TELECOMMUNICATION TRAFFIC

The telephone was the invention of Alexander Graham Bell (born Scotland, March 3, 1847, died August 2, 1922), who discovered the principles that make it possible to transmit speech by electrical means. Bell was granted a patent for the invention of the telephone on March 7, 1876, and the telephone network was born on March 10, 1876, when Bell transmitted the first telephone message to his assistant: "Mr. Watson, come here, I want you." The first long-distance call was placed by Bell in 1877 over a distance of 29 km between Salem, Massachusetts and Boston. On July 9, 1877, Bell created the Bell Telephone Company, which was reorganized in 1885 as the American Telephone and Telegraph Company (AT&T).

Today, there are more than 400 million telephones worldwide. The telephone network is by far the largest and most sophisticated network in the world. In contrast to the telephone network, computer networks have a relatively short history. In the beginning, computers were expensive, occupied large volumes, and consumed vast amounts of power, thus making the idea of a computer network impractical. With the invention of the transistor in 1948 by Walter H. Brattain, John Bardeen, and William Shockley at Bell Laboratories, and especially the integrated circuit (IC or chip) by Jack S. Kilby in 1958 and Robert Noyce in 1959, computers became smaller and affordable. The first IC-based computer was produced by Digital Equipment Corporation in 1963. Fiber-optics communications was invented by Robert Maurer in 1968. The Advanced Research Projects Agency (ARPA) of the US Department of Defense started a computer network called ARPANET in 1969 that would link universities, government, and businesses. ARPANET was officially retired in 1990 and thus was born the Internet. Today, although the Internet is the largest public computer network in the world and is growing at an unprecedented rate, the largest telecommunications network in the world is the telephone network.

The invention of the microprocessor by Marcian E. Hoff, Jr. in 1971 further enhanced computers, and computer networks became generally available shortly thereafter. A major factor in the development of computer networks was the invention of the Ethernet Local Area Network (LAN) by Robert N. Metcalfe in 1973 (based on his ALOHA system for radio communications). The first Ethernet LAN adapter was shipped by 3Com (a company founded by Metcalfe) on September 29, 1982. Computer networks have since evolved very quickly, and today a computer with access to the Internet is becoming a common household appliance.

The telephone network was originally designed to carry voice traffic. Inventions such as the facsimile (fax) in 1921 (Western Union) which was standardized in 1966, the modem in 1956 (Bell Laboratories), and the videophone (Bell Laboratories) in 1964 introduced new traffic types which the telephone network was not designed to carry. For example, the short holding times of facsimile transmissions, as well as the long holding times and automatic redial of modems, are very different from the holding times and redial patterns of normal voice calls. Internet access by modem is a major source of routing and congestion problems for the telephone network.

Today, traffic analysis methodologies for the telephone network with voice calls are well-developed, but the networks are being reengineered to handle the new types of traffic. At the same time, computer networks are evolving at a very fast

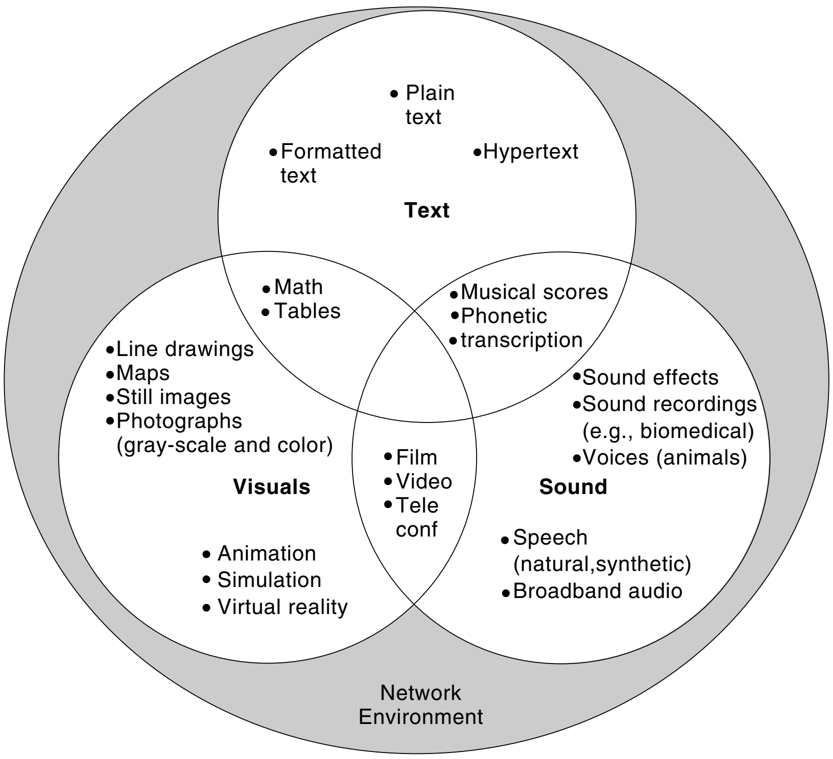

Figure 1. Diagram illustrating the diverse material over nonswitched and packet networks.

pace and are being designed to carry voice, data, and video traffic. New analysis methodologies are being developed as a result of the evolution from voice networks to more general purpose networks.

