Motion Video Coding for Packet-Switching Networks — An Integrated Approach

Michael Gilge and Riccardo Gusella †

International Computer Science Institute
1947 Center Street #600, Berkeley, California 94704
email: gilge@icsi.berkeley.edu, gusella@tenet.berkeley.edu

ABSTRACT

The advantages of packet video, constant image quality, service integration and statistical multiplexing, are overshadowed by packet loss, delay and jitter. By integrating network-control into the image data compression algorithm, the strong interactions between the coder and the network can be exploited and the available network bandwidth can be used best. In order to enable video transmission over today's networks without reservation or priorities and in the presence of high packet loss rates, congestion avoidance techniques need to be employed. This is achieved through rate and flow control, where feedback from the network is used to adapt coding parameters and vary the output rate. From the coding point of view the network is seen as data buffer. Analogously to constant bit rate applications, where a controller measures buffer fullness, we attempt to avoid network congestion (eq. buffer overflow) by monitoring the network and adapting the coding parameters in real-time.

1. INTRODUCTION

The transmission of video over packet-switching networks enables novel applications based on the integration of several services into a common format. It is believed that the integration of audio, video, and discrete data into multimedia-workstations connected by packet-switching networks will have as much impact as the introduction of the personal computer itself.

Conventionally, only circuit-switching, application-specific networks have been used for video, e.g. CATV. Usage of these constant-rate channels for variable-rate transmission results in wasted capacity since the channel has to accommodate worst-case rates. Using packet-switching network, video delivery with variable rate and constant quality can be achieved efficiently by virtue of statistical multiplexing[2,20]. A thorough analysis and modelling of statistical multiplexing is given in a paper by Maglaris et al.[23].

In order to exploit the advantages of packet video, service integration, constant quality, and statistical multiplexing, the following problems associated with the transmission of video over packet-switching networks have to be overcome: First is the discrepancy between the data rates that can be accommodated by today's computer systems and networks on the one hand, and the values associated with real-time video that have to be bridged through data compression techniques on the other hand. Fortunately, images contain a lot of redundant and irrelevant information that can be successfully exploited by several compression algorithms.

Secondly, the service of continuous media is associated with certain end-to-end requirements in terms of bandwidth, delay and jitter that are not met by today's packet-switching networks or computer systems. The general directions of research that are currently pursued can be categorized into two classes: The first approach aims at extending current operating systems and network protocols to provide performance guarantees[6,24]. These schemes necessitate admission control and will not be in service in the near future on a broad scale. The other approach adapts the continuous-media stream to the given systems, in which service is not guaranteed but is delivered on a best-effort basis. In these systems, packet loss, for example, has to be compensated for by the receiver. This leads to a complex problem, because not only are both channel capacity and data rate variable, but they also interact.

Several researchers have begun investigating video coding algorithms for transmission of video over packet-switching networks. Their work points out the advantages of constant image quality and bandwidth sharing by statistical

† Riccardo Gusella is currently with Hewlett-Packard Laboratories, 1501 Page Mill Road, Palo Alto, CA 94304

multiplexing; they propose that redundancy and error correction protocols be employed to deal with packet loss; but, in general, they fail to provide adequate solutions to the network congestion problem. In the literature we observed that either an image data compression viewpoint or a networks viewpoint is taken: image coding algorithms are mostly simulated using greatly simplified network models, e.g. fixed packet loss rate, and/or they assume the existence of network services not yet widespread such as networks, which respect different packet priorities or offer classes of service[7,17], and/or networks that guarantee the delivery of a certain percentage of the packets[10]. Network issues, on the other hand, are mostly based on models of the video source, e.g. Markov models[23].

Among the techniques proposed for packet video coding are vector quantization[12], subband coding[3,17], motion-compensated interframe coding[26], two-layer coding[10], pyramidal encoding[28], discrete cosine transform[20], and DPCM[30]. A good overview of new image coding results for ATM networks can be found in the recent Image Communication journals[13]. Network issues, which are also partly discussed in the literature cited above, are error recovery, congestion control[15,8], statistical analysis and modelling of packet loss[22], and characterization of video sources[25].

Although the mutual dependency of video coding and packet networks has already been pointed out by several researcher[1,18,7], no satisfactory solutions have been proposed so far. Since channel bandwidth varies with network load conditions, minimum packet loss will be achieved by senders that can adjust dynamically their maximum transmission rates to match the channel bandwidths. Motion video compression algorithms are suited to produce such changing maximum transmission rates as several compression parameters can be varied[4] in response to congestion signals from the network. Thus, we propose an integrated approach to motion video coding for packet-switching networks: the coding algorithm and the network (through a network estimator) are part of a feedback loop that seeks to avoid network congestion. Unlike other approaches that try to maintain constant picture quality, our scheme produces lower received signal quality when the network load increases. However, in general, we use bandwidth more efficiently since a coding agent ensures best image quality for a given bandwidth.

Section 2 of this paper introduces the general notion of feedback communication system and their applicability to packet video. This leads to a network-integrated solution where the networks serves as data buffer. The main components of this system – video coding agent and network estimator – are discussed in section 3. The results of experiments with a real-time software coder that uses local-area networks interconnected by routers to transmit video frames are reported in section 4.

