Metrics for QoS in Real-Time Interaction over the Internet

Christophe Salzmann and Denis Gillet

***Abstract*-- Real-time Interaction over the Internet (RTI2) is a new kind of Internet service that is required typically by remote experimentation applications. From a quality of services (Qos) point of view, RTI2 has constraints that differ from usual multimedia services such as video-conferencing or video streaming. While most of the classical Internet applications retrieve stored multimedia content with loose time constraints, in RTI2 the information representing the actual behavior of a physical system remotely operated should be provided to the user in a minimal delay.**

This paper presents the metrics needed to define the level of interaction in RTI2. These metrics are essential to successfully implement an end-to-end (E2E) control scheme which adapts the data transmission no only to the network load, but also to the server and client processing and display capabilities. The proposed scheme is based on a cascade structure that permits E2E adaptation to provide the user with the necessary level of interaction.

***Index terms*—Quality of service, Internet, metrics, real time, interaction**

I. Introduction

The key issue in implementing real-time interaction over the Internet with physical systems is to enable the control and the perception of their dynamics at distance. Since interaction is involved, the need for the operators to get feedback as quickly as possible on actions carried out is a constraint that requires dedicated solutions [1].

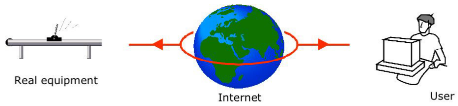

Fig. 1. Typical remote experimentation setting

School of Engineering, Swiss Federal Institute of Technology – EPFL, Switzerland, *first.lastname*@epfl.ch

RTI2 aims at providing the user with the best possible feedback such that he/she is not disturbed by the drawbacks inherent of the distance. There are three key aspects that need to be satisfied in order to provide a suitable quality of service, namely the level of interaction, the perception of the dynamics and the transmission of semantic information sufficient to reproduce the state of the distant system and its conditions of operation.

A. *Interaction*

The level of interaction can be characterized by the delay observed between the time an action is performed by user and the time its effect can be visualized by the user. The delay represents the round trip time measured at the application level. This delay is function of many factors -especially buffers- that will be described thereafter.

B. *Dynamics*

The dynamics of the distant system needs to be perceived at the client side. If the pace at which the information is acquired by the server and delivered to the client application is not adequate, the user might get a biased or wrong perception of the actual behavior of the distant system.

C. *Semantic content*

The semantic content has to be rich enough to enable the estimation of the state and the conditions of operation at the client side. There are various options to provide this information. Video image, Virtual-Reality representation or measurements displayed in oscilloscope graph can be used for that purpose. For a given type of representation, the more qualitative information is sent, the better the state can be perceived. For instance, a good quality picture, bigger in size, is more informative than a low quality image, smaller in size.

The three key aspects of RTI2 – interaction, dynamics, semantic content - can be translated in three communication parameters – delay, pace, size - that can be measured. Using these values, dedicated real-time solutions to implement RTI2 are explored.

D. IO implementation solutions

Various solutions can be explored to efficiently implement real-time interaction over the Internet. While the video streaming solution looks suitable for RTI2, the use of buffers to smooth the Internet bandwidth variation and to display images at a constant rate to the user makes it inappropriate since the buffering process add delays to the transmission [2]. Video conferencing is another real-time application that seems comparable to RTI2 but it carries differences: the priorities for the video and the audio are inverted. Sound is preponderant to image for video conferencing while this is not the case for RTI2. There might even be not sound at all. In video conferencing the amount of data transferred between the two parties is generally symmetrical, this is not the case for RTI2 where only a small amount of data goes from the client to the server but a large amount of data flows from the server to the client. Another difference is the scalability of the used bandwidth. RTI2 bandwidth usage ranges from a few bytes per second to Kilobytes per second. The former bandwidth corresponds to a client application running on a PDA with a Bluetooth network access and the later correspond to a client running on a desktop computer with a LAN access. The lowest values cannot be considered for video conferencing due to sound quality constraints.

A straightforward solution to implement RTI2 is to use a communication channel that can guarantee a given quality of service [3], such as a given bandwidth and latency, via reservation or by other means. This can be done by placing additional intelligence in the network at the routers level. While this solution might be a promising one, it not only requires a widely accepted agreement among manufacturers and providers regarding new communication protocols, but also asks for expensive software upgrades for most of the already deployed routers.

Instead of trying to modify the routers behaviors, the proposed solution is based on an end-to-end scheme that can be implemented at the application level.

II. END-TO-END METRICS

The End-to-End metrics takes into account the characteristics of the overall transmission path, from the information capture on the server to the information rendering on the client side.

