Evaluation of Real-Time Delays for Networked Telerobotics

Mayez A. Al-Mouhamed[1], Onur Toker[1], Asif Iqbal[2], and Syed M.S. Islam[1]

(1) College of Computer Science and Engineering (CCSE), King Fahd University of Petroleum and Minerals (KFUPM), Dhahran 31261, Saudi Arabia. mayez/onur/sislam@ccse.kfupm.edu.sa.

(2) IPP Multisensorics, Center for Sensor Systems, ZESS, University of Siegen, D-57068, Germany. iqbal@ipp.zess.uni-siegen.de.

Abstract— A real-time telerobotic system consisting of client station (operator) and server station (slave arm) interconnected by a computer network has been implemented using a distributed component framework. To minimize overall delays a multi-threaded execution is proposed for pipelining of information processing and real-time transmission. Thread engineering allowed pipelining stereo grabbing and live transfer of stereo video data. Different scenarios are statistically analyzed to relate the effect of thread manipulation to overall time delays. Analysis of telerobotic delays through three campus routes with different network loads is presented. A sampling rate of 120 Hz is achieved for force feedback and 50 Hz for operator commands when network load is below 80%. Copying stereo images from cameras to memory is done in 24 ms. Stereo video transfer operates at a rate of 17 fps. Total reference delays for force and stereo are 8 ms and 83 ms, respectively. The environment interaction delay is 183 ms (5.5 Hz) when slave arm is operated at 10 Hz. However, short instantaneous traffic irregularities may cause deviation and scattering from above reference rates.

Index Terms— Network delays, real-time control, relaying stereo video, streaming force feedback, telerobotics.

I. INTRODUCTION

THE performance of networked teleoperation systems is based on timely streaming of highly-demanding dynamic media to interface human operator to the actuators and sensors of a remote robot. A central problem is communication delays [1]: satellite links, for example, often have round-trip delays that last from a fraction of a second to several seconds. In telesurgery [2] delays can greatly slow down task execution, as the surgeon must pace the procedure to wait to see the effects of commanded motions. Delays in teleoperation with force feedback [3], [4] can cause instability of the robot control system, although various techniques can help to minimize this problem.

Real-time network and protocol transmission delays, jitter [5], and processing times need to be reduced to ensure guaranteed quality of service for robot commands, stereo vision, and force feedback. In a computer network, the communication delays and traffic capacity vary with flow direction and irregularly change with traffic conditions.

Teleoperation [6] on packet switched LAN indicates that when packet size is increased from 64 to 1024 bytes, the delay is increased from 5.6 ms to 13.4 ms due to computational overheads. LAN performs well even in the presence of traffic caused by other users until the total network congestion where delays become unpredictable. The operator performance is quite insensitive to a fairly small data loss. Transmission delay causes a decrease in operator performance almost linearly. Jitter produces a disturbance in velocity.

In Internet telerobotics using TCP/IP sockets [7] and Vx-Works real-time multitasking system, reliable connectivity can be established but with delay jitter in the arrival rate of originally synchronous packets. Packet inter-arrival delay varies from a few ms to a few seconds and average delay for a small packet ranges from 50 ms to 100 ms over the US. Asynchronous packets do not preserve their chronological order. TCP/IP can be reasonably used for packet with 256 bytes with a 10Hz sampling rate.

In [8] a telerobot is implemented using TCP/ATM in which two LANs are connected to an ATM backbone. Specification of Quality-of-Service (QoS) includes application timing, criticality, clock synchronization, and reliability. The use of constant bit rate in ATM allows a tightly constrained transmission delay which is suitable for real-time applications. ATM is used to guarantee time service frequency at the computing nodes (QoS brokerage). The sampling intervals reported are 0.4, 0.3, and 0.2 s for TCP/IP, raw ATM, and TCP/IP over ATM, respectively. A scattering transformation is proposed to overcome the problem of jitter in bilateral feedback systems [9]. Data is sampled at constant rate and transmitted with its sampling time which is ised to set up the data release time.

