NETCICATS: Network-Conscious Image Compression and Transmission System*

Sami Iren[1], Paul D. Amer[1] and Phillip T. Conrad[2]

[1] Department of Computer and Information Sciences
University of Delaware, Newark, DE 19716 USA
Email: {iren,amer}@cis.udel.edu

[2] Department of Computer and Information Sciences
Temple University , Philadelphia, PA 19122 USA
Email: conrad@acm.org

Abstract. NETCICATS is a software system for empirically evaluating *network-conscious image compression*, an approach that does not simply optimize compression, but which optimizes overall performance when compressed images are transmitted over a lossy packet-switched network such as the Internet. Based on Application Level Framing, an image is compressed into path-MTU-size Application Data Units (ADUs) at the application layer. Each ADU contains enough information to be processed independently of all other ADUs. Each ADU can be delivered to the receiving application out-of-order, thereby enabling faster progressive display of images. NETCICATS allows the empirical investigation of the combination of transport protocol features and compression algorithms that perform best over a lossy packet-switched network. It includes software components from the network layer (e.g., lossy router), transport layer (e.g., innovative transport protocols), and application layer (e.g., compression algorithms, browsers, etc.). We describe each component of the system and explain how the whole system is used. This paper also presents two network-conscious image compression algorithms: network-conscious GIF and wavelet zerotree encoding.

1 Introduction

For many years, developments in image compression had one primary objective: obtaining the minimum image size. We argue that image compression algorithms should take into account that those images are likely to be transmitted over networks that will lose and reorder packets. Therefore, compression algorithms should not focus solely on achieving minimum image size; algorithms should be optimized to give the best performance when image data is lost or arrives out-of-order.

---

* Prepared through collaborative participation in the Advanced Telecommunications/ Info Distribution Research Program (ATIRP) Consortium sponsored by the U. S. Army Research Laboratory under the Fed Lab Program, Agreement DAAL01-96-2-0002. This work also supported, in part, by ARO (DAAH04-94-G-0093).

We apply the concept of *network-consciousness* [8] to *image compression*, an approach that takes network Quality of Service (QoS) into consideration when designing image compression. Network-conscious image compression is based on the philosophy of Application Level Framing (ALF) [2]. An image is divided into path-MTU-size[1] pieces, called *Application Data Units* (ADUs), at the application layer. Each ADU contains enough semantic information to be processed independently of all other ADUs. As a result, each ADU then can be delivered to the receiving application immediately, without regard to order, thereby potentially enabling faster progressive display of images.

Our approach to testing the network-conscious image compression hypothesis consists of two phases. In phase one, we developed Network-Conscious Image Compression and Transmission System (NETCICATS) to observe the relation between compression algorithms and transport protocols over network with varying characteristics. We want to see how different compression techniques behave when combined with different transport QoS at different loss rates. In phase two of our research, we modified two popular image compression techniques, namely GIF89a[2] and SPIHT [15] (wavelet zerotree encoding), to make them network-conscious.

Our research demonstrates that with a combination of innovative transport protocols and only a small penalty in compression ratio, today's standard compression algorithms can be modified to provide significantly better progressive display of images, and hence performance, in the Internet and wireless environments. In this paper, we describe the tools we have developed to investigate this approach, and present some experimental results gathered using these tools.

Section 2 describes network-conscious image compression and the motivation for it. Section 3 introduces NETCICATS. Sections 4 and 5 summarize two prototype implementations of our approach: network-conscious GIF and network-conscious wavelet zerotree encoding, respectively. Section 6 concludes the paper with a summary.

2 Network-Conscious Image Compression

A network-conscious compressed image is one that is encoded *not* simply to give the *smallest size* for a specified image quality, but to give the *best (i.e., smallest) response time - image quality* combination to an end user retrieving the image over a packet-switched network [10, 11]. The basic characteristics of a network-conscious compressed image are: (1) application level framing, (2) progressive display (preferably multi-layered), and (3) robustness and adaptiveness to different user needs and various networking conditions.

The key feature of network-conscious image compression is that it produces path-MTU-size self-contained blocks (ADUs) that can be decompressed indepen-

---

[1] MTU (Maximum Transmission Unit) is the maximum frame size that a link layer can carry. A path-MTU-size ADU is one that can be transmitted from source to destination without the need for IP layer fragmentation and reassembly.