2. FEEDBACK COMMUNICATION SYSTEM

One of the great promises of variable bit rate coding is constant image quality. Unlike in constant bit rate environments, where the quality of demanding scenes may be severely degraded in order to stay below line bit rate, packet networks offer capacity on demand. This advantage is overshadowed by the possibility of packet loss, which may affect image quality at the receiver even more severely than normal coding artifacts, and by the variability of packet delays, which may make it difficult to use packet-switching networks for interactive applications. Data transmission protocols overcome packet loss by retransmission at the cost of added delay and network load. Retransmission of lost packets for interactive services, e.g. video conferencing, is already infeasible because of large end-to-end communications delay. Retransmission should be avoided for continuous media services altogether because of the possibility of a positive feedback on network capacity: frequent retransmissions can increase network saturation and cause increased packet loss[14]. Depending on the size of the buffer space available, long packet delay may be observed in place of packet loss. Due to the mentioned real-time characteristic of continuous media, delayed packets must be discarded, resulting in wasted network capacity.

In order to take advantage of today's computer networks, which do not offer any bandwidth guarantees and do not support different classes of service, as required by many proposed coding algorithms, we have to find ways to lower the high packet loss rates and long delays that can occur in packet-switching networks. The packet loss rate produced by noise has been reported in the range from 10^{-6} for copper connections to 10^{-12} for optical connections. However, packet loss due to buffer overflow can be as high as 10^{-2}, or even higher, under severe congestion and can occur in any network. To make matters even worse, packet loss frequently occurs in bursts. Another factor that makes feedback communication so important is the fact that the introduction of gigabit-speed networks does not solve the congestion problem; on the contrary, the coexistence of low- and high-speed networks results in speed mismatch and increases the chances of congestion[16].

Error correction schemes are only useful for a small number of bit errors. Once the maximum number of correctable errors is exceeded, error correction fails. For packet loss, layered coding schemes in combination with error concealment techniques have proven to be very effective. However, techniques like error correction or redundant transmission by strongly overlapping coding layers are counterproductive to the goal of compact data representation and efficient channel utilization. Since packet loss is likely to impact image quality much more than an increased compression ratio, we argue that a coding algorithm should adapt to the network load conditions in the first place. It can be expected that image quality will benefit more from congestion avoidance than from the strict adherence to the constant quality through variable bit rate principle.

End-to-end services are supposed to operate at a wide range of data rates. This property is required in order to realize different qualities of service, but can also be exploited for congestion avoidance. In addition to data rates changes caused by the variable information content of video scenes, variability may be artificially increased by changing the coding parameters during data transmission. This of course necessitates a flexible coding scheme, one that operates at a range of compression factors, and a defined interaction between the application and the network. The idea is that the performance of the network channel (and system in general) at every instant in time determines the quality of the continuous media service. This way a graceful degradation of the image quality as the network load increases can be realized.

A scheme that negotiates constantly the data rate between the network and the coder is depicted in **Fig. 1**. The coded video data may be communicated to the network by a standardized description, as indicated by the term 'VideoPostScript'. The two building blocks can be interpreted as agents for the application and the network resource, respectively. Notice that **in current approaches, these two agents do not talk to each other:** the coder sends data blindly into the network, expecting timely and complete delivery at the receiving end. Clearly, without any knowledge of whether packets are received at the decoding end, or are dropped along the way, the resulting picture quality must be by far worse than what could be achieved at the same bandwidth with information about the available resources over time.

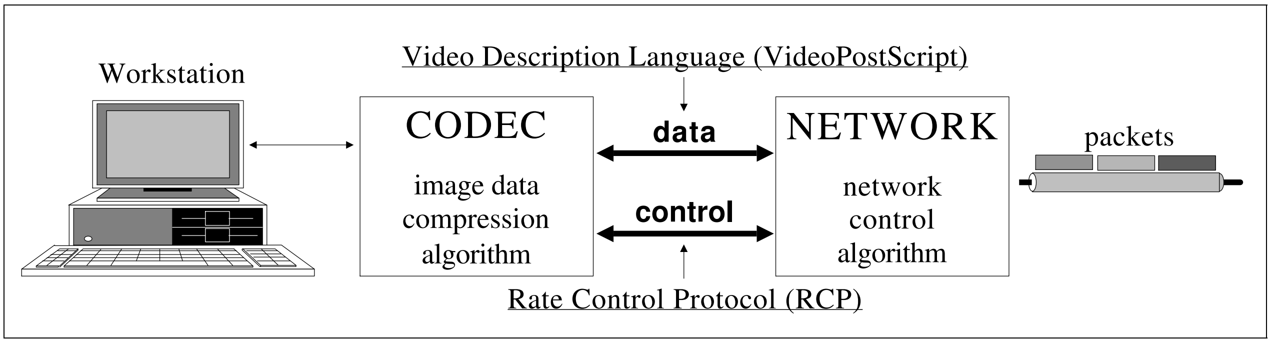

Fig. 1 : A feedback communication system provides information about network congestion. The rate-control protocol adapts coding parameters in real-time in order to avoid packet loss.

The network estimator informs the coding agent about the current status of the network by means of a rate-control protocol. The feedback initiates long-term changes in the coder output rate. Means and strategies for adapting coding parameters are discussed in section 3.1. In addition, in order to adapt the short-term rate of the sender to that of the receiver, we combine rate control with flow control, which is explained in section 3.2. The feasibility of rate and flow-control strategies for network congestion control has been investigated[15,19] and shown to be effective for packet loss reduction.

The approach proposed here does not necessarily invalidate the notion of constant quality. Instead, a multirate service attempts to reach the optimum image quality that is possible given the channel capacity[26]. Rate control cannot totally eliminate packet loss, and other techniques, as for example layered coding and error concealment, are needed in combination. Note that this approach does not assume or is restricted to any kind of channels: for networks with guaranteed performance rate control reduces to a selection of the quality of service; networks that offer different classes of service will lower the impact of rate control.