Multimedia Traffic

Multimedia traffic refers to the transmission of data representing diverse material over telecommunications networks. Figure 1 shows the diversity of the traffic classified into three groups: (i) text, (ii) visuals, and (iii) sound.

In addition to traditional unformatted plain text, symbolic material may also include formatted text with numerous control characters, mathematical expressions, tabular data, phonetic transcription of speech, music scores, and other symbolic representations such as hypertext for nonlinear browsing. The visual material may include line drawings, maps, and gray scale or color still images and photographs, as well as animation, simulation, virtual reality, video, and teleconferencing. Sound content may include telephone- and broadcast-quality speech to represent natural voices (human and animal), wideband audio for music reproduction, and recordings of sounds such as sonograms or other biomedical signals. Representations of the remaining senses such as touch and smell are not excluded from multimedia. In the present networking practice, the range of multimedia is limited to sound, video, and text.

The continuing success of multimedia depends on the solution of at least four difficult technical problems: (i) how to encode the diverse material ranging from text to video and audio, (ii) how to store and transmit the electronic representations efficiently, (iii) how to distribute the electronic material to end users, and (iv) how to search nontextual material such as images and sound. We shall address the first two problems because such multimedia representations lead to large files, and efficient storage and transmission requires compression using source encoding. A data source coder minimizes the bit rate of the data. On the other hand, a perceptual source coder minimizes the bit rate of the input signal while preserving its quality. According to the Shouten diagram, this corresponds to the removal of both redundancy and irrelevancy. This orthogonal principle of both redundancy reduction and irrelevancy removal does not correspond to the maximization of signal-to-noise ratio (i.e., the minimization of mean-squared error) and is central to the second-generation of codecs (coder-decoder).

A source coder is often followed by (a) a channel coder which adds redundancy for error protection and (b) a modem which maximizes the bit rate that can be supported in a given channel or storage medium, without causing an unacceptable bit error probability. Ideally, the entire process of source coding, channel coding, and modulation should be considered jointly to achieve the most resilient bit-stream for transmission.

Specifying the characteristics of traffic in multimedia environments is more difficult than in circuit-switched systems in which a fixed bandwidth channel is held for the duration of a call and only arrival times of calls and their durations are required. Packet-switched systems carrying multimedia have variable bit rates with bandwidth on demand. This calls for knowledge not only of the statistics of the sources, but also of the rules for assembling the packets.

DIGITAL DATA

Digital data are defined as arbitrary finite-state representations of source information. Signals, on the other hand, are constrained finite-state representations of temporal or spatial entities, including the important subclasses of physical analog signals such as speech, audio, image, and video. The object of data or signals (source) coding or compression is a compact digital representation of the source information. Often, the receiver of data is a computer, while the receiver of signals is a human. There are important differences between data and signal compression because the source signal is non-Gaussian and nonstationary and, in general, has a complex and intractable power spectrum, with the added complication that the human receiver does not employ a mean-squared-error criterion to judge the quality of the reconstructed signal (1). Instead, humans use a perceptual distortion criterion to measure source entropy. This leads to two approaches to source compression: lossless and lossy.

Exact and Inexact Models of Data and Signals

We distinguish between critical (precise) data and noncritical (imprecise) data. For critical data such as computer code and scientific data, both their representations—archival and transmission—must be exact, without losing a single bit. Techniques dealing with such data are called lossless. For noncritical data such as video, images, speech, biological signals, and casual text, their representations may be imprecise, as long as the perceptual quality is maintained. Techniques that allow minor details to be ignored in noncritical data are called lossy. It should be noted that the archiving or transmission of either data representation should be exact.

Models and Attributes of Source Compression

Source compression refers to the removal of redundancy from a source (data or signals) by a proper mapping into codewords, carrying either all or only the essential part of the necessary information about the source so that decompression is possible either without loss of information or with partial loss of information, respectively. Redundancy is a probabilistic measure (entropy) of the deviation of probabilities of the occurrence of individual symbols in the source with respect to the equal symbol probabilities. If the character probabilities are all equal, the entropy becomes maximal, and there is no redundancy in the source alphabet, implying that a random source cannot be compressed. The objective of the lossless compression techniques is to remove as much redundancy from the source as possible. This approach cannot produce large source compression. The quality of code is determined by the difference between the code entropy and the source entropy; if both are equal, then the code is called perfect in the information-theoretic sense. For example, Huffman and Shannon–Fano codes are close to perfect in that sense (2). Clearly, no statistical code will be able to have entropy smaller than the source entropy.

