# Transport Protocols for the Internet Interactive Applications

**Jae-young Lee**

(jla155@sfu.ca)

**ENSC 835 Spring 2009**

**Project Report**

**School of Engineering Science**

**Simon Fraser University**

August 13, 2009

## Abstract

TCP and UDP protocols are most widely used transport protocols in the Internet. Even though most Internet applications rely on TCP or UDP transmission, they are not designed for the interactive applications that require real-time operations. Several approaches proposed new transport protocols for the interactive applications. In this project, the existing TCP and UDP protocols are investigated in terms of the interactive applications. The new protocols named real-time protocol (RTP), real-time network protocol (RTNP), interactive real-time protocol (IRTP), and efficient transport protocol (ETP) are also investigated. Simulations including TCP, UDP and ETP are performed by using OPNET Modeler. As a performance measure, the end-to-end delay in teleoperation scenario is used.

# Table of Contents

1. Introduction ..... 4
2. Protocols for Interactive Applications ..... 5
    2.1 Existing Protocols ..... 5
        2.1.1 TCP ..... 5
        2.1.2 UDP ..... 5
    2.2 Interactive Application Protocols ..... 5
        2.2.1 RTP (Real-time Protocol) ..... 5
        2.2.2 RTNP (Real-time Network Protocol) ..... 6
        2.2.3 IRTP (Interactive Real-time Protocol) ..... 6
        2.2.4 ETP (Efficient Transport Protocol) ..... 6
3. OPNET Simulations ..... 7
    3.1 OPNET Implementation ..... 7
        3.1.1 Simulation Scenario ..... 7
        3.1.2 Existing Protocols Implementation (TCP and UDP) ..... 9
        3.1.3 ETP (Efficient transport protocol) Implementation ..... 10
    3.2 Simulation Results ..... 11
        3.2.1 Existing Protocols (TCP and UDP) ..... 11
        3.2.2 ETP (Efficient transport protocol) ..... 12
4. Discussion ..... 15
5. Conclusion ..... 16
6. References ..... 17

## 1. Introduction

The interactive applications are the Internet-based applications that send and receive real-time information between end systems. Recently, these interactive applications such as teleoperation, internet game, internet telephony, audio, and video services, have been widely deployed over the Internet. For example, in the teleoperation, which consists of a human operator, a teleoperator, and the Internet between them, the human operator sends motion data by user's manipulation and receives reflecting force data from the teleoperator in remote environment.

Unlike other elastic applications including e-mail, web, remote terminal access, and file transfer, the interactive applications are highly delay-sensitive. The time delay over the Internet is unknown and variable according to the network conditions. Thus, for the interactive applications, it can be observed that the real-time information is impaired due to the variable delay, which may lead undesirable real-time operation [1].

In order to solve the problem, there have been a lot of efforts by various approaches. The control system approaches including wave-variable transform have been suggested focusing on the stable operation between end systems [2], [3]. The signal processing approaches based on the prediction and estimation schemes to obtain unimpaired real-time information have been proposed recently [4], [5]. It can be found out that most approaches are based on the control system and signal processing approaches.

On the other hand, only few transport protocol approaches have been proposed. The transport protocol approaches are based on the modifications to the existing TCP and UDP protocols. The TCP and UDP are most widely used transport protocols in the Internet applications. Since they are not initially designed for the interactive applications, the interactive application protocols, called real-time protocol (RTP) [6], real-time network protocol (RTNP) [7], interactive real-time protocol (IRTP) [8], and efficient transport protocol (ETP) [9], have been introduced.

This project investigates the existing TCP and UDP protocols in terms of the interactive applications including the teleoperation. The interactive application protocols, which have been proposed recently, are also investigated. Simulations are performed by using OPNET Modeler. The simulations include the existing TCP and UDP protocols in the teleoperation scenario. The ETP, which is based on the inter-packet gap (IPG) insertion, is also implemented and compared with the existing TCP and UDP protocols. After discussing about the simulation results, the conclusion is provided.