To improve real-time performance of telerobotics a multi-threaded execution is proposed for live transfer of force, command, and stereo data. Different software optimizations are statistically analyzed. Delays are evaluated using three campus routes with different network loads. Thread engineering is use for minimizing overall transfer time of stereo video data. Specification of video and force packet times are presented and analyzed. Effects of non-deterministic surge in network load are presented. Using the above real-time telerobotic system, contact between the slave arm tip and environment is presented using kinesthetic force display.

The organization of this paper is as follows. In Section II the network specification are presented. In Section III the
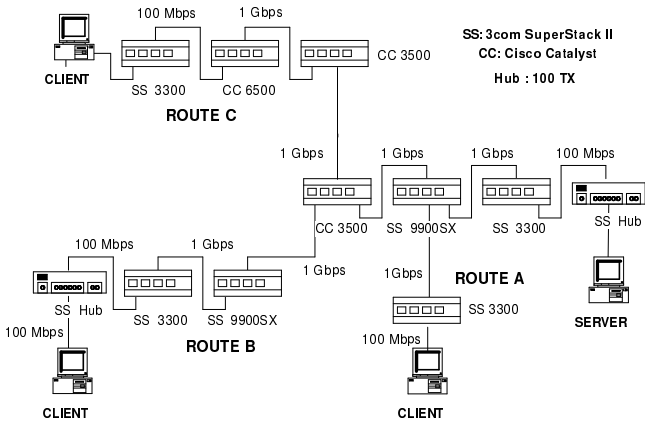
Fig. 1. Campus network routes used between a fixed server and moved client

Fig. 2. Distribution of inter-arrival times of force packets

video and force delays are statistically presented and evaluated. Section IV presents a comparison of achieved performance to other contributions. Conclusion and future directions are presented in Section V.

II. NETWORK AND SYSTEM SPECIFICATION

In this section the specification and configuration of the network, software, and computers used is presented prior to addressing the performance analysis of the proposed telerobotic framework. Detailed design aspects of proposed telerobotic system can be found in Part 1 of this paper.

The client and server are run on two PCs having 2-GHz Intel P4 processors with 1GB DRAM and 512 KB cache memory. Control of master and slave arms is done using Eagle PCI 30FG data acquisition cards. Each of client and server PCs is attached to a campus network by using a 100 Mbps NIC card (3com EtherLink XL PCI). The server PC is interfaced to two Sony Handycam digital cameras using a 400 mbps FireWire PCI (IEEE-1394) card. The client PC uses an NVIDIA GetForce4 Ti4600 as display adaptor to interface with an SVGA resolution Cy-visor DH-4400VP 3D head-mounted display.

Both client and server PCs run under MS Window 2000. The vision server software uses MS Visual C++ with .NET framework 1.1 under Microsoft development environment 2003. The imaging device used is Microsoft DV camera and VCR. The PUMA server is implemented using MS Visual C# with the above .NET framework.

Network delays are studied using three campus networks denoted by routes A, B, and C as shown on Figure 1. The server station is fixed in one location and the client station is moved to each of the three terminals in routes A, B, and C. The connectivity between the client and server stations is defined as follows. Refer to above figure for networking specification and component abbreviations. In route A, each packet travels across three L2/L3 switches (SS 3300 and SS 9900 SX) and a 100 Mbps hub (SS 100 TX). In route B, each packet travels across two L2/L3 switches using 100 Mbps input links (SS 3300), one L3 backbone switch (CC 3500) that uses 1 Gbps link at input and output, and two L2/L3 switches (SS 9900SX) using 1 Gbps links a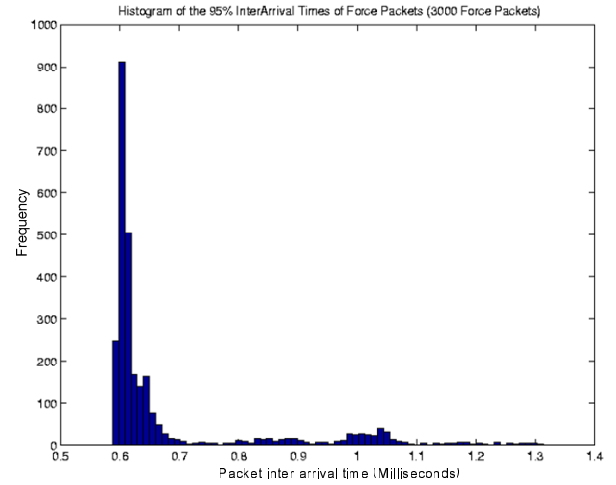nd two 100 Mbps hubs. In route C, each packet travels across three L3 backbone switches (two CC 3500 and one CC 6500) working as routers using 1Gbps links, three L2/L3 switches (SS 3300 and SS 9900SX) and a 100 Mbps hub.