The E2E metrics are the achieved block size and the achieved block pace. A block is defined as the aggregated information that represents the state and the operating conditions of the distant system at a given time. For example, a video image combined with the measurements acquired concurrently correspond to a block. The achieved block size is the amount of data that is effectively transmitted at a given time. This time is derived from the pace used by the server to send the block. The achieved pace corresponds to the E2E pace measured at the client side. The source block pace and size may vary from one block to the other due to the adaptation scheme, therefore the block has to carry these informations within.

The ratio between the achieved block size and the achieved block pace correspond to the achieved E2E bandwidth. Different block sizes and block paces, corresponding to different QoS settings, can produce the same achieved bandwidth. For example a bandwidth of 100 Kb/sec corresponds to 10Kb blocks sent 10 times per second, but also corresponds to 20 Kb blocks sent 5 times per second. If the client computer such as a PDA were not able to decode and play more than 5 images per second, half of the image would be waste with the first set of values. These metrics permit the implementation of an adaptation scheme (described in section 4) for the end-to-end structure defined thereafter.

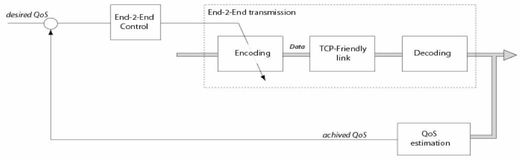

Fig. 2. End-to-End structure and adaptation scheme

III. END-TO-END STRUCTURE

In RTI2 the system to be controlled correspond to end-to-end information transmission path which goes through three main steps: the system state encoding, the transmission of the information over the Internet and the decoding and the rendering at the client side (Fig. 2).

E. Encoding

The encoding consists in transforming the system state and its conditions of operation to its digital representation. This is mainly done via an AD/DA converted connected to an analog sensor. One of theses sensors can be CCD camera that would produce an image. The image can then be compressed to reduce the size of the transmitted data. Other source of data such as measurements made via a DAQ board can be aggregated to the image to form a block. The pace at which the blocks are sent is function of the system dynamics. The encoding step is characterized by the time taken to produce a block of a given size.

F. Decoding

The decoding step is very similar to the encoding one. The data just goes the reverse path, first the block decomposition, then the data decompression and finally the information rendering.

G. Transmission

The transmission step differs from the above since there is no handle to control the transmission over the Internet once the data leave the computer and until it is received at the other end. This is due to the non-deterministic aspect of the network and to the nature of the protocol used to transmit the data. The routers along the transmission path simply do their utmost to deliver the data to the receiver as fast as possible despite the variation of the available bandwidth due to the network load. In other words neither bandwidth nor latency can be guaranteed.

The Encoding-Transmission-Decoding steps define the E2E transmission structure which the E2E application should adapt to while following the user desired QoS settings.

IV. End-to-end adaptation scheme

The adaptation scheme is based on the achieved size and achieved pace representing the achieved QoS. The idea is to adapt the block size and/or the block pace in order to adapt to the E2E bandwidth according to the user QoS settings. This adaptation scheme allows the use of a wide variety of client systems, from multi-GHz computers to PDAs.

A cascade approach is proposed to describe the two adaptation loops (Fig. 2). The inner loop is the TCP-Friendly link which represents the transmission loop that follows the network dynamics (as defined in section 5), and the outer loop being the E2E transmission loop that adapt to the E2E bandwidth. The adaptation scheme modifies the block pace and/or block size in accordance to the user defined QoS settings.

The figure 3 shows the QoS measurements made at the client side. The achieved block size is measured at time *T*, which derived from the expected block pace. When compared to the original block size, only 70% of the expected block is received. With these information the E2E adaptation scheme can decide to either reduce the block size or increase the time between blocks in order to adapt to the E2E bandwidth represented by the slope.

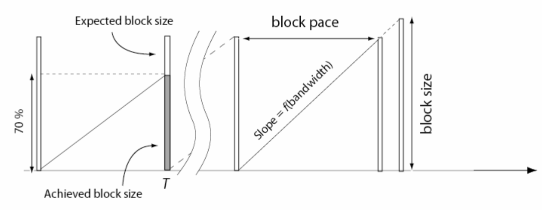

Fig. 3. E2E measurements and adaptation

The E2E adaptation scheme permits to take adequate measures when the network is not the bottleneck or when there is a large discrepancy between the server and the client system performances.

While the dynamics of the blocks is controlled by the E2E adaptation scheme, the dynamics of the packets transmission over the network follows the rules defined by the mandatory TCP-Friendly protocol.

