

# Efficient Transport Protocol for Networked Haptics Applications

Raul Wirz<sup>1</sup>, Manuel Ferre<sup>2</sup>, Raul Marín<sup>1</sup>, Jorge Barrio<sup>2</sup>,  
José M. Claver<sup>3</sup>, and Javier Ortego<sup>2</sup>

<sup>1</sup> Computer Engineering and Science, University Jaume I, Castellon, Spain  
{wirz, rmarin}@icc.uji.es

<sup>2</sup> Automatics, Electronics Engineering and Industrial Computers Department,  
Universidad Politécnica de Madrid, Madrid, Spain  
{mferre, jbarrio, jortego}@etsii.upm.es

<sup>3</sup> Computer Science Department, University of Valencia, Valencia, Spain  
jclaver@uv.es

**Abstract.** The performance of haptic application is highly sensitive to communication delays and losses of data. It implies several constraints in developing networked haptic applications. This paper describes a new internet protocol called Efficient Transport Protocol (ETP), which aims at developing distributed interactive applications. TCP and UDP are transport protocols commonly used in any kind of networked communication, but they are not focused on real time application. This new protocol is focused on reducing roundtrip time (RTT) and interpacket gap (IPG). ETP is, therefore, optimized for interactive applications which are based on processes that are continuously exchanging data. ETP protocol is based on a state machine that decides the best strategies for optimizing RTT and IPG. Experiments have been carried out in order to compare this new protocol and UDP.

**Keywords:** Transport protocols, internet protocols, haptics, bilateral control, controller.

## 1 Introduction

Haptic devices are characterized for interchanging bidirectional flows with the exterior. On the one hand, they are used to generate movement references and, on the other hand, they serve as force reflection and pairs, generated upon interacting with virtual objects. Good work conditions could be defined through a communication channel that transmits data without time delay in a speed of approximately 200-600 kbits. This work frequency allows the efficient sending of information from a haptic interface to other processes. A haptic interface does not necessarily require wide bandwidth compared to other types of applications such as video transmission. However, a constant frequency without significant delays or loss of data needs to be maintained. As data transmission

conditions worsen, a significant loss of environment perception through a haptic interface is given [1]. Communication worsening can be induced due to such factors like speed reduction in the transmission, loss of data and delays in data transmission.

Regarding the communication architecture in haptic applications, it can be centralized or distributed. In many cases, it is centralized which means that one, two or three haptic devices at maximum are connected to the same computer through a port with an improved data transmission (USB, FireBox) or to the own computer internal bus. In this way, an increased and stable data transmission is assured between the application processing the virtual environment and the haptic interfaces.

Nowadays, it draws a lot of attention to the development of distributed applications where haptic interfaces are connected to different network places. In the case of a private network with a preconfigured number of computers and applications, it is possible to guarantee the right working of communications within the network. On the contrary, in the case of an open network where interfaces are placed relatively far to the network some new problems regarding data transmission appear. Such problems are caused by network jams, since they bring about packet losses, transmission delays of data stored in buffers and speed reduction in data transmission.

Protocols traditionally used to develop distributed applications are TCP [2] and UDP [3]. In applications where the maximum bandwidth is to be fully exploited, the UDP protocol is used. This protocol is commonly applied in teleoperation, robotic or control applications. To the date, little attention has been drawn to the design of transport protocols which optimize data transmission for this kind of interactive applications. The unique known developments till now are Real-Time Network Protocol (RTNP) [4] and Interactive Real-Time Protocol (IRTP) [5] but with limited results. The first model includes a priority mark in the packets, leaving the network jam problem unresolved. And the latter one has only its implementation on Linux available, making its use quite limited [6].

The work introduced by this paper evaluates a new transport protocol known as Efficient Transport Protocol (ETP) which aims at optimizing the available bandwidth within a network so that the highest number of packets is sent without affecting each packet RoundTrip Time (RTT). Furthermore, the device controller is reported by ETP about the actual RTT at any time, a relevant feature of this protocol. It means a significant step forward since haptic device controller can make decisions to counteract the communication delay effect.

Next sections in this article are organised as follows. Section 2 describes in detail the ETP protocol working, which is governed by a state machine acting according to the RTT and IPG of each packet. Section 3 introduces the ETP protocol advantages for haptic applications, the controller stability and the accuracy in data reflection are to be highlighted. In section 4 the experiments carried out with the ETP protocol are described. Finally, conclusions of the present work are drawn in last section.