In the next section analysis of telerobotic delays is presented in above three campus networks.

III. PERFORMANCE EVALUATION

Evaluation is carried out at the following levels: (1) streaming force in presence of video, and (2) thread engineering and delays in live transfer of stereo video.

A. Streaming force in presence of video

The performance of streaming force feedback between the server and client stations mainly depends on (1) total delay and (2) inter-arrival rate.

Denote by NU the percentage of network utilization. We first present the performance obtained for route A and later we repeat the experiments for route B and C. The above network interconnects computing systems for faculty, administration, and PC labs and provide access to Internet. The reference NU during the running of proposed telerobotic system with the above utilization is measured as about 10% for 1 Gbps links and 80% for 100 Mbps links of routes A, B, and C.

The force feedback thread is responsible of reading the force sensor, computing the force and moments at the tool frame, and transmitting the force data to the client station. We expose the running of the force thread to factors that may affect its performance such as concurrent CPU thread execution, network communication, and change in the route between the client and the server. Each force data packet is 48 bytes. Each video picture is 288×360 pixels and each pixel is 3 bytes. One picture is 0.3 MB. A stereo frame consists of 2 pictures or a data volume of 0.6 MB which requires a bandwidth of 5 Mbps/Frame for its network relaying.

The server throughput on network in the case of streaming only force packets is about 1 KHz as shown on Figure 2. Specifically for 90% of the cases the inter-arrival times of
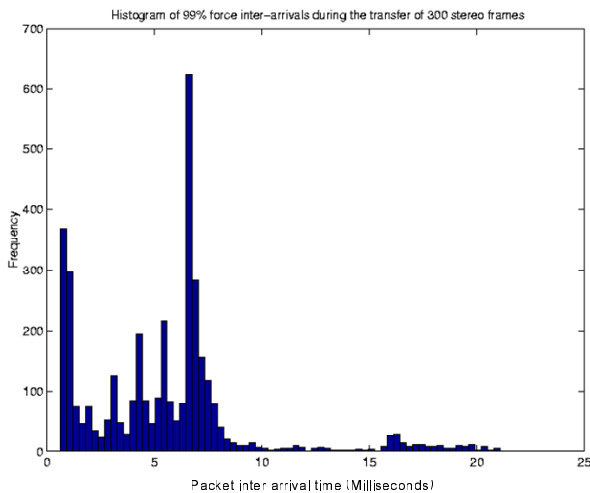
Fig. 3. Distribution of arrival times of force packets during video transfer

Fig. 4. Distribution of copy times from SampleGrabber to main memory

force data packets is below 1 ms. The average time to copy one stereo frame from the SampleGrabber to the DRAM is 24 ms. However, the video copying time is increased to 60.5 ms if we enable a thread to only read force information without network transfer. If the network transfer of force packets is enabled, the video copying time increases to 33.5 ms because when force packets are transferred on the network the internal processor resources are exclusively used by the stereo copying thread.

The transfer of force and video information leads to a force packet rate of 250 Hz due to sharing of CPU and network resources among the force and video threads. In 90% of the cases the inter-arrival time is below 4 ms. Although the video thread was continuously active the video transfer was ON and OFF in this case. For route A, Figure 3 shows the distribution of force packet inter-arrival times during active video transfer instance (ON and OFF) which corresponds to a reference time of 8 ms (mainly from 1 to 8 ms) but may occasionally increase up to 21 ms. Therefore, the reference server rate of force packets is 120 Hz and may occasionally drop to 47 Hz. Testing shows that processing time of force packets (reading, computing, and packing) on the server station is negligible compared to its transmission time for all studied routes. We conclude that a reference for the total force packet time is 8 ms. Inspection of the inter-arrival times of force packets in presence of active video thread (ON) shows that the worst case corresponds to arrival times in a range of 6 ms to 21 ms.