# 2. Protocols for Interactive Applications

## 2.1 Existing Protocols

### 2.1.1 TCP

TCP is a transport layer protocol that provides reliable and connection-oriented service to the Internet applications such as e-mail, web application, remote terminal access, and file transfer. This protocol guarantees that packets are received at destination by performing retransmission mechanism. And it also performs congestion control in the Internet by adjusting the transmission. In the teleoperation, one of the interactive applications in the Internet, TCP may be useful for initial connection establishment between human operator and teleoperator and for delivery of crucial data at the nodes. However, the retransmission mechanism and congestion control in TCP lead relatively large variation of time delay and delay jitter, which may not be appropriate for real-time data transfer.

### 2.1.2 UDP

UDP, on the other hand, provides unreliable and connetionless service to the Internet applications such as streaming multimedia and voice over IP. This protocol basically sends datagram from a sender to receiver as fast as possible. Unlike TCP, it does not guarantee packets to be received at destination, and it does not perform any congestion control of the network. Thus, the data transfer in UDP environment can be accomplished without significant time delay and variations. In teleoperation, UDP may be useful for transfer of time sensitive haptic data even though it does not guarantee reliable data transmission and may lead to data losses.

## 2.2 Interactive Application Protocols

This section presents four transport protocols, which are especially designed for the Internet interactive applications. The interactive application protocols are based on modifications to existing TCP and UDP protocols for faster transmission. In section, the real-time protocol (RTP), real-time network protocol (RTNP), interactive real-time protocol (IRTP), and efficient transport protocol (ETP) are discussed.

### 2.2.1 RTP (Real-time Protocol)

The RTP has been introduced and standardized especially for real-time multimedia services. Since it has been designed for video and audio services by using an intermediate buffer to compensate the variation of time delay, it is not recommended for the transport protocols of other interactive applications. For example, in case of the teleoperation, the buffer contains larger total time delay that may adversely affect the smooth manipulation of the robot.

### 2.2.2 RTNP (Real-time Network Protocol)

The RTNP, which is designed for the teleoperation, has been proposed since the end-to-end delay between two end systems depends not only on the network, but also on software provided by operating system. Thus, this protocol is implemented based on the UNIX environment for faster transmission of haptic data and this is why the use of this protocol is limited.

### 2.2.3 IRTP (Interactive Real-time Protocol)

This protocol reconfigures and simplifies the packet header for the efficient and fast transmission of haptic data in the teleoperation. The IRTP takes advantage of TCP for the connection establishment and the transmission of crucial data. And it also uses UDP for the real-time data transmission which needs to be performed as fast as possible.

### 2.2.4 ETP (Efficient Transport Protocol)

The ETP has been introduced for the teleoperation over the Internet. This protocol mainly deals with the IPG, which is the time delay between two successive data packets. Figure 1 shows the data packet flow with IPG insertion. The IPG can be adjusted according to the network conditions. This IPG adjustment also implies the network congestion control that is only performed by TCP. That is, in case that the network is congested, the IPG can control the data rate depending on the available bandwidth, instead of window size scheme in TCP. Therefore, the combination of the IPG and UDP is recommended due to the fact that UDP does not provide congestion control.

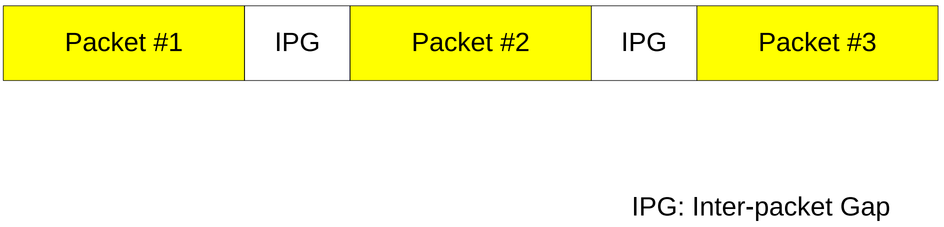


**Figure 1. Inter-packet gaps between data packets**

# 3. OPNET Simulations

## 3.1 OPNET Implementation

### 3.1.1 Simulation Scenario

In order to verify the performances of interactive application protocols, simulation scenarios were implemented by using the OPNET Modeler. Figure 2 shows a teleoperation scenario consists of two subnets located in West and East. In each subnet, the LAN models based on the star topology and servers were configured. The human operator and teleoperator were implemented in the West and East subnet respectively. To provide realistic Internet environment, the background traffic load was implemented between two subnets as shown in Figure 3. The IP cloud was also implemented with 1 percent of packet discard ratio and variable packet latency from 1ms to 100ms. Table 1 shows the simulation parameters used in this OPNET simulation scenario.