[2] GIF89a is a Service Mark of CompuServe, Inc., Columbus, OH.

dently of each other. When these blocks are transmitted over a lossy network, they can be received and processed out-of-order, thereby permitting better progressive display. ADUs permit the use of a more efficient transport protocol that does not need to preserve order. Having simpler transport protocols is especially important for wireless environments because of their hosts' limited power supply [11].

Assuming some loss, the expected buffer requirements at the transport receiver for an unordered protocol are always less than the buffer requirements for ordered protocols [13]. Furthermore, out-of-order delivery of ADUs reduces the jitter at the receiving application. In ordered transport protocols, ADUs that are received out-of-order are kept in the buffers. When missing ADUs finally arrive, ADUs waiting in the buffer are delivered as a group to the application. This approach makes the delivery of ADUs to the application more bursty. The burstiness may result in bottlenecks at the receiving application [9].

Another advantage of compressing an image into ADUs is that their transmission can be tailored to each ADU characteristic. Not all parts of image data are uniform and require the same QoS. For example, low frequency coefficients (i.e., important data) of a wavelet image require a reliable service. On the other hand, high frequency coefficients (i.e., less important details) can tolerate a certain level of loss. Independent ADUs enable the use of different QoS such as reliability and priority for each ADU type.

Network-conscious compressed images are robust and can also adapt to different networking conditions easily. A lost or bit-errored packet will not destroy an entire image. A network-conscious compressed image can be transmitted over a very low bandwidth lossy network as well as a high bandwidth reliable network. The same compressed image can even be used, without any modifications, in a multipoint communication, where each participant has different requirements.

3 NETCICATS

NETCICATS allows us to empirically investigate the combination of transport protocol features and compression algorithms that perform best over a lossy packet-switched network. NETCICATS was flexibly designed for testing network-conscious image compression with several compression algorithms, a wide range of transport layer services, and on a variety of network conditions with different loss rates and bandwidths.

Since wavelet-based image and video coding has become popular, we based our compression algorithms on wavelet transformation. Recently, several wavelet-based encoding schemes have been developed which outperform DCT-based algorithms in terms of both objective criteria (bit rate versus distortion) and subjective criteria [6]. As Shapiro reports, "the main contribution of wavelet theory and multiresolution analysis is that it provides an elegant framework in which both anomalies and trends can be analyzed on an equal footing." This framework is important in image processing because, "edges, which can be thought of as anomalies in the spatial domain, represent extremely important information despite the fact that they are represented in only a tiny fraction of the image samples" [16].

Figure 1 depicts NETCICATS components to transmit wavelet-encoded images over a lossy, low-bandwidth network. Main components are (1) an image sender, (2) an image receiver, (3) a lossy router, and (4) a reflector.