3. NETWORK-INTEGRATED VIDEO CODER

A feedback communications system can be implemented by integrating the network into the video coder design. This approach is in contrast to many schemes published so far, that disregard the interaction between the coder and the network. We attempt to achieve congestion control in two layers: at the network layer, flow control is employed in order to avoid short-term congestion. For long-term congestion avoidance, a rate-control scheme is implemented at the application layer: based on information about the current status of the network coding parameters are adapted accordingly. The structure introduced in the following subsections employs a video coding agent and a network estimator. The coding agent reacts to information gathered by the network estimator and adjusts the coder output rate accordingly. A practical implementation is finally described in section 3.3, which compares the network to a data buffer.

3.1 Video Coding Agent

In contrast to audio communication, video communication is in fact very well suited for variable data rate approaches. This follows from the high amount of compressibility (redundancy and irrelevancy) of digital video data. Redundancy occurs because of the high correlation of spatial/temporal adjacent picture elements. The digital image data also contains irrelevant information, which the human observer is not capable of perceiving, e.g. due to masking, or not interested in receiving. Besides, video may be presented at many different quality levels, even at the lowest level, providing still very valuable visual information. Due to the discrete temporal nature of video, in contrast to audio, video also allows playback at variable frame rates. As a last measure, video can always be frozen. Initial experiments with variable quality/variable frame rate have already been reported[25]. The results indicated that although the frame rate was varied, subjective evaluation was favorable as compared to constant frame rate and bit rate.

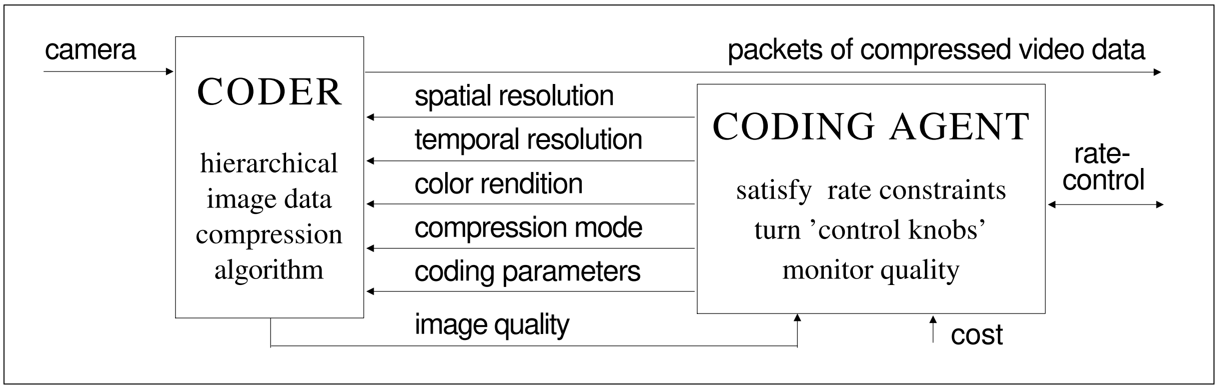

Fig. 2 : The coding agent orchestrates setting of coding parameters in order to satisfy rate constraints and ensure the best possible picture quality.

A multirate video service can be implemented by modifying one or many of the parameters depicted in **Fig. 2**. Most of the proposed image coding schemes vary only one or a very small number of the possible parameters, or do not allow for parameter changes during operation. In order to arrive at a good compromise between constant image quality and the effect of packet loss, it might however be necessary to increase the range of reachable data rates that can be adjusted through rate control[21]. An even stronger argument is the capability of achieving a multirate service: the same coding scheme can be used for different levels of quality, from videophone to broadcast quality. Even inside the same category, different quality levels are possible and may be selected by the user according, for instance, to cost constraints.

The following list shows the type of 'control-knobs' that are to be influenced by the rate-control information and that can be adjusted to achieve varying output rate od quality of service:

- **Spatial Resolution:** Digital images are sampled at a rate with respect to the highest spatial frequencies that are to be conserved. Pyramidal or subband coding schemes represent images at varying levels of spatial resolution through lowpass filtering and subsampling, This effectively reduces the number of pixels to be coded for a low resolution version of the image.

- **Temporal Resolution:** The same approach can be used to reduce temporal resolution. Again, the initial temporal sampling done by a camera for each frame already constitutes a reduction in the amount of data with respect to the capabilities of the human visual system. Further reduction is possible by temporal filtering and subsampling, even without artifacts, depending on the speed of motion in the scene. At the receiver site, frames are either simply repeated, linearly or motion-adaptively interpolated. As an extreme measure, the display may be frozen, changing from motion to still video mode.
- **Color Rendition:** Color information is effectively provided by enhancing the luminance by two color chrominance components, e.g. YUV or YIQ. Human visual perception is already exploited by spatial subsampling of the color components. The amount of color information can be reduced further by coarse quantization of the chrominance components, down to B/W-only transmission.
- **Compression Mode:** All the above measures can be combined with various compression techniques that exploit the high correlation of spatial/temporal adjacent picture elements and properties of the human visual system. Improved coding efficiency, i.e. higher data compression, is in general connected with higher computational complexity and is also more sensitive to packet loss; examples are intraframe vs. interframe coding.
- **Compression Parameters:** For each of the modes mentioned above, a multitude of parameters determine the coder output rate and the resulting image quality. Examples are the quantization parameters used in transform coding and the resolution of the motion vectors for motion-compensated prediction.

A given rate or a certain reduction in coder output rate can be achieved by a variety of different parameter settings. It is the expertise of a system designer together with subjective evaluation of the obtained image quality at the receiver that helps deciding on the optimum parameter selection for a given application. Our video coding agent implements a flexible coding scheme that offers selectable quality of service and can react upon rate-control commands by graceful degradation. It is the responsibility of this agent to control the parameters in a way that observes the data rate or cost constraints and at the same time yields the best picture for that rate. In that sense the video coding agent has to act like an expert on video coding and orchestrate the parameter setting accordingly.