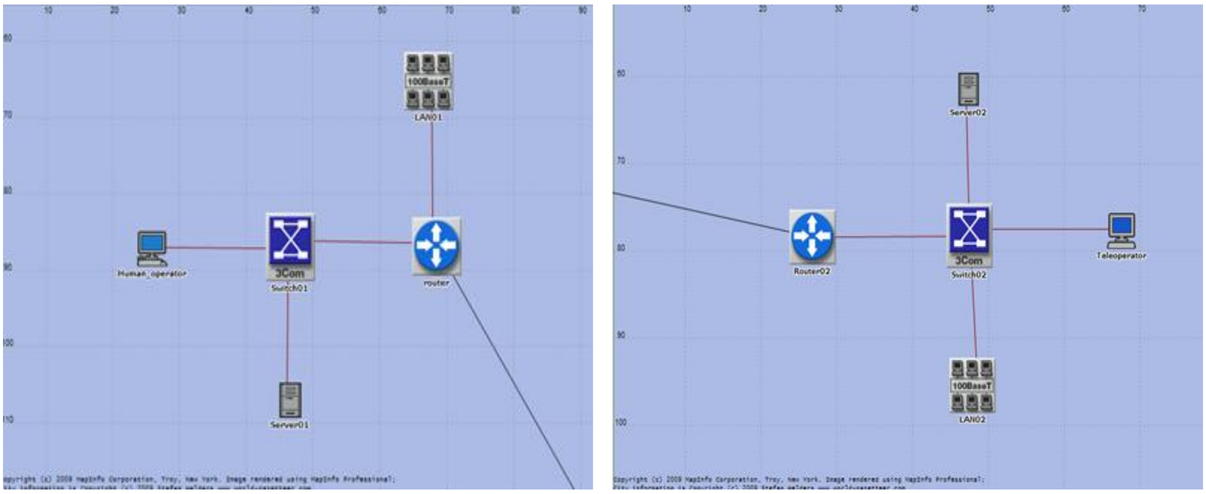


Figure 2. OPNET project view: WAN topology (top), West and East subnets (bottom)

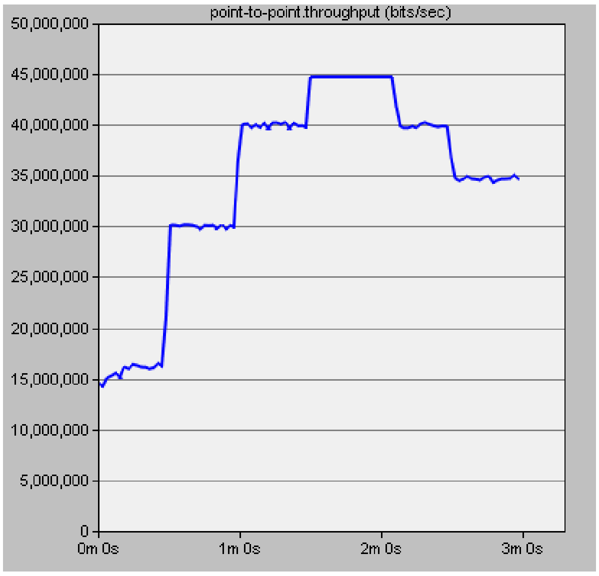


Figure 3. Background traffic load between two subnets

| Models | Details |
|---|---|
| Workstation | Usage: Human operator and teleoperator<br>Supported profile: CBR 1Mbps |
| LAN | Number of clients: 25<br>Supported profile: Voice over IP (PCM quality) |
| IP cloud | Packet discard ratio: 1%<br>Packet latency: 1ms - 100ms |
| Switch | Model: Nortel BL_BLN_4s_e4_f_sk8_tr4_base |
| Router | Model: Nortel BL_BLN_4s_e4_f_sk8_tr4_base |
| Link | WAN link: DS3 (45 Mbps)<br>LAN link: 10BaseT (10 Mbps) |
| Server | Default Ethernet server |