## 2 Efficient Transport Protocol (ETP) for Interactive Applications

As we have seen, the protocols available for interactive Internet applications are not good enough so that a new protocol has been designed, which provides better performance and it is well integrated with the control algorithm that manages the task. The main objective of this protocol is providing the minimum RTT (Round Trip Time) and the maximum frequency (i.e. minimizing the Interpacket Gap - IPG) in a point to point connection.

### 2.1 Preliminary Considerations

Two important aspects must be controlled when an interactive application is running: The IPG (Interpacket Gap) time that it is the time between two packets and the RTT (Round Trip Time) that is the time elapsed for a packet to go to the receiver and back again. These two aspects are the most important when a new protocol is designed to use in interactive applications.

**RTT Behavior.** In interactive applications we can find at least two different flows: The first one (i.e. flow 1) goes from the device A to the device B, and it contains the orders or commands. Second, the other flow (i.e. flow 2) goes from the device B to the device A providing feedback. The time that takes for a packet to reach the device B (i.e. flow 1) and come back to the device A (i.e. flow 2) is called RTT, and for bilateral interactive applications it is important to obtain the minimum possible value.

Bandwidth and RTT are very much related terms so that when the consumed bandwidth increases the network can get some congestion and the RTT could increase due to the packet loss or because the buffers of the intermediate routers get more packets to deliver. If there is no congestion the RTT would get its minimum, which is excellent for interactive applications. Moreover, for an interactive application it is normal to set the same consumed bandwidth for the flow 1 and the flow 2 (i.e. using UDP or TCP). However, there are situations where the available bandwidth for a flow should be different than the available bandwidth of the other flow. For example, using asymmetric connections, where the uploading bandwidth is different than the downloading one. Even, the congestion could be detected only in one of the flows. To solve this situation and obtain a possible minimum RTT we propose that flow 1 will be independent from the flow 2, in others words, the bandwidth of flow 1 must be independent from bandwidth of flow 2. Bandwidth in flow 1 and bandwidth in flow 2 are controlled changing the IPG in both sides.

**IPG Control.** It is important to have a minimum IPG to have a correct interactive application. We could express as well the sending frequency as the bandwidth of a particular flow.

Flow 1 and flow 2 are two independent flows where each flow has its own bandwidth, which must be controlled. A way to perform this bandwidth control

would be using a sending window. This is actually what TCP does. However, this kind of congestion control does not guarantee a constant sending rate, due to the fact that the packets are sent as fast as possible for the current sending window and then the protocol waits for the ACKs to return. For interactive application it is more interesting using a rate based control mechanism, which is performed through the IPG Control [7]. To control the IPG we propose having an asymmetric technique, which consists of the device A controlling the IPG of the device B, and the device B controlling the IPG of the device A. This allows the system to better adapt to asymmetric scenarios like ADSL based Internet access.

### 2.2 Protocol Description

The only requirement that the protocol needs is setting conditions for the minimum and maximum IPG limit. The protocol will try to obtain the minimum IPG limit (i.e. maximum frequency) by controlling the network congestion using the transmission rate. For that, the protocol will take into account the IPG and the transmission time.

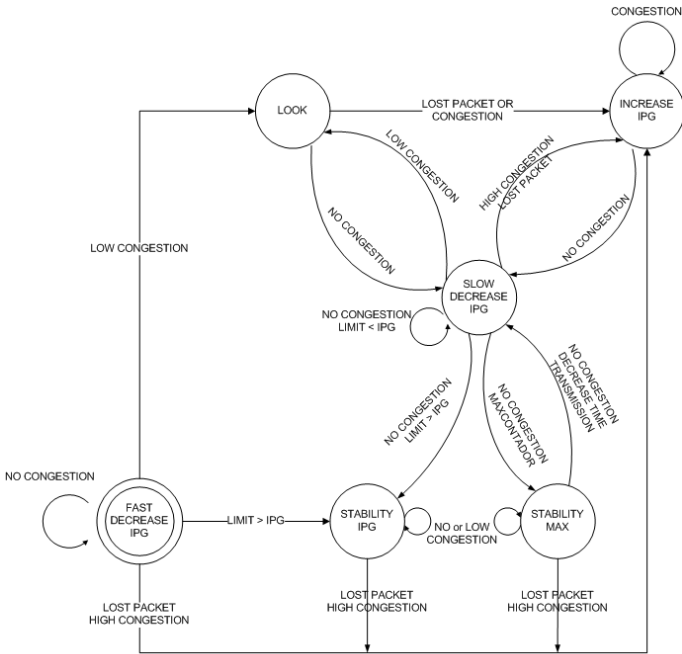


Fig. 1. Protocol phasers

Transition between different states can be observed in figure 1, where changes are defined in each cycle (A cycle is set by the transmission time if the transmission time is more than the IPG required or it is set to a fixed value if the

transmission time is lower than the IPG required). The IPG is controlled by the protocol to control the packet injection in the net.