3.2 Network Estimator

The purpose of the network estimator is to monitor the traffic on a packet-switching network and to estimate the bandwidth of the channel used by the coding algorithm, **Fig. 3**. The codec in turn will use this information to regulate its maximum output rate. The propagation delay of end-to-end feedback severely limits its use for video communication. Instead, the network estimator, based on its knowledge of the state of the network, can provide instant feedback to the coder. Any intelligence in the network itself can be used to enhance the accuracy of the state estimation process.

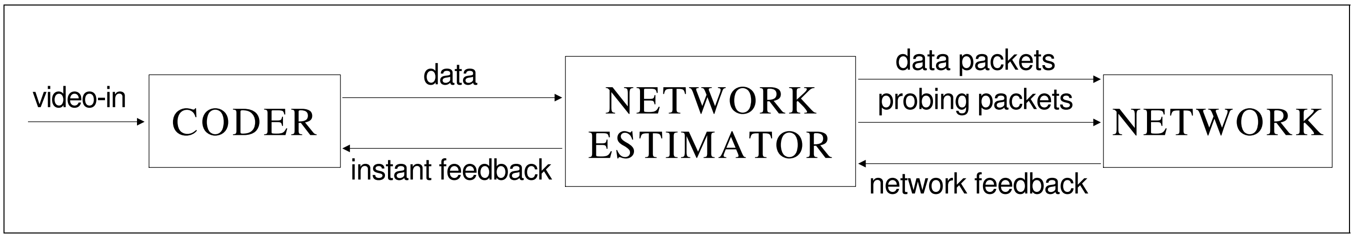

Fig. 3 : The network estimator monitors the network data flow, keeps track of packet loss, and provides instant feedback to the coder.

In packet-switching networks that offer no admission control, i.e. network that do not impose any restriction on input traffic, a temporarily excessive offered load will cause congestion at the bottleneck links. Congestion results in packet loss due to lack of buffer space. A network that carries real-time information must limit the number of channels that are open, otherwise, even if congestion does not develop, the throughput on each of the existing channels is reduced below a level at which the channels can provide acceptable service. In addition, no control on the number of open channels may also result in delays that are intolerable by real-time applications. Conceivably, an integrated network will have several classes of services. Admission control may only be applied to those classes, which are given a higher priority.

Although the most common cause of network congestion is input overload, congestion can also develop in cases, in which the total input traffic is below the network capacity. This can occur for instance because of improper routing that conglomerates most traffic through a few low-capacity links, or because traffic clusters inside the network producing packet trains that travel together. Controlling congestion through flow control has long been been a fundamental research issue in network design[9]. Recent new approaches include work on protocol timers[14], feedback from routers that results in a protocol-header bit being set when the queue length on a router exceeds a maximum predetermined value[27], new queuing disciplines such as stop-and-go[11], and feedback on the utilization of network bottlenecks obtained from the acknowledgements received in response to two packets sent as a pair[19].

In addition, in order to address flow and congestion control, but mainly to provide new mechanisms for the delivery of real-time streams, researchers have begun exploring the notion of networks that preserve or bound packet jitters[6,5,24]. In these networks, routers or ATM switching nodes run algorithms to ensure that packets produced by selected real-time, rate-controlled sources are delivered within the required bounds without altering significantly the spacing between subsequent packets. Jitter-control algorithms typically rely on globally synchronized clocks. Whereas these approaches look worth investigating, many feel that the complexities of the proposed designs will make the network difficult and expensive to manage. Nevertheless, we do not discount that in the future there may be specific network support for continuous media.

Our approach addresses the case of transmission of video streams over today's networks and future networks that will not provide specific support for real-time transmission. However, if the network is operated with high utilization, admission control becomes necessary, in order to guarantee at least a minimum grade of service for real-time channels. We cannot use reliable protocols in connection with real-time, interactive video because of the stringent delay requirement of video frame delivery. But by channel flow-monitoring (e.g., if the network can provide senders with information about the throughput and delay performance of its channels) we can exploit the adaptability of motion-video compression algorithms in order to provide better network resource utilization and reduce (and possibly eliminate) packet loss on these channels. Flow monitoring will be easier on smart networks, networks that maintain per channel state information, or that perform per channel queuing. But it can be performed also on today's dumb networks, such as a mesh of local-area networks connected by IP routers that handle packets through a single queue in first-in-first-out fashion.

Flow-monitoring techniques can be divided into two classes: techniques that do not require network cooperation and rely exclusively on end-to-end feedback signals, and techniques in which the network itself plays a role.

- **End-to-end techniques:** Flow-monitoring requires the cooperation of the receiving station. The cooperation can be active if the receiving station provides the sender periodically with information that the sender can use to estimate the channel bandwidth. It can be passive if it is the sender who requests some feedback from the receiver from time to time.
- **Network-agent techniques:** These techniques involve the direct participation of network agents, whose task is to monitor the flow and provide feedback to the senders, who should reduce their channel rates if they are to avoid packet loss. The feedback may take the form of "source quench" messages, or can be communicated by means of header bits that the network agent can toggle. Because of its nature, this technique can only be active.

Flow-monitoring can also be divided into two classes according to the mechanisms used to implement it.