**Table 1. OPNET simulation models and details**

### 3.1.2 Existing Protocols Implementation (TCP and UDP)

Based on the OPNET simulation scenario discussed in section 3.1.1, the existing TCP and UDP protocols were implemented in order to evaluate the end-to-end delay performances. The TCP and UDP protocols can be selected by using the "Application Configuration" in the Figure 2. In this simulation, the TCP Reno version and standard UDP were simulated. To implement the data flow between the human operator and teleoperator, the "Task Configuration" was used. For each data flow, 1 Mbps constant bit rate (CBR) was implemented. Table 2 shows the parameters used in the "Task Configuration". The Table 2 refers that 2 packets with 500 bytes are sent in 8 milliseconds.

| Task attribute | Parameters |
|---|---|
| Packet size | 500 bytes |
| Inter-request time | 8 ms |
| Packets per request | 2 |

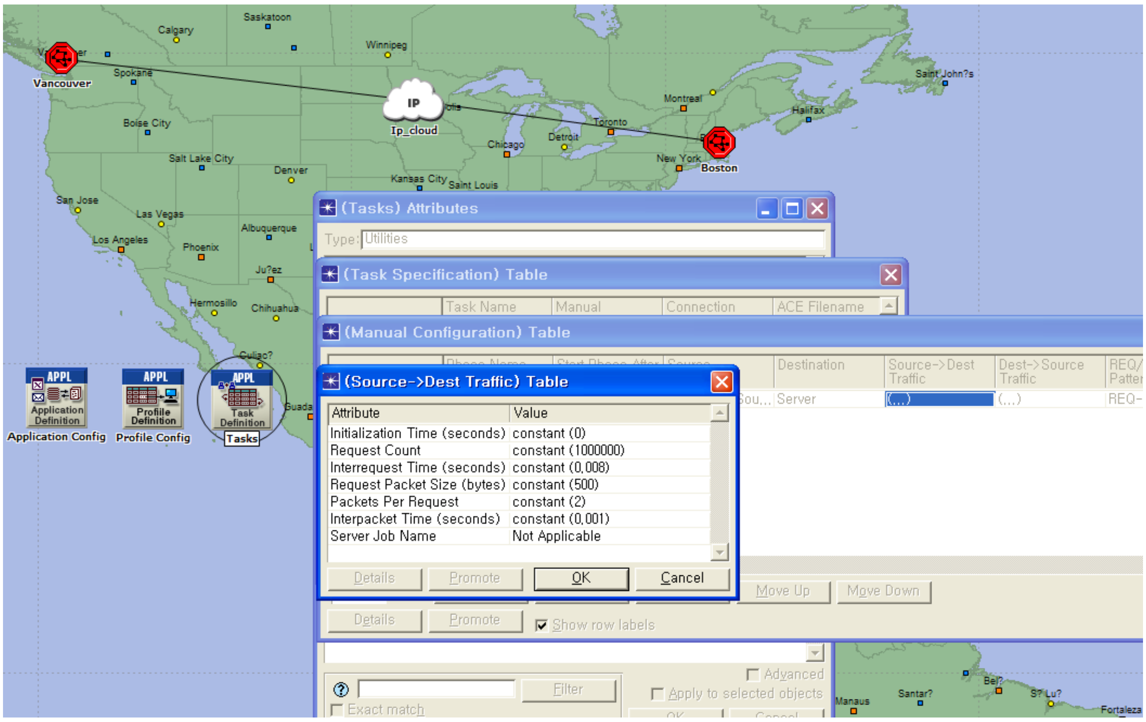
**Table 2. Task configuration parameters for TCP and UDP**