In the case of multi-streaming of force, command, and video packets the average inter-arrival time of force packets is 1.1 ms and all the population remains under 8 ms. The force thread is having more opportunity due to the presence of a command thread. Some peaks appear on the force packet arrival times which correspond to periods of active video transfer. The delays caused by the implicit software and network transfer overheads in using .NET remoting and Shim assembly for streaming of force data are quite comparable to network protocol delays for the same routes using traditional TCP socket for data transfer.

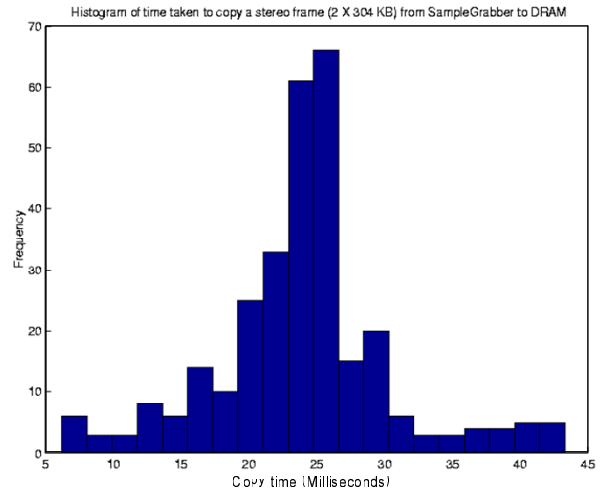The use of .NET technology for streaming of force packets

provides nearly the same inter-arrival delay for routes A, B, or C which is also comparable to the delay for one single hub using TCP sockets. The Gbps switches over all three routes normally operate with low NU values. Some sub-net may occasionally reach the congestion level which is manifested by an NU exceeding 80% on at least one 100 Mbps link of a given route. In this case the distribution of inter-arrival times of force packets, in the presence of live video transfer, becomes scattered and unpredictable in a range going from 8 ms to 30 ms.

B. Delays in live transfer of stereo vision

In this sub-section we study: (1) the delays in copying the video data from cameras to computer main memory, (2) performance of thread engineering for live video transfer in route A, and (3) video performance in routes B and C.

1) Copying from SampleGrabber to main memory: To measure the copying time on the server from SampleGrabber to main memory we consider two cases: (1) a single stereo thread, and (2) a stereo copying thread with a force thread.

In the case of a single stereo thread, the distribution of inter-arrival times of 300 video frames is shown in Figure 4. The mean value of 24 ms for which the 95% confidence interval falls below 25 ms.

In the case of a stereo copying thread from SampleGrabber to main memory with a force thread, the force is read as fast as possible without data transfer over the network. The mean copying time increased from 24 ms to 60.5 ms and 90% of the data lies between 8 and 150 ms. CPU time sharing (active force thread) has significant effect on stereo copying on the server. This is a useful feedback to the need for real-time operating systems and parallel I/O streaming.

In the case of stereo copying thread with force thread reading and transferring data over the network the mean stereo copying times decreased to 33.46 ms. The improvement over a copying time of 60.5 ms is due to the release of CPU resources to the video copying thread due to the use of blocking socket. This indicates that the multi-threaded execution of
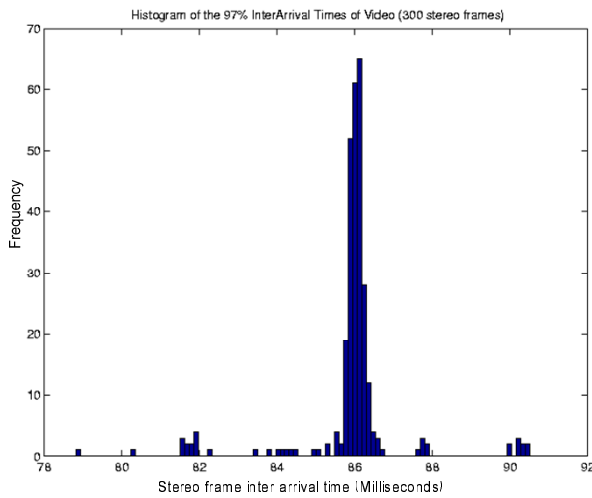
Fig. 5. Distribution of stereo frames using the serialized transfer approach
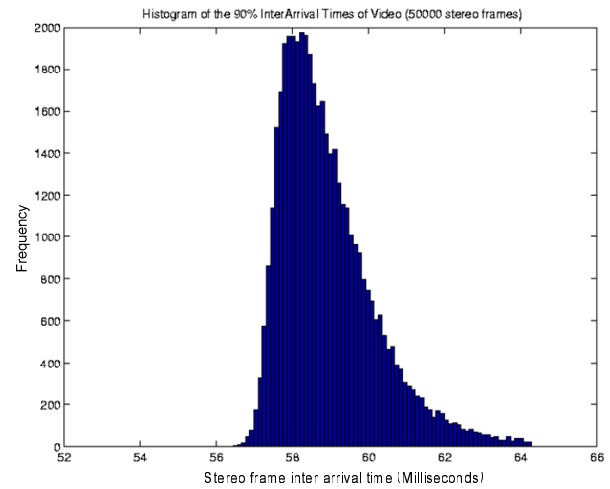

