Performance Evaluation of Transport Protocols for Internet-Based Teleoperation Systems

Jae-young Lee, Shahram Payandeh, and Ljiljana Trajković
Simon Fraser University
Vancouver, British Columbia
Canada
E-mail: {jla155, shahram, ljilja}@sfu.ca

Abstract

An Internet-based teleoperation system is an interactive application where a human user transmits movement data of a robotic device while simultaneously receiving reflected force data from a remote teleoperator. Performance of such real-time applications is highly sensitive to the Internet delay and data loss. In this paper, we describe the efficient transport protocol (ETP) designed for Internet-based teleoperation systems and demonstrate that it reduces the round trip time (RTT) between a human user and a remote teleoperator. We use OPNET Modeler to simulate the ETP protocol and compare its performance with the transport control protocol (TCP) and the user datagram protocol (UDP) by measuring end-to-end delays in various simulation scenarios. We observe that with UDP as a transport protocol, ETP reduces the end-to-end delay by introducing an inter-packet gap (IPG).

1. Introduction

In a teleoperation scenario, a human user manipulates tools stationed in a remote environment through an interactive communication medium. A typical teleoperation system consists of a human operator, a remote teleoperator, and a communication medium between the two. It enables a wide variety of applications such as military tasks, space robotics, underwater operations, and long distance medical diagnostics and surgeries. Recent research activities have focused on Internet-based teleoperation systems because of their availability, ease of access, and low cost [1].

An Internet-based teleoperation system is an interactive application where the human operator sends motion or velocity data according to a user's manipulation and receives reflecting force data from the teleoperator through the Internet. Unlike other applications such as e-mail, web, remote terminal access, and file transfer, interactive applications are highly sensitive to the Internet delay and data loss. The Internet delay varies according to network conditions and, hence, affects the motion and force data in Internet-based teleoperation systems. This subsequently leads to undesirable performance and may cause instability of the overall teleoperation system [2].

Many approaches proposed to solve the time delay issue in Internet-based teleoperation systems are based on control systems and signal processing approaches. In the area of control systems, the proposed wave variables transformation and its extensions have focused on the stability between two end-systems in the case of the constant time delay [3], [4]. In the area of signal processing, prediction and estimation algorithms have been introduced in order to deal with the real-time information impaired by Internet delays [5]–[7]. In the area of Internet protocols, the transport control protocol (TCP) and user datagram protocol (UDP) are widely used for most Internet applications. The real-time transport protocol (RTP) has been developed by the audio and video transport working group to support the recent deployment of real-time multimedia streaming applications [8]. However, because these protocols were not initially designed for Internet-based teleoperation systems, new transport protocols have been developed based on the modifications to the existing TCP and UDP. The real-time network protocol (RTNP) [9], interactive real-time protocol (IRTP) [10], and efficient transport protocol (ETP) [11], [12] are examples of proposed transport protocols for Internet-based teleoperation systems.

In this paper, we review transport protocols for Internet-based teleoperation systems. We also investigate the suitability of the existing TCP and UDP for this application by using OPNET Modeler. We use OPNET Modeler to simulate ETP combined with the inter-packet gap (IPG) algorithm in an Internet-based teleoperation scenario. The comparison with the existing TCP and UDP are also shown.

2. Protocols for Internet-Based Teleoperation Systems

TCP: TCP is a transport layer network protocol that provides reliable and connection-oriented service to a variety of Internet applications. This protocol employs packet retransmissions to guarantee that packets are received at destination. It also performs congestion control to regulate the transmission rate in the network. In teleoperation systems, TCP may be useful for establishing an initial connection between the human operator and teleoperator and for delivering crucial data. However, the TCP retransmission mechanism and the congestion control algorithm lead to relatively large variations of time delay and delay jitter, which may not be appropriate for real-time data transfers.