On the other hand, lossy compression achieves higher compression ratios by removing some information from the source. The critical issue is what constitutes essential information and what information is irrelevant. Irrelevancy is defined as the difference between the critical and noncritical information in the source. Unlike probabilistic measure defining redundancy, irrelevancy is determined by possibilistic, belief, or fuzzy perceptual measures.

Compression and decompression processes may have a number of attributes. A compression is reversible if the source data can be reconstructed from the compressed codewords. The compression is noiseless when no information is added into the decompressed data, thus making it the exact replica of the source. It is also lossless if no information is removed from the recovered data. For example, the Huffman, Lempel–Ziv–Welch (LZW), and WNC (3) techniques are noiseless and lossless. In contrast, nonreversible (or lossy) mapping (data compaction or abstraction) removes redundancy using approximate methods, and the exact reconstruction of the source is not possible. For example, speech compressed using the linear predictive coding (LPC) or adaptive differential pulse code modulation (ADPCM) algorithms cannot be reconstructed exactly.

Compression is called transparent when it is done without interaction with a computer programmer. Compression that is not transparent is also called interactive. Compression may be either statistical (e.g., Huffman) or nonstatistical (e.g., LZW). In statistical techniques, symbol usage statistics or data types must be provided in advance, based on either an average or local analysis of the actual data. Since the statistics gathering process requires a single pass and the compression another, these techniques are also called two-pass techniques. In contrast, nonstatistical techniques employ ad hoc rules designed to compress data with some success. Techniques may produce codewords that are either of fixed length (LZW) or of variable length (Huffman), with the former giving higher compression ratios.

Statistical and nonstatistical techniques may also be classified as adaptive or nonadaptive. An adaptive (or dynamic) compression does not require advance knowledge, and it constructs the necessary translation tables (e.g., LZW) or data statistics (e.g., dynamic Huffman and WNC) based exclusively on the incoming source stream. They are also called one-pass techniques. Nonadaptive (or static) techniques generate codewords, without affecting the original translation rules or data statistics. Depending on the method of outputting the codewords, a compression technique may be classified as stream oriented or block oriented. A stream-oriented technique outputs a codeword as soon as it is available, while the block-oriented technique must wait until the compression of a block is completed. For example, arithmetic coding is a block-oriented technique, with recent improvements that include incremental operation whereby the entire block is broken into smaller ones that are output as soon as they are completed (e.g., WNC and our implementation). Compression may also be regenerative or nonregenerative. In nonregenerative techniques, the translation table must be transmitted to the receiver, or else decompression is not possible. Regenerative methods do not require the transmission of translation tables, because they are capable of reconstructing the table from the codewords. If the compression and decompression phases take the same effort (time and real estate), then it is called symmetric. Clearly, compression methods that are reversible, noiseless, lossless, transparent, adaptive, regenerative, and symmetric seem to be the most desirable.

Source compression can be done in either hardware, software, firmware, or any combination of them. Software solutions may be applicable for slow data streams (megabits per second, Mbps), while modern parallel pipelined hardware solutions may provide speeds of hundreds of Mbps.

Performance of Source Compression

Experimental results (4) show that statistical variable-length Huffman technique compresses text by a ratio of 20 : 1.

There are many other techniques capable of compressing and decompressing data efficiently (5–7).

Modeling Digital Data Sources

Since digital data sources vary from very slow (input from a keyboard) to very fast (file transfer), their detailed characteristics must be established on an individual basis. Consequently, packetization schemes also vary greatly. Despite the diversity, specifying data source characteristics is simple and involves the Poisson process because much averaging takes place due to a very small ratio of the average bit rate of the individual data sources in a network to the bit rate of the multiplex links.

DIGITAL SPEECH

The quest to compress speech has been one of the major research endeavors for the last 50 years, and it still remains difficult because of the inherent nonstationary nature and variability of speech (8). Speech can be compressed with respect to its dynamic range and/or spectrum. Dynamic range reduction is used in telephones with the logarithmic A-law (Europe) or B-law (North America) capable of reducing the linear range of 12 bits (72 dB) to a nonlinear range of 8 bits only, thus compressing the bit rate by 1.5 : 1; the uncompressed rate is 8000 samples per second, or 8 ksps, times 12 bits per sample, resulting in 96,000 bits/s, or 96 kbps, while the compressed rate is 8 ksps $\times$ 8 b = 64 kbps. This constitutes the pulse code modulation (PCM) narrowband speech (300 Hz to 3300 Hz) coding standard (ITU-T G.711). A wideband speech (50 Hz to 7000 Hz) improves intelligibility and naturalness of speech while a band-limited speech sounds metallic. The 64 kbps ITU-T G.722 standard for wideband speech is the reference standard. Terrestrial or satellite-based digital sound broadcasting (DSB) with a sampling rate of 48 ksps uses the MPEG Layer II audio coding at 128 kbps.