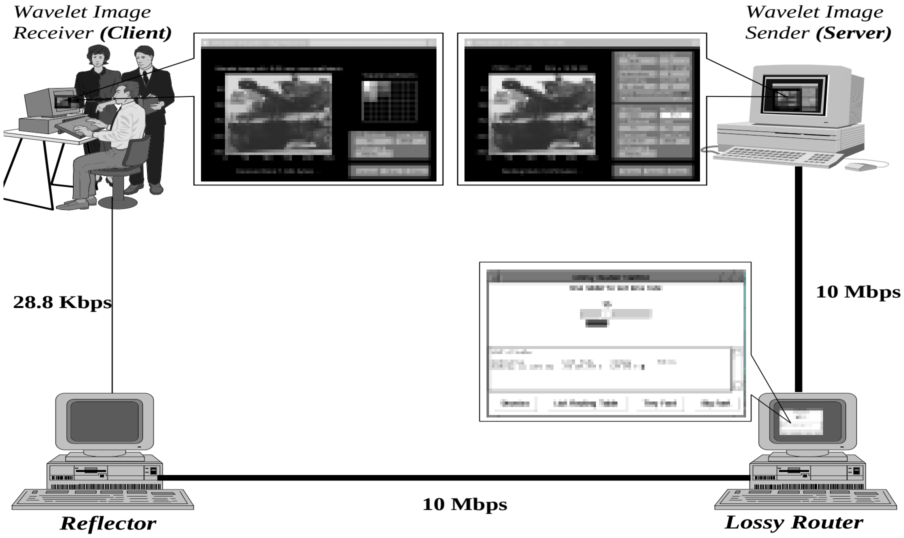

Fig. 1. Network-Conscious Image Compression and Transmission System

The *image sender* allows a user to flexibly control an image's quality and size, and the transport QoS between server and client. The image quality and size are controlled by user-adjustable parameters such as thresholding level (i.e., percentage of wavelet coefficients that are set to zero), quantization level (i.e., number of bits used for quantization), type of mother wavelet (e.g., Daubechies, Haar, Coiflet, etc.), levels of decomposition, and encoding method. The image quality can be measured both subjectively by visualizing the image, and objectively by using its PSNR.

The *image receiver* progressively displays image data as it arrives. In addition to the progressive image, a separate array of grids is updated as each ADU arrives. This grid identifies which wavelet coefficients are currently being displayed, and is especially useful to observe the effects of missing coefficients at different resolutions. A "stop" button allows a user at the image receiver to cancel the transmission of all additional ADUs without severing the transport connection. This saves network bandwidth when an image has progressed sufficiently for the receiver to make a decision (e.g., for tele-medicine—"transport patient vs. do-not-transport", or in situational awareness— "friend vs. foe", "target vs. non-target"). Halting transmission mid-way is made possible by the

ADN-Cancel feature of certain innovative transport protocols being developed by the authors. This feature allows cancellation of messages that have already been submitted to the transport layer. The application specifies an Application Data Name (ADN) for each message, and can cancel the transmission of any message (or group of messages) by specifying its (their) ADN [4]. ADN-cancel is a service that is not supported by either TCP or UDP.

In the University of Delaware Protocol Engineering Lab (PEL), the image sender and image receiver run on two different stations on the same Ethernet. To simulate packet loss on the Internet, routing tables direct all client/server traffic through a special *Lossy Router (LR)*. This LR allows experimenting with different levels of packet loss between the server and client. The LR is intended to run on a machine serving as an IP gateway. All of the normal routing functions of that host are replaced by this software. This way, any communications between two hosts which route their data through the LR will be affected. The current LR simulates any of three loss models: Bernoulli, deterministic, or 2-Step Markov. When forwarding each IP packet, the LR deliberately drops the specified percentage of packets according to the specific loss model. In the Bernoulli model, the given percentage of packets are dropped randomly. In the deterministic model, every k^{th} packet is dopped, where k is a user-controlled parameter. In the 2-Step Markov model (also called Gilbert loss model), there are two states, and losses only occur in the “loss” state (see Figure 2). The average loss rate, L, is defined by $L = p/(p + q)$. This model represents losses with burstiness.

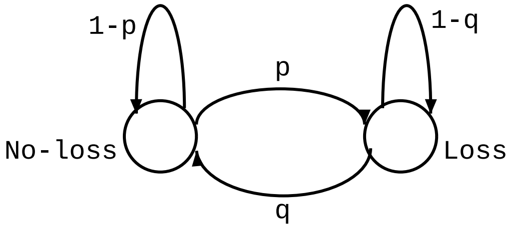

Fig. 2. 2-Step Markov Loss Model

The LR is controlled with an FTP-like protocol called the Simple Lossy Router Protocol (SLRP). This is a simple text-based control protocol that provides the communication between the LR user and the LR software. The LR user can set the loss rate and loss model, list and update the routing table on the lossy router, and collect statistics. A user-friendly interface, called the lossy router control client, hides the details of this protocol from users.

The *Reflector* functions in a similar manner to the LR, except that instead of dropping IP packets, the Reflector delays their forwarding to simulate a lower bandwidth link. This software is useful when an actual lower bandwidth link (such as a PPP link) is not available. Recently, we have replaced this component with U.S. Army SINCGARS combat net radios (see Figure 5).