UDP: UDP provides unreliable and connectionless service to the Internet applications such as streaming multimedia and voice over Internet protocol (IP). UDP sends datagrams from a sender to receiver as fast as possible. Unlike TCP, it does not guarantee packets delivery and it does not perform congestion control. Hence, the data transfer using UDP can be accomplished without significant time delay and variations. In teleoperation systems, UDP may be useful for transfers of time sensitive haptic data even though it does not guarantee reliable data transmission and may lead to data loss.

RTP: RTP has been introduced to carry the real-time information between two end-systems in the recent deployment of interactive Internet applications such as video and audio streaming services, Internet telephony, and Internet games. This protocol employs an intermediate buffer in order to deal with the variation of time delay and it is suitable for real-time video and audio services [8]. In the case of teleoperation systems, the intermediate buffer introduces additional time delay, which eventually causes larger end-to-end delay between end-systems and even instability of the overall system. Therefore, deploying an intermediate buffer is generally not recommended for Internet-based teleoperation systems.

RTNP: RTNP has been introduced as a protocol for Internet-based teleoperation systems [9]. This protocol was designed for UNIX environments because the end-to-end time delay depends not only on the network condition, but also on the software supported by the operating system. Due to this limitation, this protocol may not be used on other platforms.

IRTP: IRTP has been also designed for Internet-based teleoperation systems [10]. This protocol reconfigures and simplifies packet headers of haptic data in order to reduce the end-to-end delay. IRTP employs TCP for the connection establishment and the transmission of crucial data. It then uses UDP to transmit the remaining haptic data, which need to be transmitted as fast as possible.

ETP: ETP, also designed for Internet-based teleoperation systems, focuses on reducing the end-to-end delay between a human operator and a teleoperator by introducing a time gap, called the IPG, between two successive data packets [11], [12]. A simple illustration is shown in Figure 1.