- **Window-based mechanisms:** In this class, the sender rate is regulated by a window (measured in packets or bytes), whose size is chosen at connection-establishment time. The window is closed as packets are sent; it is opened as acknowledgements are received. If the window decreases to zero, the source will stop sending packets. A window mechanism requires the active participation of the receiver. If the product of the round-trip channel delay and the maximum channel bandwidth is greater than the window size, then the window size determines the channel throughput. Thus, in high-speed networks, a window-renegotiation scheme may allow communicating clients to increase their throughput. Conversely, if the window becomes too large because of congestion that develops in the network, window-renegotiation may decrease the chances of packet loss.
- **Rate-based mechanisms:** In this class, the sender and the receiver agree on a packet rate, which is maintained by the sender without the need for the receiver to acknowledge each packet. Mechanisms to change the flow are required if communication clients want to protect themselves from packet loss caused by channel overload, or if they need to communicate at a higher data rate, when the network allows it.

Flow-monitoring techniques and mechanisms can be combined to produce a number of schemes by which communication clients can obtain network status information. In a window scheme based on end-to-end acknowledgements – an active technique – the sender renegotiates the window size based on estimated round-trip times obtained with the acknowledgements. (Notice that depending on the type of network, window renegotiation may simply be a completely local operation.) A window scheme can be employed with (passive) periodic packet probes, whose network utilization estimates are used to change the window size. The length of intervals between successive probe transmissions will have to be studied. In a special form of this method, a pair of packets are sent back-to-back. The time interval between the receipt of the corresponding acknowledgements is proportional to the load in the network's bottleneck. Finally, a window scheme can be used in association with network-agent techniques. In this scheme window renegotiation occurs under direct response from network provided signals.

Packet rate flow-monitoring schemes can be based on end-to-end information from the receiver, which can monitor the received packet jitter and signal the sender to decrease its rate. Packet rate schemes can be combined with passive probes in order to obtain the information necessary to change the rate. They also work with network-agent techniques under a setup in which the network could inform the transmitter that it is time to reduce the flow by sending a *'source quench'* message.

In response to flow-monitoring notifications, whether they are provided by the receiver or by the network itself, the compression algorithm can change its output data rate accordingly. The feedback signals are available at the transmitter with some delay after the time they are originated. The compression algorithm will also add some delay, as in general it will take a finite amount of time to switch to a new regime. The effects of these delays on the stability of the controlling techniques will have to be studied in detail.

In the experimental system we have designed, we provide flow monitoring at two levels: the OSI Transport and Presentation layers. At the Transport layer the main issues are reducing the burstiness of the sender stream, limiting the number of packets travelling as a train, and adapting the short term rate of the sender to that of the receiver. At the Presentation layer the main goal is to modulate the rate produced by the coding algorithm so that some measure of network congestion is reduced. The flow-monitoring techniques selected in the experimental system will be described in section 4.2.

3.3 Network as Buffer

Conventional video coding schemes for constant data rate flatten the variable coder output rate, which is caused by fluctuation in the information content of the video stream, by means of a buffer operating as a leaky bucket, **Fig. 4**. The amount of data in the buffer is monitored by a controller, which adapts one or more coding parameters in order to prevent the buffer from overflowing. Buffers have also been used in connection with packet video and variable data rate[28]. In general, any buffer increases the delay of the processing chain from sender to receiver, which is especially critical for interactive services.

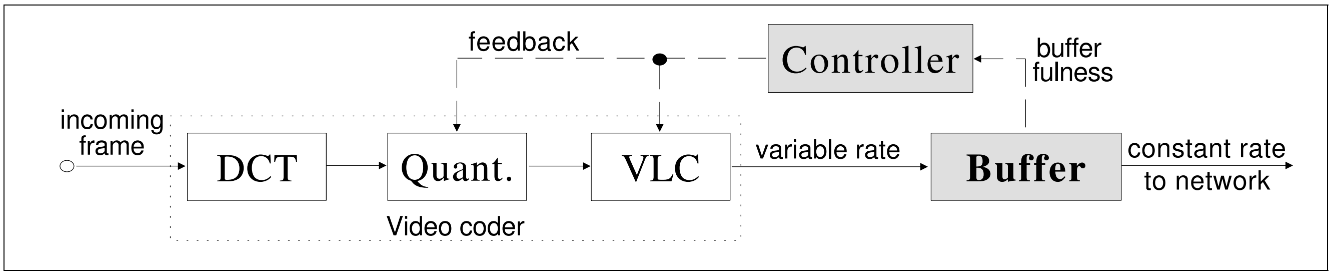

Fig. 4 : Conventional buffer implementation: buffer flattens output rate; buffer fullness controls coding parameters.

With packet-switching networks the output rate may vary and no buffer is needed[21]. We therefore replace the buffer by the network as shown in **Fig. 5**. Now, packet loss inside the network is equivalent to buffer overflow in the previous case. As we have mentioned, the vast majority of packet losses inside the network is caused by congestion. In this model, the controller has to estimate the network performance and to provide a feedback signal that indicates the state of the network and congestion situations. This feedback signal is used as before to adapt the coding parameters in real-time in a way that avoids congestion and associated packet loss. Note that in contrast to the leaky bucket buffer case, the network is a non-linear, variable-capacity, and variable-rate leaky bucket.