$$IPGProposed = \frac{TimeEstimated}{Counter} \quad (1)$$

The protocol has six different states and they are described as follows:

- FAST DECREASE IPG: Fast Start State. This is the initial state where the protocol has a fast increase of the sending ratio (i.e. fast decrease of the IPG). In this state the transmission rate increases until congestion is detected or the IPG limit sets by the user is reached.

$$Counter_{t+1} = Counter_t + 1 \quad (2)$$

- LOOK: Wait State. The counter value is saved and if congestion is detected in the following cycle, MAX value will be updated with the counter value that was previously saved. Max value is the counter value where the network can find congestion.
- INCREASE IPG: Penalize State. In this state the IPG is increased to obtain less transmission rate because congestion or a loss packet are detected. The new IPG value is set by the following formula where Gamma depends on the congestion quantity: 1% if it is low, 20% if it is high and 50% if it a loss packet.

$$Counter_{t+1} = Counter_t - \gamma * Counter_t \quad (3)$$

- SLOW DECREASE IPG: Positive State. In this state any congestion is detected and the transmission rate is increased slowly using the following formula:

$$Counter_{t+1} = Counter_t + \frac{1}{Counter_t} \quad (4)$$

- STABILITY IPG: IPG Stability State. This state represents a situation where the IPG limit set by the user is reached and any congestion is detected. The objective of this state is keeping the actual IPG as closed as possible to the IPG limit.
- STABILITY MAX: MAX Stability State. This state represents a situation where congestion will be detected in the near cycle. MAX is a last counter value where congestion was detected and where a new congestion has a high possibility to be detected so, when the MAX value is reached the transmission time is observed. If the transmission time is equal or less in the MAX level, it is that the congestion that was in that level has disappeared and the IPG value can be increased to obtain the IPG limit.

### 3 Efficient Transport Protocol (ETP) for Haptics Applications

Significant improvements can be obtained from haptics applications by using ETP protocol. Main advantages of this protocol are RTT awareness and IPG

optimization. They allow relevant benefits for haptic networked applications, such better algorithms for device controllers and increased virtual environment perception. These two points will be discussed in following sections.

### 3.1 Control Considerations Using the ETP Protocol

Force and movement are processed by haptic controllers in order to implement impedance or admittance behaviour. These controllers can be modelled according to the next figure. IPG time represents the frequency of package transmission. Communication channel between haptic controller and remote process applies a delay of  $N_1$  and  $N_2$  packet interchange in each direction. These delays are due to network buffers, moreover they can be different for packets that are sent or received. Therefore, network transmission can be defined as  $N_1$  packets that go from the haptic controller to the remote process and  $N_2$  packets to the opposite direction.

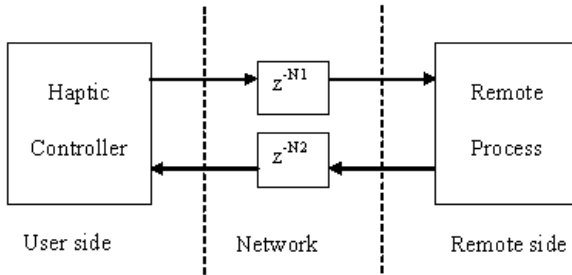


Fig. 2. Flow of packets in the network

RTT represents the frequency that closes the control loop of the haptic controller. This frequency has a significant impact in the stability of the haptic controller. Delays are well known to provoke unstable systems. However, knowing the RTT, different strategies can be implemented in order to maintain the system stable. A first strategy can be done by limiting control gains according to RTT variations. It is important to take into account that in presence of big delays, haptic perception is significantly deteriorated in spite of having a stable system.