Packet #1	IPG	Packet #2	IPG	Packet #3

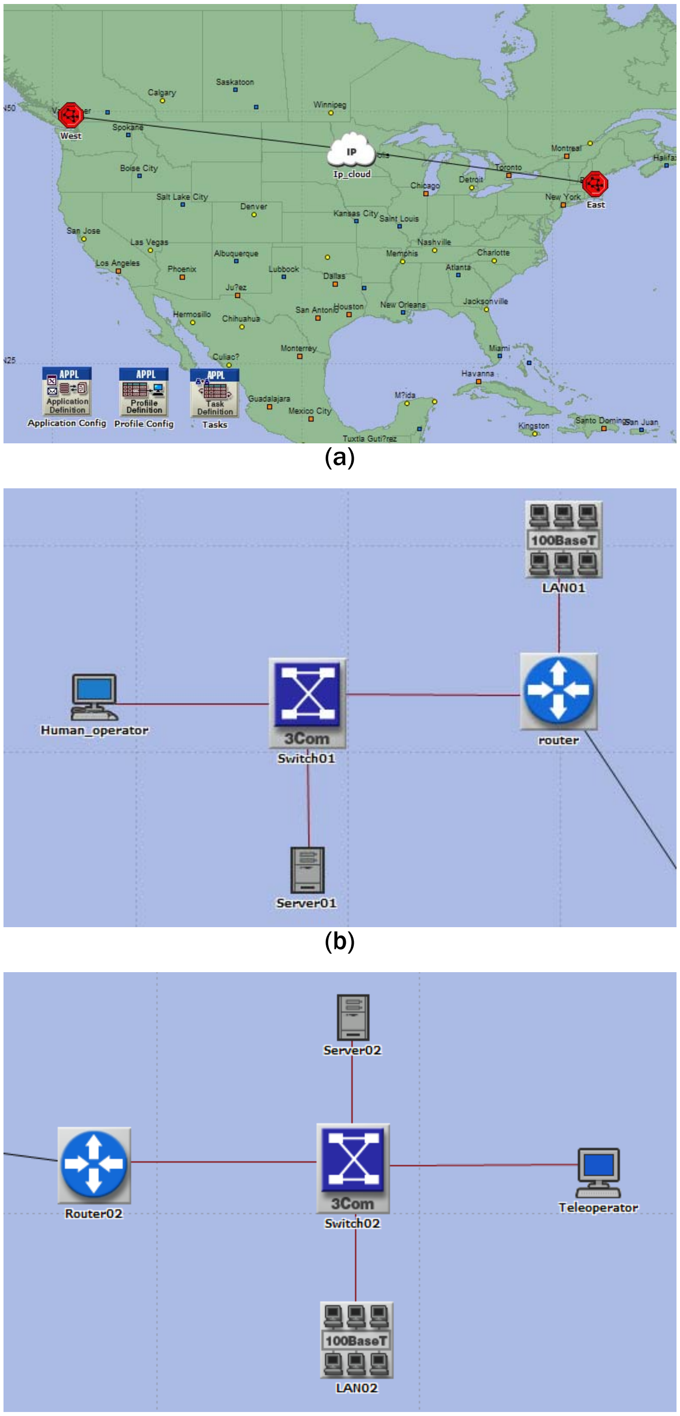
IPG: Interpacket Gap

Figure 1: Inter-packet gaps between data packets.

The IPG may be controlled depending on network conditions. This IPG control provides a congestion control in the network similar to TCP congestion control using the window size. For example, in the case when the network is congested, increasing the IPG, instead of using the TCP window size scheme, may control the data rate within available bandwidth. UDP alone does not provide any congestion control mechanism and, hence, the IPG control with UDP is recommended for ETP.

3. OPNET Simulation Model

In order to evaluate and compare the performance of protocols suitable for teleoperation systems, we implemented an Internet-based teleoperation system using OPNET Modeler. The OPNET model of a teleoperation system in a wide area network (WAN) is shown in Figure 2(a). A human operator and a teleoperator are located in the West and East subnets, respectively. The configurations of each subnet including servers and local area network (LAN) models based on a star topology are shown in Figure 2(b) and 2(c). The details of the OPNET simulation model are given in Table 1. In order to provide a realistic Internet environment, a background traffic load was introduced between two subnets, as shown in Figure 3. The IP cloud has a 1% packet discard ratio and variable packet latency between 1 msec and 100 msec.



(a)

(b)

(c)

Figure 2. The OPNET model of an Internet-based teleoperation system: (a) WAN topology; (b) Subnet including a human operator; (c) Subnet including a teleoperator.

Models	Details
Workstation	Usage: human operator and teleoperator Supported profile: CBR 1 Mbps
LAN	Number of clients: 25 Supported profile: Voice over IP (PCM quality)
IP cloud	Packet discard ratio: 1% Packet latency: 1 msec – 100 msec
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) LAN link: 10BaseT (10 Mbps)
Server	Default Ethernet server

Table 1. Parameters of the OPNET simulation model.

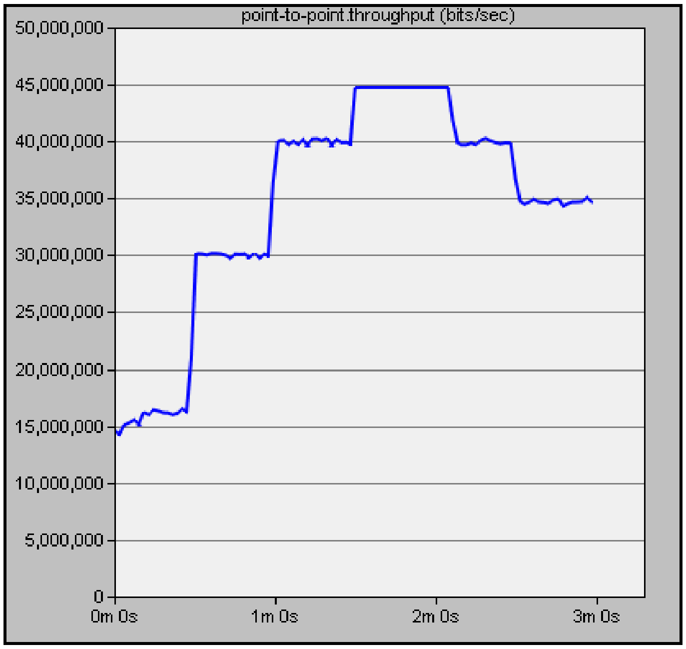


Figure 3. Background traffic load between a human operator and a teleoperator.