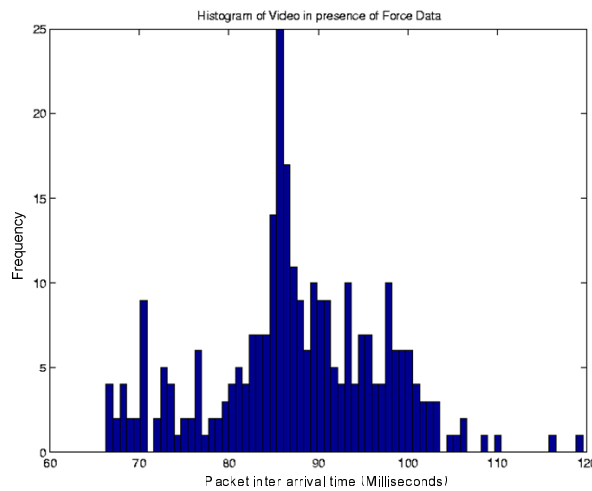Fig. 7. Distribution of inter-arrival times of stereo frames for route A

Fig. 6. Distribution of video packets with active force for serialized transfer

three concurrent threads is left to the best effort of Windows OS.

2) Live transfer of stereo video for route A: Performance of live video transfer is evaluated as follows: (1) a single buffer with serialized transfer, and (2) double buffer, concurrent transfer. Synchronous windows sockets are used for the client server video interfaces.

A single buffer is used in the serialized video transfer on server. The sending thread waits for the two SampleGrabbers to write stereo frame data to global buffer in order to send it to client with disabled display. Figure 5 shows the distribution of inter-arrival times of 300 stereo frames which is a Gaussian distribution with a mean of 86.5 ms or 11.6 fps. The distribution of inter-arrival times of video packets in the presence of force thread is shown on Figure 6. The above two distributions show the effects of resource and network conflicts caused by concurrent threads. Although the average arrival times are nearly the same the distribution is scattered due to indeterminism in the CPU execution and pre-emption of both video and force threads.

The double buffer, concurrent transfer scheme is evaluated

using a single buffer configuration on the server. Figure 7 shows the distribution of inter-arrival times of 50,000 stereo frames transferred between client and server with disabled display. Statistically this is a Gumbel distribution with a mean value of 59 ms and 90% of the data lying between 56 and 65 ms. A transfer rate of 17 fps is achieved. The maximum delay observed is 1299 ms which obviously is coming from network congestion and the minimum value is 53.5 ms. Time was saved in activating the SampleGrabbers and receiving the buffer ready notification. This proves that pipelining of video copying and transfer over the network is superior to traditional sequential processing demonstrated in serialized transfer.

Testing shows that a frame rate greater than 10 fps gives good viewing and refresh rate of 85 hertz eliminates any flickering. Some simple manipulation experiments, to move objects by looking at 3D scene on the computer screen wearing shuttering glasses and HMDs showed good depth perception of the viewer.