Three different encoding methods have been implemented: run-length encoding based on path-MTU-size blocks, LZW encoding based on path-MTU-size blocks, and LZW encoding based on an entire image. The path-MTU is another user-defined parameter provided to test the effects of segmentation/reassembly on image data. In the first two encoding methods, each ADU contains enough information to be decoded and located in the receiver's progressive display (i.e., network-conscious compressed image). The authors recognize that these encoding methods may not be optimal methods for encoding wavelet coefficients. NETCICATS is designed to easily integrate future encoding methods for evaluation. Section 5 discusses one potential better encoding technique to encode wavelet coefficients.

Once an image's quality and size are decided, the NETCICATS' user selects among several different transport protocols and QoS via the Universal Transport Library (UTL), a library of transport protocols that provides application programmers the ability to write to a single Application Programming Interface, then test their application with many different transport protocols [4]. UTL is a library of C functions that a programmer can link with an application. The application can then vary the transport protocol used by altering a single parameter on the "listen" call (passive open) or the "connect" call (active open). By allowing flexible experimentation with different transport protocols, the experimenter can isolate particular aspects of transport services to better understand each one's effects. Figure 3 illustrates how UTL is used by client/server applications. The services currently available to applications via the UTL API are presented in Figure 4.

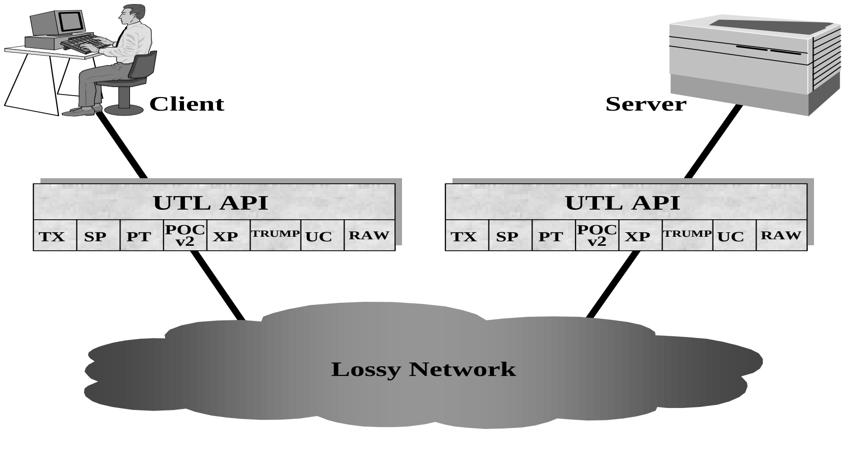

Fig. 3. How UTL is Used in Client/Server Applications

Since wavelet transformation provides a multiresolution representation (information at several levels of importance), NETCICATS allows the use of different QoS paramenters for different resolutions, such as using a specific degree of partial reliability for each layer of information.

UTL API							
TX	**SP**	**PT**	**POCv2**	**XP**	**TRUMP**	**UC**	**RAW**
•Ordered •Reliable	•Ordered •Reliable •Access to buffered data	•Partially ordered •Reliable •Sync support	•Partially ordered •Partially reliable •Sync support •Access to buffered data	•Unordered •Partially reliable	•Unordered •Partially reliable (time-based; application provides staleness time for each message)	•Unordered •Unreliable	•Pass-thru service

Fig. 4. Services Available to Applications via UTL

Recently, we have added some new components to NETCICATS: three U.S. Army SINCGARS combat net radios for doing experiments in a wireless, low-bandwidth environment. Although the signaling rate of these radios is 16 Kbps, the effective throughput is less than 2 Kbps. We have also added a browser, much like a conventional web browser, for experimenting with the two modified network-conscious image compression algorithms (see Sections 4 and 5). This new environment is shown in Figure 5.

One of the authors' motivations when designing NETCICATS was to develop a system for a hypothetical military communications system for transmitting images of either (1) wounded soldiers for telemedicine, or (2) images of equipment such as tanks, airplanes, etc. for intelligence gathering [3]. Therefore, we have run experiments primarily with military-related images such as tanks, airplanes, missiles, etc.

Here are some of the author's initial observations on using NETCICATS. These observations support our hypothesis on network-conscious image compression.