### 3.1.3 ETP (Efficient transport protocol) Implementation

The ETP is based on the IPG insertion between successive packets. In this project, the fixed values of IPGs are inserted in both TCP and UDP cases. The IPG can be also implemented by using the “Task Configuration”. Table 3 shows the parameters used in the “Task Configuration” when 1ms IPG is inserted. It refers that 2 packets with 500 bytes are sent in 9 milliseconds. That is, by introducing 1ms IPG, the data rate is reduced to 0.9 Mbps. The implementation of IPG by using the “Task Configuration” is shown in Figure 4.

| Task attribute | TCP |
|---|---|
| Packet size | 500 bytes |
| Inter-request time | 8 ms |
| Packets per request | 2 |
| IPG | 1 ms |

**Table 3. Task configuration parameters for ETP with 1ms IPG**



(Source->Dest Traffic) Table

| Attribute | Value |
|---|---|
| Initialization Time (seconds) | constant (0) |
| Request Count | constant (1000000) |
| Interrequest Time (seconds) | constant (0.008) |
| Request Packet Size (bytes) | constant (500) |
| Packets Per Request | constant (2) |
| Interpacket Time (seconds) | constant (0.001) |
| Server Job Name | Not Applicable |

**Figure 4. Implementation of IPG using “Task Configuration”**

## 3.2 Simulation Results

### 3.2.1 Existing Protocols (TCP and UDP)

As a performance measure, the end-to-end delay between the human operator and teleoperator was collected for 2 minutes. After obtaining the end-to-end delay, the average, minimum, maximum, and standard deviation of time delay were measured. The end-to-end delay in both TCP and UDP cases are shown in Figure 5 and the measurements are summarized in Table 4. According to the Figure 5 and Table 4, the variation of the end-to-end delay in UDP is smaller than that in TCP.

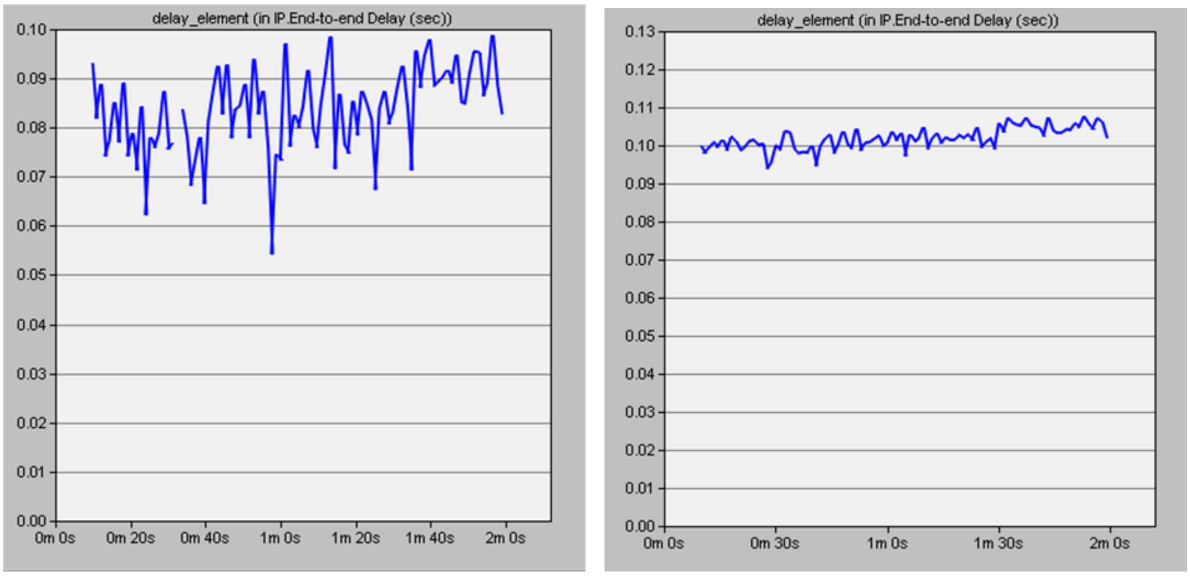


**Figure 5. End-to-end delay in TCP (left) and UDP (right)**

| Delay | TCP | UDP |
|---|---|---|
| Average | 83.3 | 101.9 |
| Minimum | 54.4 | 93.9 |
| Maximum | 98.9 | 107.6 |
| Standard deviation | 8.4 | 2.8 |