In summary, the lowest transmission delay for live video data transfer is observed in network A. While copying a stereo video frame to main memory takes 24 ms network transfer of live video takes 59 ms, e.g. an arrival rate of stereo frames of 17 fps. In other words, the total time of stereo frames is 83 ms between server and client from reading cameras to HMD display. The average delays caused by route A are similar to that obtained when both client and server are interconnected to one single LAN segment (Hub) in route A. Testing shows that total operator motion computation and command time is negligible at the client. However, the fastest motion on the available PUMA system is reasonably at the 100 ms level given its mechanical constraints and serial interface. Therefore, the total round trip delays from force sensing and video capturing to executing operator command in response to above feedback is about 183 ms which corresponds to an interaction frequency of 5.5 Hz. The Gbps switches and links operate at a lower rate than their real capacity. The above parameters can be considered as system references when NU is below the 80% reference level for each used 100 Mbps link on route A. The lack of accurate synchronization between force and stereo
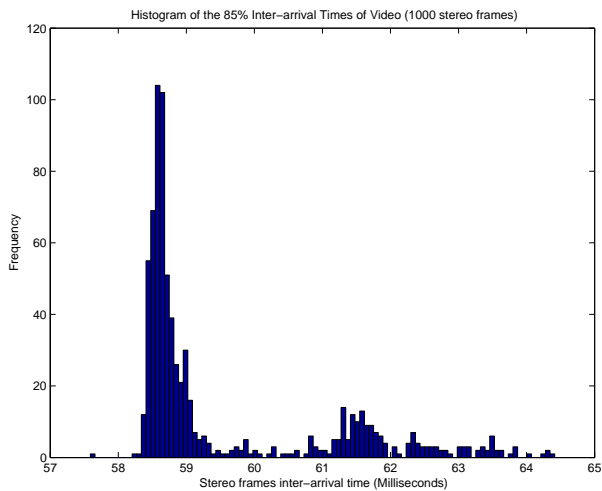
Fig. 8. Distribution and scattering of inter-arrival times of stereo frames in route B in presence of force packet streaming
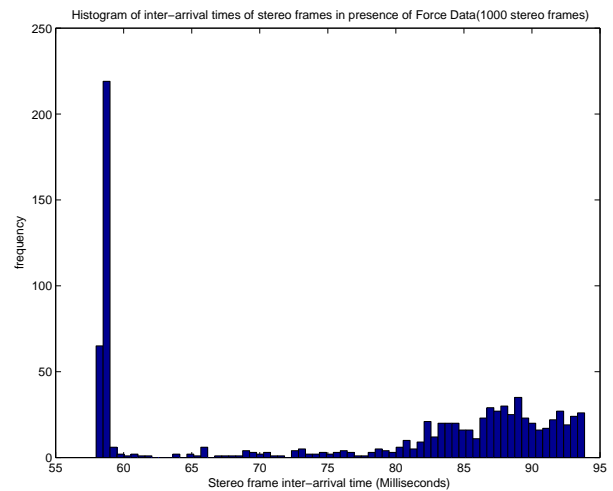

Fig. 9. Distribution and scattering of inter-arrival times of stereo frames in route C in presence of force packet streaming

frames has not shown to be critical in telerobotic experiments.

Occasionally an increase of NU above the 80% level for one or more of the 100 Mbps links (congestion) of route A leads the distribution of video and force inter-arrival times to be characterized as: (1) the dominant part of the distribution is still at the above reference values, and (2) some scattered distribution starts to appear in the region above the reference delays. At the above congestion level the width of the scattered region extends to 30% of the reference delays for the video. For example the scattering of the video distribution may extend to the range of 59 ms to 80 ms.

3) Live transfer of stereo video for routes B and C: In network B, it is noticed that when NU is below the 80% level for all 100 Mbps links the stereo and force frames preserve the same distribution pattern as in network A, but the whole distribution is slightly shifted with an increase in the average inter-arrival times to 60 ms for the stereo data. The increase in the reference delay of 1 ms is due to buffering of video data when hopping from one switch to another. The distribution of video packets for route B is quite similar to that of route A shown on Figure 7.