3.1 TCP and UDP Implementations

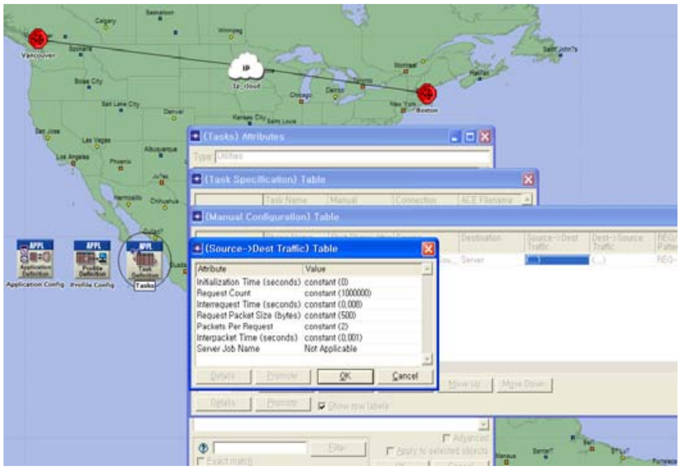
We evaluated performance of TCP and UDP based on the OPNET model of a teleoperation system. Both TCP and UDP were selected by the application configuration shown in Figure 2(a). In this simulation, we used TCP Reno and UDP. The task configuration shown in Figure 2(a) was used to generate data flows between the human operator and the teleoperator. 1 Mbps constant bit rate (CBR) traffic was used to simulate data flow from the human operator to the teleoperator. The same data rate was used for the feedback data flow from the teleoperator to the human operator. The details of the task configuration parameters for each data flow are shown in Table 2. By using the task configuration, the 1 Mbps data rate was generated using two packets of 500 bytes that are sent every 8 msec.

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

Table 2. Task configuration parameters for TCP and UDP.

3.2 ETP Implementation

The ETP implementation is based on the IPG between successive packets. In this study, we used the IPG values between 1 msec and 8 msec for both TCP Reno and UDP. The IPG may be simulated by manipulating the task configuration, as shown in Figure 4. Once a base protocol was selected using the application configuration, the IPG parameters were set in the task configuration. The task configuration parameters are shown in Table 3. In the case of 1 msec IPG, two packets with 500 bytes are sent every 9 msec. Hence, by introducing the 1 msec IPG, the data rate was reduced from 1 Mbps to 0.9 Mbps.



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. IPG simulation using task configuration.