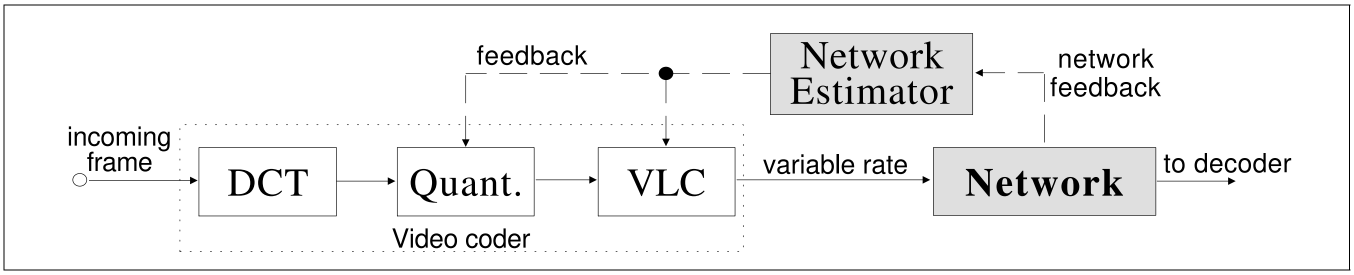

Fig. 5 : Network as a buffer: packet loss equals to buffer overflow; coding parameters are adapted according to the flow-control commands issued by the network estimator.

The use of an input buffer before the network has been shown to be impractical because of the long duration of data peaks in video scenes and therefore large buffer requirements[2]. Employing the network itself as buffer not only guarantees smallest possible delays, it also allows to use the same tools and approaches that are conventionally used for buffer design to be used in this case.

4. EXPERIMENTS

Several problems make the implementation of the proposed feedback system particular hard: network and application do interact; if the output rate increases due to higher information content in the scene, the likelihood of network congestion and therefore packet loss also increases. This and other complex interactions are not accounted for in network models. Most researchers designing coding algorithms for packet video use fixed packet loss rates and simulate the network by randomly discarding packets[18]. A fixed loss rate is however very unlikely, as packet loss usually occurs in bursts. Therefore, in order to get realistic results, we decided to use actual packet transmission on a testbed consisting of local area networks connected by gateways.

4.1 Real-time Video Coding

Because of our approach we necessitate image coding algorithms capable of running in real-time. Since we had no dedicated coding hardware, we had to implement a rather simple algorithm in software to provide adequate compression and means of adaptivity, and that would run in real-time on a Sun SPARCstation 2. We selected a typical videophone image format (CIF) with 352 pixels per line, 288 lines, 8 bit per pixel black and white, and no interlace. A frame rate of 30 frames per second (fps) would give only 33 ms processing time, therefore a temporal subsampling by a factor of 2 was used, i.e. 15 fps, corresponding to a source bit rate of 12 Mbits/s.

The main factor in obtaining a substantial image coding gain is the strong correlation of spatial and temporal adjacent pixels. Considering intraframe coding only, spatial correlation can be exploited if homogeneous regions of the image are represented by a single gray value. A simple solution is provided by a quadtree coding scheme[29]. Because it is highly unlikely that very large regions in the image are homogeneous, we first subdivide the whole image into blocks of size 16×16 pixels. Based on a splitting criterion, e.g. variance of the pixels, each block is then recursively subdivided into 4 square subblocks of equal size. The splitting process is controlled by a threshold: if the variance of the gray values of a block is below the threshold, the splitting stops and the current block is represented by the mean luminance value. The smallest block size of 2×2 pixels is not further subdivided.

The algorithm, as described, leads to a natural recursive implementation. Real-time speed was reached through bottom-up processing, which avoids multiple calculation of the splitting criterion and the overhead connected with recursive code. Additionally, a linear error measure was used, which can be implemented without floating point multiplications or divisions: if the mean value of each of the 4 subblocks does not differ from the mean value of the combined block by more than the threshold, the 4 subblocks are merged and represented by the mean value of the combined block. For the given image size, the algorithm executes in approximately 35 ms per frame. Depending on the threshold and the image content a compression factor between 7 and 30 is reached, maintaining good image quality. Decompression times at the receiver are slightly higher, approximately 38 ms, because of longer memory write cycles.

Robustness against packet loss is one major factor favoring layered coding schemes. Subband coding is often cited in this context, because lost data in the higher frequency bands causes only slight impairments, e.g. lost detail. The baseband in subband coding, however, is very sensitive to packet loss because it carries most of the information. Solutions to this problem circle around schemes like guaranteed transmission of the baseband data, high priority of packets containing baseband data, and replacement of missing areas by the previous frame. The first two solutions call for special network protocols, while the last solution yields unacceptable artifacts in the presence of motion. A better solution are error concealment techniques[31]; in particular techniques that exploit the correlation between coding layers, as reported by Wang and Ramamoorthy[32], seem very promising and question the priority/guarantee approach proposed by many researchers.

In order to make the simple coding scheme proposed above robust against packet loss and enable error concealment, under the constraint that this should run in real-time in software, a simple interlace scheme is used[3]: as in conventional television, each frame is split into two fields, using odd and even lines, respectively. Each field is coded separately. Missing lines, caused by packet loss, are interpolated from the adjacent lines of the other field. The loss of both fields at the same location is very unlikely by virtue of the congestion avoidance principles employed.

In summary, the proposed scheme performs extremely well, given its simplicity. Even without explicit layering, robustness against packet loss was achieved and together with error concealment packet loss has minimal impact on picture quality. Although computational constraints do not allow to implement more complex layered coding scheme without dedicated hardware, the obtained results are useful and serve as a worst-case estimate: if video quality is acceptable despite the simple coding algorithm, more complex schemes will perform even better, assuming equal rate-adaptation capabilities.

4.2 Transport Layer: Flow Control

The Transport protocol used in the experimental system is a window flow-controlled protocol. We use the term *'packets'* for the individual protocol messages; a compressed video frame requires the transmission of several packets. Each packet is marked with a sequence number, and contains a timestamp field, filled by the sender with the transmission time. The use of the timestamp field will be described later. The receiver acknowledges each delivered packet by sending back a message with the same sequence number and timestamp. The window, maintained by the sender, is expressed in terms of the maximum number of packets that can be outstanding at any given time. After transmitting a packet, the sender decreases the window by one. If the window is zero, no packet is transmitted. Upon receiving an acknowledgment the sender opens the window by one.