**Table 4. End-to-end delay measurements in TCP and UDP**

### 3.2.2 ETP (Efficient transport protocol)

In order to evaluate the performance of the ETP, the IPG was inserted in both TCP and UDP cases. In case of the TCP, 4ms IPG was inserted. The end-to-end delay was obtained and compared with the standard TCP case as shown in Figure 6. Table 5 also describes the comparison between the IPG inserted TCP and standard TCP. According to the results, the IPG insertion does not improve the end-to-end delay performance in the TCP case.

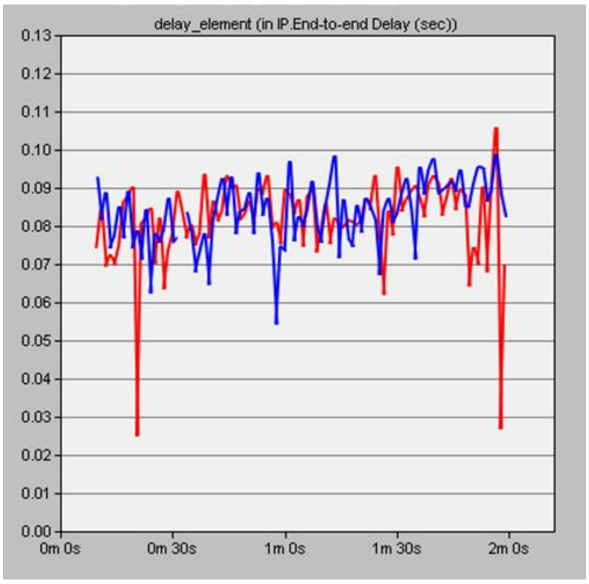


**Figure 6. End-to-end delay in standard TCP (blue) and IPG inserted TCP (Red)**

| Delay | TCP | IPG in TCP (4ms) |
|---|---|---|
| Average | 83.3 | 81.4 |
| Minimum | 54.4 | 25.3 |
| Maximum | 98.9 | 105.9 |
| Standard deviation | 8.4 | 11.3 |

**Table 5. End-to-end delay measurements in standard TCP and IPG inserted TCP**

For the next simulation, the IPG insertion scheme was evaluated in the UDP case. The end-to-end delay performance was measured when 1ms IPG was inserted in UDP, and it is compared with the standard UDP as shown in Figure 7 and Table 6. According to the results, the average time delay was reduced by 6.2ms when only 1ms IPG was inserted. Thus, the IPG insertion improves the end-to-end delay performance in the UDP case.

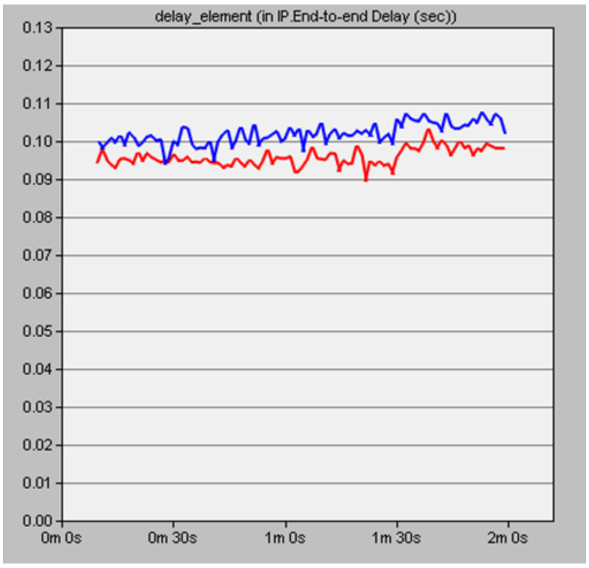


**Figure 7. End-to-end delay in standard UDP (blue) and IPG inserted UDP (Red)**

| Delay | UDP | IPG in UDP (1ms) |
|---|---|---|
| Average | 101.9 | 95.7 |
| Minimum | 93.9 | 89.5 |
| Maximum | 107.6 | 103.2 |
| Standard deviation | 2.8 | 2.3 |

**Table 6. End-to-end delay measurements in standard UDP and IPG inserted UDP**

In the UDP case, it is expected that the end-to-end delay can be reduced as the IPG is increased. To simulate this, the IPG was increased from 1ms to 8ms, and they were compared each other with respect to the end-to-end delay performance as shown in Figure 8 and Table 7. According to the results, the average time delay was reduced as the IPG is increased. However, the standard deviation, which refers variation of time delay, tends to increase as the IPG is increased.