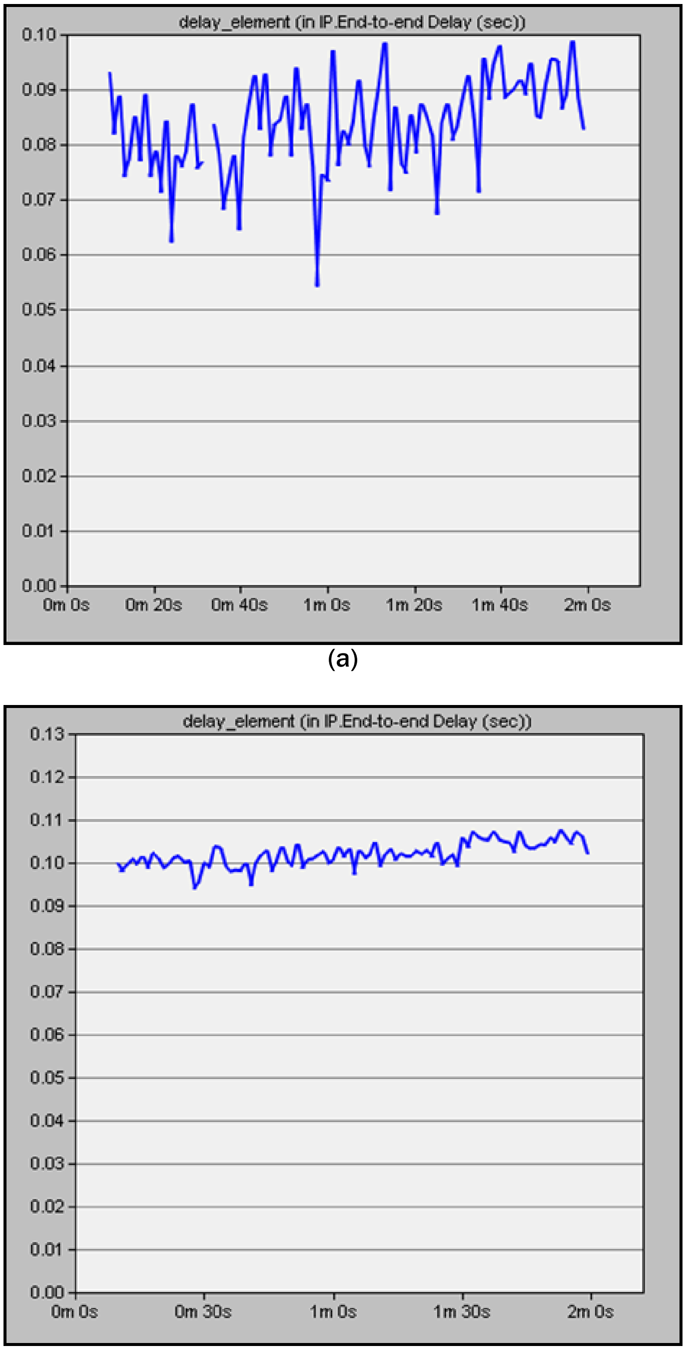
Task attribute	Parameters
Packet size	500 bytes
Inter-request time	8 msec
Packets per request	2
IPG	1 msec – 8 msec

Table 3. Task configuration parameters for the ETP implementation using the IPG.

4. OPNET Simulation Results

4.1 TCP and UDP simulation scenarios

As a performance measure, the end-to-end delay between the human operator and teleoperator was collected over a two-minute interval. The average, minimum, maximum, and standard deviation of the end-to-end delay were collected. The end-to-end delays and the simulation results with TCP Reno and UDP are shown in Figure 5 and Table 4. The standard deviation of the end-to-end delay with UDP is smaller than that with TCP Reno. Since the variation of the time delay in teleoperation systems may impair the haptic data and cause instability of the overall system, using UDP is reasonable even though it does not guarantee reliable data transmission.



(a)

(b)

Figure 5. End-to-end delay: (a) TCP Reno and (b) UDP.

Delay	TCP Reno	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: TCP Reno and UDP scenarios.

4.2 ETP Simulation Scenario

We first simulated the IPG algorithm with TCP Reno used as a transport protocol. Its performance was then compared with TCP Reno. In this simulation scenario, 4 msec IPG was used. The end-to-end delay over a two-minute interval was collected and compared with TCP Reno, as shown in Figure 6. The comparison between TCP Reno and TCP Reno with the IPG is shown in Table 5. With TCP Reno as a transport protocol, introducing the IPG did not improve the end-to-end delay performance, as shown in Figure 6 and Table 5.