RTT is related to the number of packages and IPG according to next equation. This relation is equal when IPG is the same for all transmitted packages at that very moment; if packages are generated by different IPGs, then the relation is an approximation.

$$RTT = IPG \cdot (N_1 + N_2) = IPG \cdot N \quad (5)$$

The TPI protocol optimizes IPG in order to minimize RTT. If too many packages are generated, then IPG decreases, but the product of  $IPG \cdot N$  increases. On the other hand; if IPG is longer, it can achieve the same value of RTT. In

such case, only one packet is transmitted by the network. Therefore, the goal of the ETP is to reduce IPG in order to increase the number of packets that are transmitted by the network, and also avoid the increase of RTT. From the haptic controller's point of view, the time required to close the control loop (RTT) has to be the least possible time, but with the most number of samples (packets) to be transmitted.

### 3.2 Strategies to Improve the Haptic Perception

Communication time delays provoke a significant degradation of haptic perception. Operator haptic perception has been evaluated in terms of transparency or reflected impedance [8],[9] for telerobotics applications. Some works have evaluated how perception is deteriorated according to the communication time delay including factors such as variable time delay and lost packets [10],[11],[12],[1]. As conclusion of these works inertias and frictions are superimpose to the desired haptic perception when communication delays appears. Stiffness is also modified by delays, stiffness is commonly perceived softer when time delay increases.

RTT of packets is provided by ETP protocol. This information can be used in order to correct the effect of the communication time delays. Strategies in this direction can be investigated but stability conditions must be taken into account. It is well known that stability and transparency have opposite effects. Therefore, any correction in the haptic perception has to guarantee the haptic device stability.

Haptic devices can be adjusted to optimize the perception under some conditions; for example, adjusting the controller parameters for a given time delay [13]. In this case, RTT awareness can be used in order to obtain the best controller for any communication time delay. It allows a less degradation of haptic perception due to delays.

## 4 Experiment

The experiment is based on a real system called MasterFinger [14] that uses the UDP and ETP protocols. To prove our protocol, the experiment was implemented with the NS2 Simulator. Main components are shown in figure 3.

The masterfinger sends the position to a PC (flow 1) and the PC returns the force values to the masterfinger (flow 2) meanwhile the PC sends to the

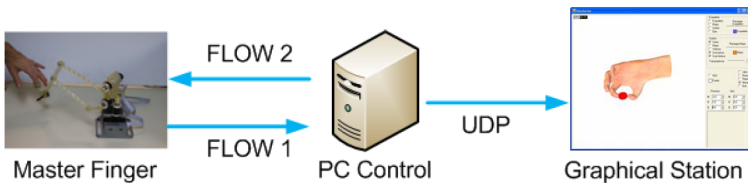
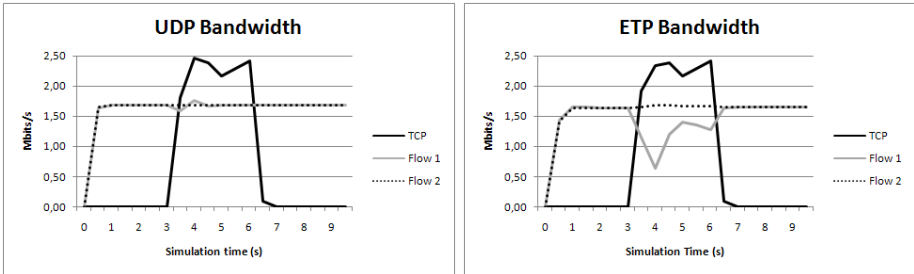


Fig. 3. Experiment Elements

graphical interface the information to paint the scenario (UDP flow). UDP and ETP protocol in flow 1 and flow 2 will be evaluated in the experiment. In this simulation, a minimum IPG time of 1 millisecond is required. However, any maximum IPG time is required. Also, the congestion is located in the same direction than flow 1 and the minimum delay is 10 milliseconds.

#### 4.1 Experiment Results

UDP and ETP protocols have different behaviors. Bandwidth behavior can be observed in figure 4. The IPG time and transmission time can be analyzed in table 1.



**Fig. 4.** UDP Bandwidth and ETP bandwidth in the NS2 Simulator

Flow 2 has simple variations because the ACK packets of the TCP protocol are not enough to do congestion so flow 2 has similar values (bandwidth, IPG, Transmission time) in both protocols. However, the flow 1 has significant variations because packet information of TCP protocol are doing congestion.