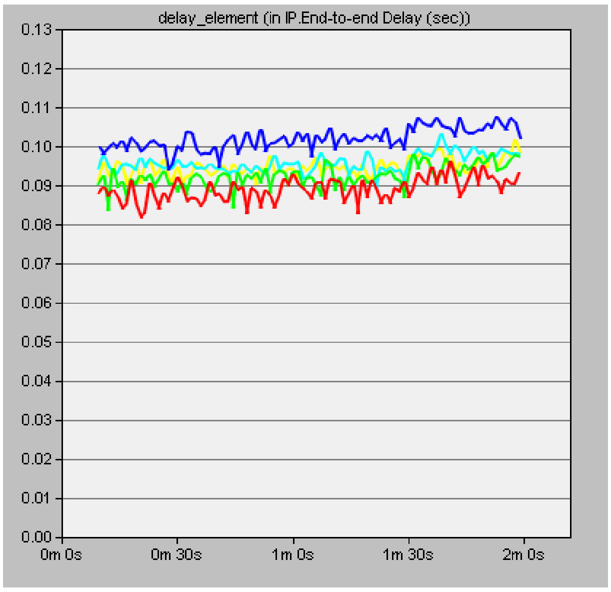


**Figure 8. End-to-end delay in standard UDP (blue), 1m IPG (pink), 2ms IPG (yellow), 4ms IPG (green), and 8ms IPG (red) in UDP**

| **Delay** | **UDP** | **IPG (1ms)** | **IPG (2ms)** | **IPG (4ms)** | **IPG (8ms)** |
|---|---|---|---|---|---|
| Average | 101.9 | 95.7 | 94.6 | 92.3 | 88.9 |
| Minimum | 93.9 | 89.5 | 89.9 | 83.6 | 81.9 |
| Maximum | 107.6 | 103.2 | 101.7 | 98.4 | 96.2 |
| Standard deviation | 2.8 | 2.3 | 2.3 | 2.9 | 3.0 |

**Table 6. End-to-end delay measurements in standard UDP and IPG inserted UDP (1~8ms)**

## 4. Discussion

According to the simulation results in section 3, the end-to-end delay is reduced as the IPG is increased in the UDP case. However, it would an open question how much the IPG needs to be increased in order to achieve the optimal end-to-end delay performance. Figure 9 shows the end-to-end delay performance when 32ms IPG is inserted in the UDP case. By comparing it with 1ms IPG case, the variation of time delay is dramatically increased, which is not desired in the teleoperation even though the average time delay is reduced. Thus, it is advised that the optimal value of the IPG needs to be selected by considering both the average delay and the variation of delay.

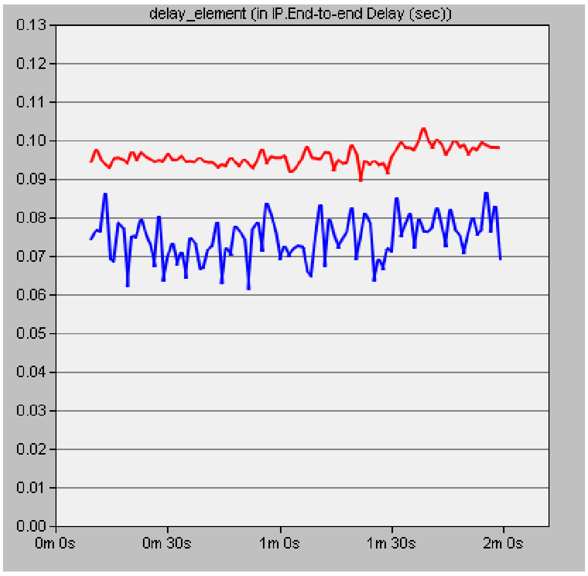


**Figure 9. End-to-end delay in UDP with 1ms IPG (red) and 32 ms IPG (blue)**

| Delay | 1ms IPG in UDP | 32ms IPG in UDP |
|---|---|---|
| Average | 95.7 | 74.2 |
| Minimum | 89.5 | 61.6 |
| Maximum | 103.2 | 86.8 |
| Standard deviation | 2.3 | 5.6 |