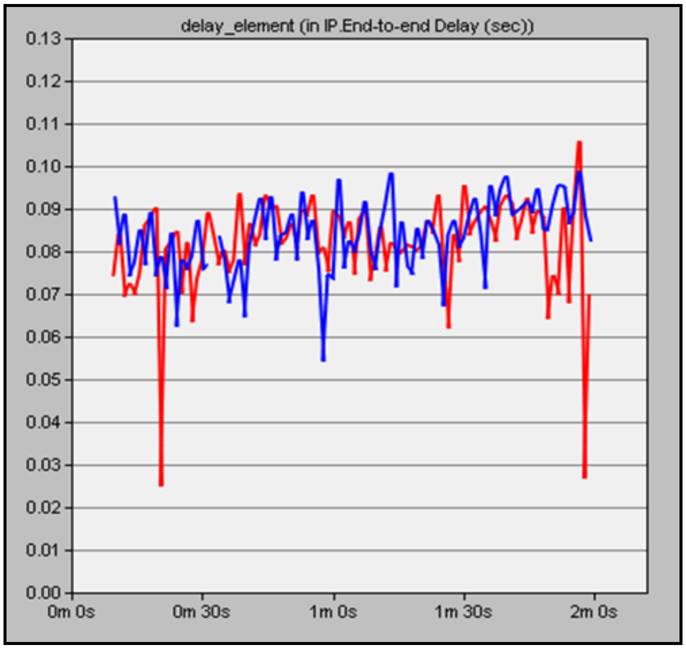


Figure 6. End-to-end delay: TCP Reno (🟦) and TCP Reno with 4 msec IPG (🟥).

Delay	TCP Reno	TCP Reno with 4 msec IPG
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: TCP Reno and TCP Reno with 4 msec IPG.

We also simulated the IPG when UDP was used as a transport protocol. In this simulation scenario, 1 msec IPG was used. The end-to-end delay over a two-minute interval was collected and compared with UDP, as shown in Figure 7. The values of the end-to-end delay are shown in Table 6. The average time delay was reduced by 6.2 msec when 1 msec IPG was used, illustrating that the end-to-end delay performance was improved by using the IPG.

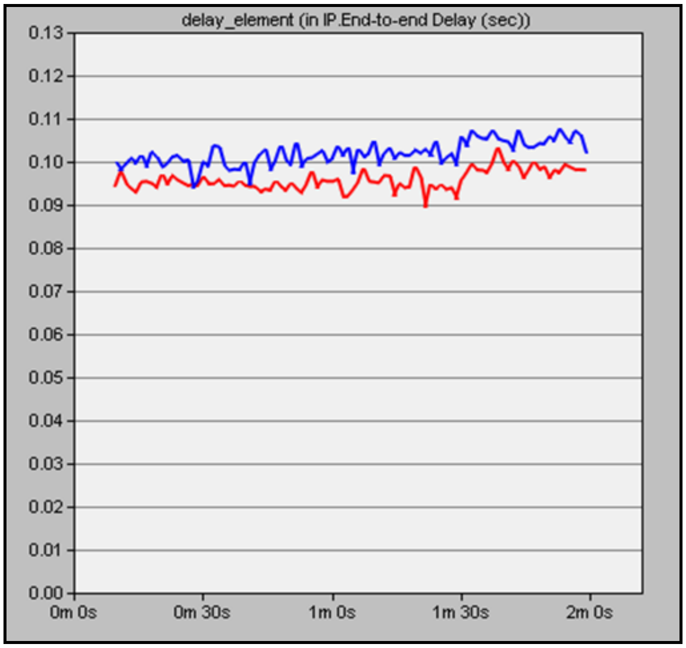


Figure 7. End-to-end delay: UDP (🟦) and UDP with 1 msec IPG (🟥).

Delay	UDP	IPG with UDP (1 msec)
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 delays: UDP and UDP with 1 msec IPG.

In the case of UDP, the end-to-end time delay decreased when 1 msec IPG was used. In this simulation scenario, the IPG increased from 1 msec to 8 msec. The end-to-end delay over a two-minute interval when the IPG varies from 1 msec to 8 msec is shown in Figure 8. The values of the end-to-end delay are shown in Table 7. The average time delay was reduced as the IPG increases when UDP was used as a transport protocol. However, with the IPG increase, the standard deviation also tended to increase, implying larger variations of the time delay.