V. TCP-Friendly communication

The aims of the E2E scheme is to adapt to the E2E performances while fairly using the Internet bandwidth. The fair adaptation to the network load is guaranteed by the use of a TCP-friendly communication that follow the

Additive Increase Multiplicative Decrease rules that are defined in the TCP protocol [4,5]. Nowadays TCP is the predominant protocol used for the Internet transmission such as web page transfer (http) or file transfer (ftp). TCP provides a reliable transmission meaning that all packets sent by the server will be delivered to the client [6], this implying that lost packets will be retransmitted. TCP also has a built-in mechanism designed to probe and to adapt the sending rate of the packets to the available bandwidth [7]. Provided that most of the Internet traffic relies on TCP (> 90%), the bandwidth will be fairly shared among these TCP connections using the network infrastructure. However, applications can use other protocols such as UDP which do not have a bandwidth adaptation mechanism [8]. If UDP-based applications do not implement their own TCP-friendly adaptation mechanism or implement a scheme that is more aggressive than TCP, the fairness among TCP and UDP connections is not respected and the aggressive source gets a bandwidth share bigger than what it should have [9]. While this behavior is tolerated in today's network policy, routers might discard transmission exhibiting such behavior in the future.

The TCP adaptation mechanism is well suited to transfer bulk data. However, this mechanism makes TCP inefficient for real-time interaction. The main problem is the delay introduced by the retransmission of lost packets. By the time the re-sent packet reaches the receiver, the real-time information contained in the packet is outdated. Therefore it is more adequate for interaction applications to ignore about the lost data and to send up-to-date information. Moreover the retransmitted information will be transferred at half the previous sending rate (AIMD rules) thus adding latency to the transmission.

A control protocol implanting the "Most Recent" principle to avoid re-sending outdated information useless for interaction has been designed [10]. The TCP Most Recent (TCP-MR) principle is to mimic the TCP protocol from a packet handling point of view to guaranty fairness, while filling the packets with the most recent available data to avoid re-sending outdated data.

The tradeoff in deploying TCP-MR is to accept losses in order to gain interactivity. Such a tradeoff is acceptable if the adaptation is carried out at the application level, given that the semantic nature of the transmitted information is known at that level.

VI. CONCLUSIONS

In this paper the metrics needed to measure the QoS for RTI2 has been introduced. These metrics were derived from the three key aspects of RTI2 (interaction, dynamics, semantic content) to produce three values (delay, pace, size) that can be measured. While the minimal delay is guarantied by the use of the TCP-MR, the pace and the size must be measured at regular interval. These measurements are made at the application level in order not to only take into account the network characteristics but the whole End-to-End characteristics. This approach permits the development of an adaptation scheme for the proposed cascaded loops structure where the inner loops guaranties the required TCP-Friendly network adaptation and the outer loop guaranties the End-to-End adaptation. This two-loops approach has the advantage of permitting an easier adaptation to future network communication protocol and network policy.

VII. REFERENCES

[1] Gillet D., C. Salzmann and P. Huguenin (2001). Distributed Architecture for Teleoperation over the Internet. Lecture Notes in Control and Information Sciences 258: Nonlinear Control in the year 2000, Springer-Verlag, 399-407, London.

[2] Feng W.and J. Rexford (1997). A Comparison of Bandwidth Smoothing Techniques for the Transmission of Prerecorded Compressed Video, Proceedings of the IEEE INFOCOM conference, pp. 58-66.

[3] Aktan B., Bohus C.A., Crowl L.A. and M.H. Shor (1996). Distant learning Applied to Control Engineering IEEE Transaction on Education, 39, pp. 320-326.

[4] Chiu D. and R. Jain (1989). Analysis of the Increase/ Decrease Algorithms for Congestion Avoidance in Computer Networks. Journal of Computer Networks and ISDN, 17,1, pp. 1-14.

[5] RFC793 (1981). Transmission Control Protocol ftp:// ftp.isi.edu/in-notes/rfc793.txt.

[6] Steven, R. W. (1994). TCP/IP Illustrated, Volume 1: The Protocols. Addison-Wesley.

[7] Jacobson, V. (1988). Congestion avoidance and control. Proceedings of SIGCOMM Symposium on Communications Architectures and Protocols, pp. 314 - 329.

[8] RFC768 (1980). User Datagram Protocol ftp:// ftp.isi.edu/in-notes/rfc768.txt.

[9] Floyd S. and K. R. Fall (1999). Promoting the use of end-to-end congestion control in the Internet. EEE/ACM Transactions on Networking, 7, 4, pp. 458-472.

[10] Salzmann Ch. and D. Gillet. Real-time interaction over the Internet. 15th Triennial World Congress of the IFAC, Barcelona, 21–26 July 2002 .