by an NSF award under Grant 0098443-IIS, and in part by NASA under award no. NCC 2-1363. The mechanical hardware for the 3-DOF ministick was purchased from UC Berkeley.

References

Omitted due to limited space.

To see a complete list of references and to read an expanded version of this article, please go to "http://www.ece.purdue.edu/~hongtan/Hongtan-pub/PDFfiles/Traylor_etal_WHC05.pdf"

[1] Due to limited space, many important details are omitted here. The URL for an expanded article is provided at the end of this page.