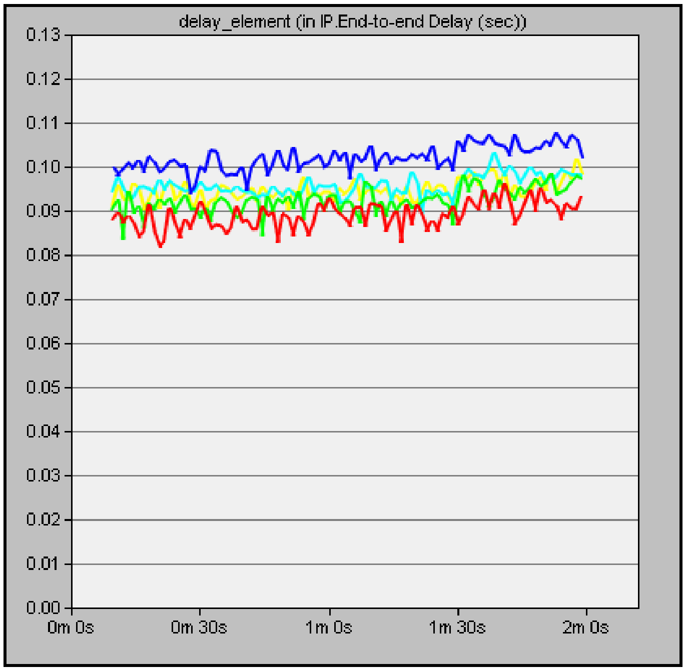


Figure 8. End-to-end delay: UDP (🟦), UDP with 1 msec IPG (■), 2 msec IPG (🟨), 4 msec IPG (🟩), and 8 msec IPG (🟥).

Delay	UDP	IPG (1 msec)	IPG (2 msec)	IPG (4 msec)	IPG (8 msec)
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 7. End-to-end delay: UDP and UDP with 1 msec to 8 msec IPGs.

Although the end-to-end delay performance improves proportionally with the IPG value, it is an open question how to achieve optimal performance. The end-to-end delay for 32 msec IPG with UDP is shown in Figure 9. The values of the end-to-end delay are shown in Table 8. The variation of time delay was dramatically increased in comparison with the 1 msec IPG case. This is undesirable in a teleoperation system even though the average time delay was reduced. Hence, an optimal value of the IPG should be selected by considering both the average time delay and its variations.

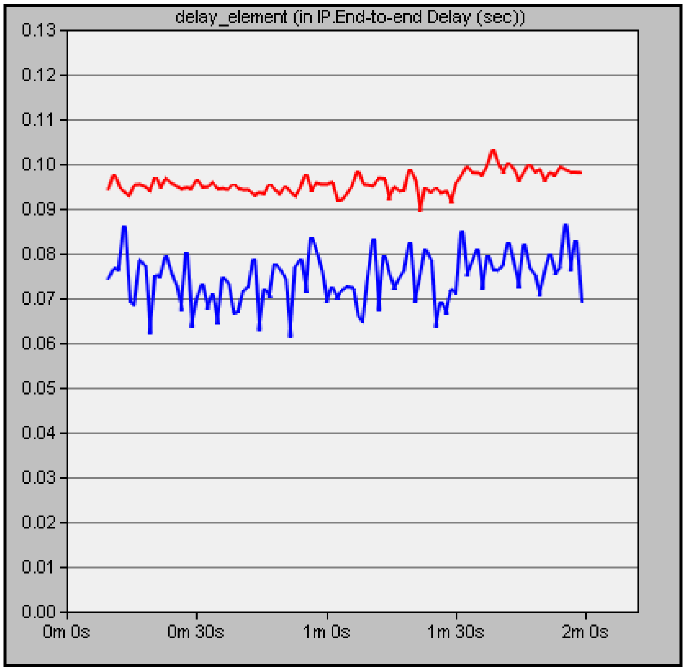


Figure 9. End-to-end delay: UDP with 1 msec IPG (🟥) and 32 msec IPG (🟦).