Packets and acknowledgements can be lost; therefore, if no measures were taken, the window would progressively shrink to zero. Since our local-area networks do not reorder messages, we simply compare the sequence number in each acknowledgement with the expected sequence number, that is the sequence number of the previous acknowledgement plus one. We can detect if a number of messages or their acknowledgements are lost by comparing the returned sequence number with the expected sequence number; we then open the window accordingly. In networks that reorder messages, the protocol layer would have to keep a list of transmitted packets with timestamps, and timeout those that are not acknowledged within a bound that depends on the round-trip time.

Under severe congestion conditions all packets or acknowledgements in a window can be lost. The protocol implementation that we have so far described would deadlock in such circumstance as the sender would be waiting for acknowledgements that never come. We solve this problem by using the timestamps to estimate the round trip time and setting a timer proportional to it when the window becomes zero. If the timer expires before receiving an acknowledgment we forcibly open the window and send a *'probe'* packet. Since timers are based on round-trip time, with high degree of confidence, expiration of a timer indicates packet loss.

The main purpose of the flow-controlled protocol is to avoid that a fast sender flood the receiving machines or gateways along the way. In other words, flow control is a method for reducing the burstiness of the sending stream and to force the sender to adapt to the speed of the receiver. The same goal could be achieved by a scheme in which the sender would transmit a frame's packets in equally spaced intervals during the frame time. There are at least two reasons for preferring a window-control scheme at the Transport layer. First, it has a finer degree of granularity and can adapt to momentary network congestion more quickly and, second, it does not require the sender to handle timers, resulting in software simpler to write and maintain. As we will discuss next, we also use rate control at the Presentation layer.

In order to couple protocol storage management and visualization processing, we modified the communication subsystem so that it could receive packets asynchronously. Assuming that a packet arrives while the receiver is decompressing a previous packet, an interrupt is generated (though the actual delivery mechanism is a UNIX signal), and the packet placed into a work queue. An important advantage of this mechanism is that acknowledgements can now be sent according to more sophisticated policies. For instance, with this scheme, it is conceivable that the receiver, instead of the sender, maintain the protocol window.

4.3 Presentation Layer: Rate Control

The Presentation layer protocol is intended to modulate the rate produced by the coder in order to adapt to the varying maximum bandwidth available over the transmission channel. Whereas the Transport-layer flow-control mechanism operates within time intervals comparable to the round-trip transmission time of each packet, rate control at the Presentation layer is meant to react to longer-term fluctuations in the channel bandwidth. Depending on the structure of the network routers, lower effective bandwidth results in longer delays and may result in higher packet losses.

At the Transport layer the rate is adjusted by requesting a variation in the compression ratio according to some control metric that estimates the performance of the network. Depending on the type of network, several different metrics can be used. One such metric regulates the rate according to the number of packet losses in the unit time. This is particularly useful when gateways have limited memory for network buffers and high input traffic translates into buffer overflows. Packet loss are easily detected by our network agent, since acknowledgements carry sequence numbers.

A second type of metric, one that reacts to long network delays, is also an important policy. This policy is more effective on networks that have routers with large buffer areas; in these networks congestion increases the delays to a level at which they can be intolerable to video applications, long before packets begin being discarded. Delays affect video delivery in two major ways: first, delayed frames produce jerky motion and reduce image quality. Second, even though one could eliminate delay variations by adding substantial buffer at the receiver, long delays make it impossible to use these communication channels for interactive video applications. In order to detect excessive delays, our protocol uses the timestamp field in the protocol header. Since the timestamp is copied back in the acknowledgements, time calculations are computed at the sender and we do not need synchronized clocks.

Because of our decisions about the format of the frames and the real-time compression scheme, we produce a compressed data rate in the order of 10 kbytes/frame. As video data is encoded by the compression algorithm, the system checks at block boundaries for the amount of data produced. Whenever the accumulated data reaches the packet capacity, or at the end of each frame, a packet is assembled and shipped to the receiver. In order to reconstruct the frame regardless of losses, the Presentation layer protocol at the receiver provides the frame number and the starting block number for each packet.

At the time of this writing, we have not fully estimated the effectiveness of our estimators of network congestion. At the conference we will present results and lessons learned from experience with the system.

5. CONCLUSIONS

The described approach aims at controlling in real-time the amount of data produced in an attempt to avoid packet loss and delays too long to support interactive packet video. Investigations have shown that almost all packet losses are caused by network congestion and internal buffer overflow. Control-theoretic analysis has also proven that congestion avoidance is feasible. In any case packet loss cannot be totally avoided, but due to the reduced number of packet losses, layered coding and error concealment can be more efficient.

Our experiments also gave insight from a different viewpoint: the available resources in terms of average network capacity and processing power have to be balanced against the computational complexity of the coding algorithm, the protocol, and the output rate of the coder. Data compression algorithms help transform network load into computational load; experiments without data compression proved the network and the protocols to be the bottlenecks, limiting the frame rate to below 6 frames per second. Complex compression schemes yielding higher compression factors would result in even slower frame rates for the given hardware. Only the combination of the compression algorithm introduced above together with Ethernet networks would deliver the best performance of 15 fps for the given hardware and image size.