- Eliminating the ordered delivery requirement of compressed image data certainly provides faster progressive display.
- Transmitting wavelet coefficients that affect the image quality most (i.e., low frequency wavelet coefficients) as early as possible improves progressive display.
- Some applications may tolerate loss of ADUs that carry low-level detail information (i.e., high frequency wavelet coefficients) as these ADUs have little

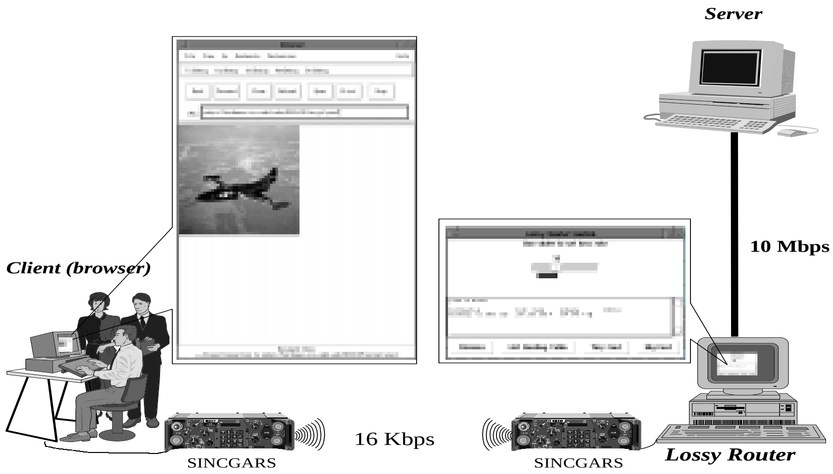

Fig. 5. NETCICATS with SINCGARS Radios and Browser

effect on the final image quality. However, loss of a higher level information severely affects the image quality.

The next step is to modify one or more of the traditional image compression algorithms to make them network-conscious, and compare their performance over a lossy packet-switched network. Sections 4 and 5 talk about two such algorithms.

4 Network-Conscious GIF

We have modified the GIF89a standard to make it network-conscious. The result, called GIFNCa [1], removes the ordered delivery requirement (and for some applications the reliability requirement) of GIF89a by framing image data at the compression phase (i.e., application level).

The tradeoff between GIFNCa and GIF89a is one of compression vs. progressive display performance. GIF89a's advantage is its expected better compression. GIFNCa's advantage is its expected *faster progressive display* at the receiver when transmitted over an unreliable packet-switched network such as the Internet.

We ran a set of experiments comparing (1) GIF89a over a reliable, ordered transport protocol called *Sequenced Protocol (SP)* vs. (2) GIFNCa over a reliable unordered protocol called *Xport Protocol (XP)*. SP and XP were both developed at University of Delaware's PEL as part of the UTL. Both SP and XP are implemented at the user-level over UDP, and use the same code for all functions (including connection establishment/teardown, round-trip-time estimation, retransmission timeout, acknowledgments, etc.); the only exception is that SP provides packet resequencing (i.e., ordered service) at the receiver, while XP does not.

The testing environment is quite similar to Figure 5 except that instead of SINCGARS radios, the Reflector was used to simulate a 28.8 Kbps link. This link is more than ten times greater bandwidth than that expected with the SINCGARS radios. Compressed images are stored on a server, and accessed by a client with an interface similar to familiar web browsers. The packets are routed through the Lossy Router and the Reflector to simulate loss and low bandwidth, respectively. Each experiment downloads a compressed image from server to client through the Lossy Router and Reflector.

The average percentages of image data being displayed at various points in time for 5%, 10%, and 15% IP packet loss rates are graphed in Figure 6. These graphs show that while both GIFNCa and GIF89a take longer to display the image as the loss rate increases, the GIFNCa performance does not degrade as quickly, i.e., it improves *relative to* GIF89a. This result is intuitive. As the loss rate increases, so does the number of buffered out-of-order packets at the receiving transport layer. These buffered packets are waiting for missing packets (in the case of GIF89a over ordered transport protocol). On the other hand, an unordered transport protocol (in the case of GIFNCa) delivers these out-of-order packets to the application (browser) as soon as possible after they arrive; no buffering for reordering purposes is needed.

While more serious and exhaustive empirical study is currently underway, these initial results highlight the potential benefit of using GIFNCa over GIF89a under lossy network conditions.