**Table 6. End-to-end delay measurements in standard UDP and IPG inserted UDP**

Furthermore, in case that the IPG insertion scheme is considered for the teleoperation scenario, the discontinuity problem of haptic data must be concerned. In the teleoperation, the human operator sends motion data no less than 30Hz sampling rate, and receives force data no less than 1000Hz in order to prevent the discontinuity of the haptic data. Since the nature of IPG insertion gives the discontinuity between successive packets, it is obvious that the value of the IPG must be selected to prevent the discontinuity of the haptic data.

Due to the fact that the sampling rate of the force data is relatively high, the use of the IPG insertion scheme can be limited. In order to use a large value of the IPG, the compression scheme for the force data can be employed. Using the compression scheme, the sampling rate of the force data can be lower than 1000Hz. Therefore, the compression scheme will allow increasing the IPG while preventing the discontinuity of the haptic data.

## 5. Conclusion

In this project, the existing TCP and UDP protocols were investigated in terms of the Internet interactive applications, especially for the teleoperation. The four interactive application protocols, which have been recently proposed, were discussed as well. Among them, the ETP based on the IPG insertion scheme was implemented and compared with existing TCP and UDP protocols by using OPNET Modeler. According to the OPNET simulation, the IPG in the TCP case did not improve the end-to-end delay performance. On the other hand, the IPG in the UDP case improves the end-to-end delay performance as increasing the IPG. It is advised that the optimal value of IPG needs to be selected to avoid the large variation of time delay and to prevent the discontinuity problem of haptic data. For the future remark, the compression scheme for the haptic data is expected in order to increase the IPG.

## 6. References

[1] E. Kamrani, H. Momeni, and A. Sharafat, “Modeling Internet delay dynamics for teleoperation,” Proc. IEEE International Conference on Control Applications, Aug. 2005.

[2] G. Niemeyer and J. Slotine, “Designing force reflecting teleoperators with large time delays to appear as virtual tools,” Proc. IEEE International Conference on Robotics and Automation, Albuquerque, NM, Apr. 1997, pp. 2212–2218.

[3] K. Kawashima, K. Tadano, G, Sankaranarayana, and B. Hannaford, “Bilateral teleoperation with time delay using modified wave variables,” Proc. IEEE International Conference on Intelligent Robots and Systems, Sept. 2008.

[4] S. Clarke, G. Schillhuber, M. Zach, and H. Ulbrich, “The effects of simulated inertia and force prediction on delayed telepresence,” Presence, vol. 16, no. 5, pp. 543-558, Oct. 2007.

[5] S. Munir and W. Book, “Internet-based teleoperation using wave variables with prediction,” IEEE Transactions on Mechatronics, vol. 7, no 2, June 2002.

[6] H. Schulzrinne, S. Deering, R. Frederick, and V. Jacobson, “RTP: a transport protocol for real-time applications,” RFC 3550, July 2003.

[7] L. Ping, "Transport layer protocol reconfiguration for network-based robot control system," Proc. IEEE International Conference on Networking, Sensing and Control, Tucson, AZ, Mar. 2005, pp. 1049-1053.

[8] Y. Uchimura, "Bilateral robot system on the real-time network structure," IEEE Transactions on Industrial Electronics, vol. 51, no. 5, pp. 940-946, Oct. 2004.

[9] R. Wirz, "Efficient transport protocol for networked haptics applications," Proc. The 6th International Conference on Haptics: Perception, Devices and Scenarios, Madrid, Spain, June 2008, vol. 5024, pp. 3-12.