Route C includes three major gigabit L3 switches, three L2/L3 switches (two 100 Mbps and one 1 Gbps), and one hub. Campus route C is used as an exit route for accessing the Internet which means that traffic burstiness (irregularity) is the highest for this route. Testing shows that route C has comparable average NU to that of other studied routes but with the highest degree of traffic irregularity. Here more than 90% of the population of stereo frames deviate by no more than ± 0.5 ms from the reference delays of route A. Similarly, when NU is below the 80% level for all used 100 Mbps links of route C the dominant part (90%) of the distribution remains in the range of 58-60 ms (stereo) and some scattering appears in the region 60-80 ms. Rare bursty traffic during which NU rapidly fluctuates between 10% and 20%, even for short time, causes the distribution to become rather uniform and unpredictable and may extend to the range of 60-120 ms for stereo frames. The start of distribution scattering of inter-

arrival times of video packets for routes B and C is shown on Figures 8 and 9, respectively.

A flow-control strategy for live stereo vision is to minimize overall stereo delays by considering a variety of cases between the following two extremes: (1) sending large packets without compression to save processing time, or (2) sending compressed video to save overall data transmission time. In the proposed framework uncompressed stereo video transfer showed to have less overall delays as compared to compressed video transfer over the studied routes. For telerobotics, one strategy is to develop a task-aware compression method for compressing background data and transfer of uncompressed, small volume, higher resolution, region-focused, video data that is essential for the current task. The operator may specify a relevant tool region to be dynamically tracked and transmitted with higher resolution and refreshing rate. A tracking algorithm detects motion in the relevant region and guides a selective compression algorithm. Thus by controlling the size of the uncompressed region and its resolution the user may set up a variety of scenarios between the above two extremes.

Nowadays network routers, buffers, switches, and links do not incur noticeable delays. Here telerobotic real-time packets hoping from one switch to another do generally accumulate a small amount of delay regardless of used route. Packets with smaller payload are less sensitive to surge in network load than packets with larger payloads. The problem of unpredictable arrival times of real-time packets is concerned with the instantaneous traffic condition rather than with delays accumulated in network switching and buffering. The non-deterministic nature of network load even for short period of time may cause severe degradation over the reference delays. Although this is the exception, the unpredictable inter-arrival times of real-time packets may significantly degrades the quality of teleoperation. In this case, there is need for a resilient telerobotic flow control to ensure smooth performance degradation under severe load conditions. One approach is a flow-control that activates remote emergency agents, at the slave site, to ensure task safety and continuity under excessive delays. At the client

station, a virtual environment may supply the operator station with kinesthetic and visual feedback to provide interaction continuity based on task locality, environment model, and history information.

IV. COMPARISON TO OTHERS

A telerobotic client-server framework [10], [11] is proposed for VB 6.0 and TCP ActiveX platforms. Read/write operations using TCP based custom protocols take 20-40 ms because of the software layers involved such as application, custom protocol, TCP ActiveX control, and Windows sockets etc. Transmitting a command signal of 48 bytes between the client and server stations takes 55 ms. In the proposed framework a similar force packet takes about 8 ms in the presence of both video grabbing and video transfer threads.

In a typical scenario when both client and server use .NET based components with TCP channels, optimized data transfer is reported [12]. TCP Channel uses a default binary formatter which serializes the data in binary form and uses raw sockets to transmit data across the network. This method is ideal if the objects are only deployed in a closed environment.

Internet-based telerobotic System [13] can be implemented using JAVA for network interfacing and video and C++ for the design of the robot controller. In a LAN setup, a transfer rate of 9-12 fps with time delays less than 200 ms for a single image of size 200×150 pixels is reported. Video images are compressed using JPEG technique. The Java-based [14] frame grabbing software takes one second for an image to move from camera to main-memory as compared to a mean value of 24 ms obtained by proposed multi-threaded approach using DirectShow. The reported video transfer rate is 1 frame every 3 seconds for a single image of 16 bit color depth over the Internet.

In the proposed approach, multi-threaded execution allows pipelining of (1) processing and (2) network transfer times. In comparison to the above results, the proposed stereo video client-server system transfers on a campus network uncompressed stereo frames of 288×360 pixels at a reference rate of 17 fps and a total time of 83 ms. Other achieved sampling rates are 120 Hz for force feedback and 50 Hz for operator commands.