5 Network-Conscious Wavelet Zerotree Encoding

Wavelet zerotree encoding is based on the hypothesis that, at a given threshold level, if a wavelet coefficient at a coarse scale is insignificant, then all wavelet coefficients of the same orientation in the same spatial location at finer scales are likely to be insignificant [16]. The embedded zerotree (EZW) encoding, originally introduced by Shapiro [16], has been proven to be a very efficient yet not complex encoding scheme. The embedded nature of the algorithm, a representation in which a high resolution image contains all coarser resolutions, effectively sorts bits in order of importance, thus yielding an effective progressive display when transmitted over low-bandwidth networks. Using this embedded coding method, an encoder can stop encoding at any point to meet a target rate. Furthermore, a decoder can stop decoding at any point in the bit stream, and still produce exactly the same image that would have been encoded at the bit rate corresponding to the truncated bit rate.

Set Partitioning in Hierarchical Trees (SPIHT), introduced by Said and Pearlman [15] as a refinement to EZW, differs from EZW in the way subsets of coefficients are partitioned and in the way significance information is conveyed.

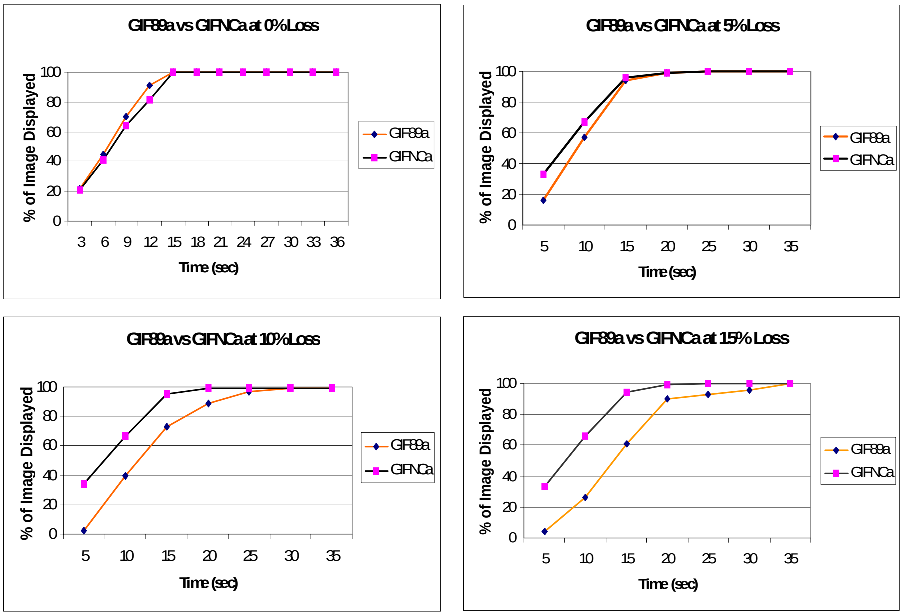

Fig. 6. Comparison of GIF89a and GIFNCa at Various Loss Rates

SPIHT is so effective that even binary uncoded transmission achieves about the same or better performance than EZW [15].

Both EZW and SPIHT are highly state-dependent, and therefore susceptible to bit errors. Even a single bit error ruins the decoding process thereby destroying an entire image. Recent studies concentrate on composing noise-robust zerotree encoders. Most of these studies are based on the idea of dividing the bitstream into several sub-streams each of which receive different amounts of error protection based on their noise sensitivity [12], or interleaving separately encoded substreams in so that any single bit error will corrupt only one substream [5, 6].

Recently, Rogers and Cosman [14] introduced a packetized zerotree encoding method on still images that produces fixed 53-byte packets and is robust against packet erasure. A similar study by Crump and Fischer [7] produced variable-length independent packets for video transmission.

Our algorithm differs since these previous studies do not consider progressive display. They solely optimize for robustness, and only the final image quality is a concern. Robustness is one of the primary features of network-conscious compression approach. Therefore, by applying network-consciousness to zerotree encoding we solve that problem by default. Our concern also considers how to modify these algorithms so that they provide a better progressive display under lossy network conditions.

We have modified the SPIHT algorithm for this purpose. Our method produces path-MTU-size, independent packets which are optimized to provide better progressive display. The algorithm is currently in its experimental phase.