6. REFERENCES

1. R.C. Brainard, J.H. Othmer, "Television compression algorithm and transmission over packet networks," SPIE, Advances in Image Coding, vol. 1001, pp. 973-978, Cambridge, MA, Nov. 1988.
2. H.S. Chin, J.W. Goodge, et al., "Statistics of video signals for viewphone-type pictures," IEEE Journal on Selected Areas in Communications, vol. 7, no. 5, pp. 826-832, June 1989.
3. J.C. Darragh, R.L. Baker, "Fixed distortion subband coding of images for packet-switched networks," IEEE Journal on Selected Areas in Communications, vol. 7, no. 5, pp. 789-800, June 1989.
4. G. Demos, "Digital communications network interactions with real-time imagery data," DemoGraFX, private communication, Nov. 1990.
5. J. Escobar, D. Deutsch, and C. Partridge, "A multi-service flow synchronization protocol," BBN Technical Report, March 1991.
6. D. Ferrari and D. C. Verma, "A scheme for real-time channel establishment in wide-area networks," IEEE Journal on Selected Area in Communications, vol. SAC-8, n.3, pp. 368-379, April 1990.
7. M.W. Garrett, M. Vetterli, "Joint source/channel coding of statistically multiplexed real-time services on packet networks," Submitted to IEEE Journal Selected Areas in Communications, Dec. 1990.
8. M.W. Garrett, M. Vetterli, "Congestion control strategies for packet video," 4th International Workshop on Packet Video, pp. 1-6, Kyoto, Japan, March 1991.
9. M. Gerla and L. Kleinrock, "Flow control: a comparative survey," IEEE Transactions on Communications, 1980, pp. 553-574.
10. M. Ghanbari, "Two-layer coding of video signals for VBR networks," IEEE Journal on Selected Areas in Communications, vol. 7, no. 5, pp. 771-781, June 1989.
11. S.J. Golestani, "Congestion-free transmission of real-time traffic in packet networks," Proceedings of INFOCOMM, 1990, pp. 527-536.
12. T.R. Hsing, "Video compression techniques: Networking aspects," Intl. Symp. on Circuits and Systems, pp.223-226, Espoo, Finnland, June 1988.
13. Image Communication, Special Issue on "Video Coding for ATM Networks, " Part I+II, vol. 3, no. 2-4, June and September 1991.
14. V. Jacobson, "Congestion avoidance and control," ACM Proceedings SIGCOMM '88, pp. 314-329, August 1988.
15. R. Jain, "A delay-based approach for congestion avoidance in interconnected heterogeneous computer networks," Computer Communications Review, vol. 19, no. 5, pp. 56-71, Oct. 89.
16. R. Jain, "Myths about congestion management in high-speed network," Digital Equipment Corporation, Technical Report no. 726, Oct. 1990.
17. G. Karlsson, M. Vetterli, "Subband coding of video for packet networks," Optical Engineering, vol. 27, no. 7, pp. 574-586, July 1988.
18. G. Karlsson, M. Vetterli, "Packet video and its integration into the network architecture," IEEE Journal on Selected Areas in Communication, vol. 7, no. 5, pp. 739-751, June 1989.
19. S. Keshav, "A control-theoretic approach to flow control," International Computer Science Institute, Technical Report, no. 91-015, March 1991.
20. F. Kishino, K. Manabe, et al., "Variable bit rate coding of video signals for ATM networks," IEEE Journal on Selected Areas in Communications, vol. 7, no. 5, pp. 801-806, June 1989.
21. S.H. Lee, L.T. Wu, " Variable rate video transport in broadband packet networks," SPIE Proceedings, Visual Communications and Image Processing, vol. 1001, pp. 954-964, Cambridge, MA, Nov. 1988.
22. D.-S. Lee, K.-H. Tzou, et al., "Control analysis of video packet loss in ATM networks," SPIE, Visual Communications and Image Processing, vol. 1360, pp. 1232-1243, 1990.
23. B. Maglaris, D. Anastassiou, et al., "Performance models of statistical multiplexing in packet video communications," IEEE Transactions on Communications, vol. 36, no. 7, pp. 834-844, July 1988.
24. M. Moran, B. Wolfinger, "A continuous media data transport service for real-time communication in high speed networks," Workshop on Continuous Time Media, Carnegie Mellon University, June 1991.
25. M. Nomura, T. Fujii, et al., "Basic characteristics of variable rate video coding in ATM environment," IEEE Journal on Selected Areas in Communications, vol. 7, no. 5, pp. 752-760, June 1989.
26. A. Puri, R. Aravind, "An interframe coding scheme for packet video," SPIE, Visual Communications and Image Processing IV, vol. 1199, pp. 1610-1619, 1989.

27. K. K. Ramakrishnan and R. Jain, "A binary feedback scheme for congestion avoidance in computer networks," ACM TOCS, vol. 8, no. 2, pp. 158-181, May 1990.
28. L. Salgado, A. Sanz, "Pyramidal encoding for packet video transmission," SPIE, Visual Communications and Image Processing, vol. 1360, pp. 1822-1833, 1990.
29. S.L. Tanimoto, T. Pavlidis, "A hierarchical data structure for picture processing," Computer Graphics and Image Processing, no. 4, pp. 104-119, 1989.
30. W. Verbiest, L. Pinnoo, "A variable bit rate video codec for asynchronous transfer mode networks," IEEE Journal on Selected Areas in Communications, vol. 7, no. 5, pp. 761-770, June 1989.
31. M. Wada, "Selective recovery of video packet loss using error concealment," IEEE Journal on Selected Areas in Communications, vol. 7, no. 5, pp. 807-814, June 1989.
32. Y. Wang, V. Ramamoorthy, "Image reconstruction by partial subband images and its application in packet video transmission, " Special Issue on "Video Coding for ATM Networks, " Part I, Signal Processing: Image Communication, vol. 3, no. 2-3, pp. 197-230, June 1991.