Delay	**IPG (1 msec)**	**IPG (32 msec)**
Average	95.7	74.2
Minimum	89.5	61.6
Maximum	103.2	86.8
Standard deviation	2.3	5.6

Table 8. End-to-end delay: UDP with 1 msec and 32 msec IPGs .

The discontinuity of haptic data should be considered when the IPG is used for a teleoperation system. In a typical teleoperation system, the human operator sends motion data at no less than 30 Hz sampling rate and receives force data at no less than 1,000 Hz in order to prevent the discontinuity of the haptic data and to maintain the closed-loop stability of the overall system. Since the IPG introduces gaps between successive packets, its value should be carefully selected to avoid the discontinuity of haptic data.

5. Conclusions

In this paper, TCP and UDP were evaluated within an Internet-based teleoperation system. The recently proposed interactive application protocols were also considered. The simulations were performed by using OPNET Modeler in order to evaluate ETP based on the IPG values. OPNET simulation results indicate that the end-to-end delay performance did not improve by using the IPG algorithm when TCP is used as a transport protocol. However, when UDP is used, the end-to-end delay performance improves as the value of the IPG increases. We conclude that an optimal value of IPG should be selected to avoid large variations in time delay and the discontinuity of haptic data.

References

[1] E. Kamrani, H. Momeni, and A. Sharafat, “Modeling Internet delay dynamics for teleoperation,” in *Proc. IEEE Int. Conf. on Control Applications*, Aug. 2005, pp. 1528–1533.

[2] G. Niemeyer and J. Slotine, “Stable adaptive teleoperation,” *IEEE Journal of Oceanic Engineering*, vol. 16, no. 1, pp. 152–162, Jan. 1991.

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

[4] K. Kawashima, K. Tadano, G. Sankaranarayana, and B. Hannaford, “Bilateral teleoperation with time delay using modified wave variables,” in *Proc. IEEE Int. Conf. on Intelligent Robots and Systems*, Sept. 2008, pp. 424–429.

[5] 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.

[6] S. Munir and W. Book, “Internet-based teleoperation using wave variables with prediction,” *IEEE/ASME Trans. on Mechatronics*, vol. 7, no 2, pp. 124–133, June 2002.

[7] J. Lee, S. Payandeh, and Lj. Trajković, “Application of prediction-based particle filters for teleoperations over the Internet,” in *Proc. The 14th IASTED Int. Conf. on Robotics and Applications, RA 2009*, Cambridge, MA, USA, Nov. 2009, pp. 22–27.

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

[9] L. Ping, L. Wenjuan, and S. Zengqi, “Transport layer protocol reconfiguration for network-based robot control system,” in *Proc. IEEE Int. Conf. on Networking, Sensing, and Control*, Tucson, AZ, Mar. 2005, pp. 1049–1053.

[10] Y. Uchimura and T. Yakoh, “Bilateral robot system on the real-time network structure,” *IEEE Trans. on Industrial Electronics*, vol. 51, no. 5, pp. 940–946, Oct. 2004.

[11] R. Rejaie, M. Handley, and D. Estrin, “RAP: An end-to-end rate-based congestion control mechanism for realtime streams in the Internet,” in *Proc. IEEE INFOCOM*, Mar. 1999, pp. 1337–1345.

[12] R. Wirz, M. Ferre, R. Marín, J. Barrio, J. Claver, and J. Ortego, “Efficient transport protocol for networked haptics applications,” in *Proc. The 6th International Conference on Haptics: Perception, Devices and Scenarios*, Madrid, Spain, June 2008, vol. 5024, pp. 3–12.