**Table 1.** Average times in experiments (milliseconds)

	UDP		ETP	
	Flow 1	Flow 2	Flow 1	Flow 2
IPG Time	0,99	0,99	1,205	0,99
Transmission Time	12,55	10,36	10,89	10,36
RTT	23		21	

Main problem of UDP protocol is that UDP protocol doesn't have any algorithm to control the congestion; therefore transmission rate is always the same. It implies that IPG and required bandwidth are close to the value programmed by the user (IPG close to 1 ms). Maintain the same transmission rate can produce congestion since transmission time can increase due to the rest of network flows of data (Transmission time increase to 12,55). Nevertheless, ETP protocol implements a congestion control so the transmission rate changes according



to the protocol algorithm. As it can see, transmission rate will decrease when congestion is detected so the bandwidth decreases and the IPG increase (1,205) by using the EPT.

## 5 Conclusions

A new internet transport protocol called ETP has been introduced and checked for haptic applications. Main goal of ETP is to reduce the roundtime trip (RTT) by adapting the interpacket gap (IPG) to the congestion network conditions. IPG can vary into a range defined by the user, bandwidth fluctuations are detected by the RTT changes and IPG is modified accordingly. This protocol represents a step forward for haptic networked applications since haptic device parameters can be adjusted to the communication conditions, and transparency strategies can be designed according to communication time delays. Simulations of ETP behaviour have demonstrated the RTT reduction versus UDP protocol when appear congestions due to flows of data.

**Acknowledgments.** This work has been partially funded by the Spanish Ministry (MEC) and the European Commission FEDER funds under Grants DPI2005-08203-C02-01, DPI2004-01920, TSI2004-05165-C02-01, TIN2006-15516-C04-02, "Consolider Ingenio-2010" CSD2006-00046', by the Fundaci Caixa Castell-Bancaixa, by the Fundaci Caixa Castell under Grants P1-1B2003-15, P1-1A2003-10, by the European Commission funds "IMMERSENCE" FP6-IST-027141 and by the EU-VI Framework Programme under grant IST-045269 - "GUARDIANS" of the EC Cognitive Systems initiative.

## References

1. Hirche, S., Buss, M.: Human Perceived Transparency with Time Delay. In: Ferre, M., et al. (eds.) *Advances in Telerobotics*. STAR, vol. 31, pp. 191–209 (2007)
2. Postel, J.: RFC 793: Transmission Control Protocol, DARPA Internet Program Protocol Specification (1981)
3. Postel, J.: RFC 768: User Datagram Protocol (1980)
4. Uchimura, Y., Yakoh, T.: Bilateral robot system on the real-time network structure. *IEEE Transactions on Industrial Electronics* 51 (2004)
5. Ping, L., Wenjuan, L., Zengqi, S.: Transport layer protocol reconfiguration for network-based robot control system. *IEEE Networking, Sensing and Control* 2005 (2005)
6. Wirz, R., et al.: End-to-End Congestion Control Protocols for Remote Programming of Robots using Heterogeneous Networks: A Comparative Analysis. *Robotics and Autonomous Systems* 56(2)
7. Jacobson, V.: Congestion avoidance and control. In: *Symposium proceedings on Communications architectures and protocols*, Stanford, California, United States, August 1988, pp. 314–329 (1988)
8. Raju, G.J., Verghese, G.C., Sheridan, T.B.: Design Issues in 2-Port Network Models of Bilateral Remote Teleoperation. In: *Proceedings of the IEEE International Conference on Robotics and Automation*, USA, pp. 1317–1321 (1989)

9. Lawrence, D.A.: Stability and Transparency in Bilateral Teleoperation. *IEEE Transactions on Robotics and Automation* 9(5), 624–637 (1993)
10. Niemeyer, G., Slotine, J.E.: Towards Force-Reflecting Teleoperation Over the Internet. In: *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 1909–1915 (1998)
11. Yokokohji, Y., Imaida, T., Yoshikawa, T.: Bilateral Control with Energy Balance Monitoring under Time-Varying Communication Delay. In: *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 2684–2689 (2000)
12. Arcara, P., Melchiorri, C.: Control schemes for teleoperation with time delay: a comparative study. *Robotics and Autonomous Systems* 38(1) (2002)
13. Azorin, J.M., et al.: Generalized control method by state convergence for teleoperation systems with time delay. *Automatica* 40, 1575–1582 (2004)
14. Oyarzabal, M., Ferre, M., Cobos, S., Monroy, M., Barrio, J., Ortego, J.: Multi-finger Haptic Interface for Collaborative Tasks in Virtual Environments. In: Jacko, J.A. (ed.) *HCI 2007. LNCS*, vol. 4551, pp. 673–680. Springer, Heidelberg (2007)