6 Conclusion and Future Work

Traditional image compression algorithms are not designed for lossy packet-switched networks and heterogeneous environments with wired and wireless links; they are optimized to minimize image size only. However, minimum image size does not necessarily provide the best performance when those images are transmitted over lossy networks, and used by the receiver for real-time decision making. The ordered-delivery requirement of these algorithms cause unnecessary delays at the receiving end.

This research investigates the relationship between compression algorithms and transport QoS parameters. Our results propose applying network-consciousness to image compression so that the compression algorithms will not be optimized only to give the minimum image size; they will be optimized to give the best performance when transmitted over lossy networks. We have developed NETCICATS to empirically evaluate network-conscious image compression, and two compression algorithms that utilize this approach: network-conscious GIF, and network-conscious zerotree encoding. Initial experiments for network-conscious image transmission are promising.

Our future study includes running experiments in the NETCICATS environment to collect and analyze extensive empirical data on network-conscious GIF and network-conscious zerotree encoding.

References

1. P. Amer, S. Iren, G. Sezen, P. Conrad, M. Taube, and A. Caro. Network-conscious GIF image transmission over the Internet. In *4th International Workshop on High Performance Protocol Architectures (HIPPARCH'98)*, June 1998.
2. D. Clark and D. Tennenhouse. Architectural considerations for a new generation of protocols. In *ACM SIGCOMM '90*, pages 200–208, Philadelphia, PA, September 1990.
3. P. Conrad, P. Amer, E. Golden, S. Iren, R. Marasli, and A. Caro. Transport qos over unreliable networks: No guarantees, no free lunch! In Nahrstedt Campbell, editor, *Building QoS into Distributed Systems*. Chapman and Hall, 1998. `www.cis.udel.edu/~amer/PEL/poc/postscript/iwqos97.ps`.
4. P. Conrad, P. Amer, M. Taube, G. Sezen, S. Iren, and A. Caro. Testing environment for innovative transport protocols. In *MILCOM '98*, Bedford, MA, October 1998. (To appear).
5. C.D. Creusere. A family of image compression algorithms wich are robust to transmission errors. In *IS&T/SPIE Wavelet Applications in Signal and Image Processing IV*, volume 2825, pages 890–900, August 1996.
6. C.D. Creusere. Image coding using parallel implementations of the embedded zerotree wavelet algorithm. In *IS&T/SPIE Symposium on Electronic Imaging*, volume 2668, San Jose, CA, 1996.

7. V.J. Crump and T.R. Fischer. Intraframe low bitrate video coding robust to packet erasure. In *DCC '97*, Snowbird, Utah, 1997. IEEE.
8. W. Dabbous and C. Diot. High performance protocol architecture. In *IFIP Performance of Computer Networks Conference (PCN '95)*, Istanbul, Turkey, October 1995. IFIP.
9. C. Diot and F. Gagnon. Impact of out-of-sequence processing on data transmission performance. Technical Report Project RODEO RR-3216, INRIA - Sophia Antipolis, France, July 1997. `ftp://www.inria.fr/rodeo/diot/rr-oos.ps.gz`.
10. S. Iren. Network-conscious image compression. PhD Dissertation, CIS Dept., University of Delaware, (in progress).
11. S. Iren, P. Amer, and P. Conrad. Network-conscious compressed images over wireless networks. In *5th International Workshop on Interactive Distributed Multimedia Systems and Telecommu nication Services (IDMS'98)*, Oslo, Norway, September 1998.
12. S.H. Man and F. Kossentini. Robust EZW image coding for noisy channels. *IEEE Signal Processing Letters*, 4(8):227–229, August 1997.
13. R. Marasli, P. Amer, and P. Conrad. An analytic model of partially ordered transport service. *Computer Networks and ISDN Systems*, 29(6):675–699, May 1997.
14. J. Rogers and P. Cosman. Robust wavelet zerotree image compression with fixed-length packetization. In *Data Compression Conference (DCC'98)*, March 1998.
15. A. Said and W.A. Pearlman. A new, fast, and efficient image codec based on set partitioning in hierarchical trees. 6(3), June 1996.
16. J. Shapiro. Embedded image coding using zerotrees of wavelet coefficients. *IEEE Transactions on Image Processing*, 41(12):3445–3462, December 1993.

This article was processed using the LaTeX macro package with LLNCS style