V. CONCLUSION

A telerobotic system transmitting live motion commands, force feedback, and stereo video has been evaluated on three campus routes with different load conditions. To minimize real-time delays a multi-threaded programming has been used to restructure sequential processing into concurrent threads that are executed in a pipelined fashion to parallelize the CPU processing with network transmission. Pipelining of grabbing stereo data with live transfer over the network allowed (1) copying a stereo frame from cameras to memory in 24 ms, and (2) live stereo video transfer at a rate of 17 fps. When network load is below 80%, the reference sampling rates for force feedback and operator command are 120 Hz and 50 Hz, respectively. The total delays for force and stereo are 8 ms and 83 ms, respectively. The slave arm is operated at a

10 Hz rate which leads to a round-trip delay of 183 ms or 5.5 Hz. Inherent network routers do incur negligible delays to above reference rates. However, short traffic burstiness may cause noticeable scattering in the above reference rates. As future direction, we proposed a task-aware video compression guided by a vision algorithm which tracks a relevant region that is transmitted with higher resolution and refreshing rate than background data. A resilient flow-control is also proposed to activate emergency agents at slave station to ensure task continuity and safety during periods of excessive delays.

VI. ACKNOWLEDGEMENT

REFERENCES

[1] T. Imaida, Y. Yokokohji, T. Doi, M. Oda, and T. Yoshikwa. Ground-space bilateral teleoperation experiment using ETS-VII robot arm with direct kinesthetic coupling. *Proc. of the IEEE Int. Conf. on Robotics and Automation,ICRA 2001*, 1:1031 – 1038, June 2001.
[2] R.D. Howe and Y. Y. Matsuoka. Robotics for surgery. *Annual Review of Biomedical Engineering*, pages 211–240, 1999.
[3] S.E. Salcudean, N.M. Wong, and R.L. Hollis. Design and control of a force-reflecting teleoperation system with magnetically levitated master and wrist. *IEEE Transactions on Robotics and Automation*, 11(6):844–858, December 1995.
[4] S.E. Salcudean, N.M. Wong, and R.L. Hollis. A force-reflecting teleop-eration system with magnetically levitated master and wrist. *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1420–1426, 1992.
[5] Xi Ning and T.J. Tarn. Action synchronization and control of Internet based telerobotic systems. *Proc. of IEEE Inter. Conf. on Robotics and Automation*, 1:219–224, 1999.
[6] F. Paolucci and M. Andrenucci. Teleoperation using computer networks: Prototype realization and performance analysis. *Proc. of the 8th Mediterranean Electrotechnical Conference, MELECON'96*, 2:1156–1159, 1996.
[7] W.J. Book, H. Lane, L.J. Love, D.P. Magee, and K. Obergfell. A novel teleoperated long-reach manipulator testbed and its remote capabilities via the Internet. *Proc. of the IEEE Inter. Conf. on Robotics and Automation*, 3(6):1036–1041, 1997.
[8] F. Goktas, J.M. Smith, and R. Bajcsy. Telerobotics over communication networks. *Proc. of IEEE Conference on Decision and Control*, 3:2399–2404, 1997.
[9] Kazuhiro Kosuge, Hideyuki Murayama, and Koji Takeo. Bilateral feedback control of telemanipulators via computer network. *Proc. of the 1996 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 3:1380–1385, Nov. 1996.
[10] A. Al-Harthy. Design of a telerobotic system over a local area network. *M.Sc. Thesis, King Fahd University of Petroleum and Minerals*, January 2002.
[11] M. Al-Mouhamed, O. Toker, and A. Al-harthy. A 3D vision-based man-machine interface for telerobotics. *IFAC Inter. Conference on Intelligent Control and Signal Processing*, April 8-11 Portugal, 2003.
[12] Microsoft. http://msdn.microsoft.com/default.asp, 2002.
[13] Huosheng Hu, Lixiang Yu, Pui Wo Tsui, and Quan Zhou. Internet-based robotic systems for teleoperation. *International Journal of Assembly Automation*, 21(2):143–151, May 2001.
[14] Teresa Ho. System architecture for Internet-based teleoperation systems using java. Master's thesis, Department of Computing Science, University of Alberta, Canada, 1999.