Speech compression can be achieved using either scalar quantization (either without memory, or with memory such as predictive quantization) or vector quantization (also with or without memory). This coding can be achieved using four methods: (i) parametric (or source), (ii) waveform, (iii) filterbank, and (iv) transform coding. The second-generation coding schemes include psychoacoustics and masking considerations (9).

Speech Compression

From the multimedia point of view, one of the most interesting compression schemes is the code-excited linear predictive (CELP) coding which belongs to the parametric coding scheme, as well as the wavelet coding and fractal coding which belong to the transform coding scheme. A real-time CELP system based on a personal computer (PC) and a digital signal processor (TMS320C30 DSP) running on a PC can compress speech from 64 kbps to 4.8 kbps (13.3 : 1) (10). The speed is partly due to a new approach to speech analysis with neural networks. This technique can be further improved by wavelet and fractal (iterative-function system) modeling of its excitation signals to achieve 2.4 kbps or even lower bit rates (11).

The LPC was a predecessor of the CELP technique. Adaptive differential PCM (ADPCM), continuously variable slope delta modulation (CVSD), and adaptive delta modulation (ADM) produce rates above 32 kbps, and belong to waveform coding.

Other signals such as music and biological data [e.g., electrocardiogram (ECG), electromyogram (EMG), and respiratory] may be compressed and decompressed using models of the signal source (e.g., the vocal tract), vector quantization and averaging, differential modulation, and error minimization, as well as wavelets, fractals, and neural networks. The wavelet transform has opened new possibilities in signal compression due to its time-frequency or space-frequency analysis, not available with the standard spectrum Fourier analysis. For music, a model of perception of wideband audio signals has been developed in the wavelet domain (12). Wavelet coefficients in each subband of the wideband audio signal are adaptively coded using truncation, uniform quantization, nonuniform quantization (A-law), optimal quantization, and learning vector quantization with Kohonen's self-organizing feature map (SOFM).

Modeling Digital Sound Sources

A given speech compression technique usually generates a constant bit rate during the speaking intervals ("talkspurts"), whereas no bits are produced during listening pauses in conversation. Experimental data indicate that for English the on–off patterns of speech can be modeled by a two-state Markov process with mean talkspurt duration of 1.34 s and mean duration of pause of 1.67 s (13). For a packet multiplex system with a number of speech channels, the number of active channels at any time follows a binomial distribution. As the number of channels increases, the distribution approaches normal.

DIGITAL IMAGES AND VIDEO

Video coding techniques often rely on transmitting the differences between successive frames, following a successful transmission of the initial frame. Consequently, the bit rate varies from very low during relatively static scenes (e.g., "talking heads" or captions) to high after either a scene cut or rapid changes in the scene. Prior to 1985, the first-generation waveform-based video coding relied on either uncorrelated individual pixels or a block of pixels, without regard to the natural representation (such as shape, texture, and color) of the image (14). The human visual system (HVS), which can recognize an image or a sequence of images using only a few information features such as edges was ignored. Coding schemes included PCM, predictive coding, transform coding, vector quantization, subband, and wavelet coding. JPEG is currently the standard for coding single-frame monochrome and color images (15). It is based on the discrete cosine transform (DCT) applied to blocks of 8 $\times$ 8 image pixels, yielding 64 output coefficients per block. The coefficients are quantized, and differences between blocks are encoded. The encoded coefficients are compressed losslessly by a Huffman coder. Another standard is the Joint Bilevel Image Experts Group (JBIG) for lossless compression of bilevel images based on contextual prediction (16).

The standards MPEG-1 (1.544 Mbps) and MPEG-2 (4 Mbps to 100 Mbps) were based on DCT with the inclusion of motion compensation and variable-length coding. MPEG-2 is aimed at high-definition TV (HDTV). Image varies from HVS at 1.5 Mbps to HDTV at 10 Mbps. The first generation of MPEG culminated in a hybrid scheme that combines a motion compensated prediction (temporal domain) and a decorrelation technique (spatial domain) and is the basis for the proposed MPEG-4 standard. Another standard H.261 was developed for image transmission rather than image storage. It produces a constant output of $p \times 64$ kbps, where p is an integer in the range 1 to 30. The basic algorithm is similar to that of MPEG, with two resolution standards: (i) quarter common intermediate format (QCIF) for desktop and videophone applications and (ii) common intermediate format (CIF) for room systems at a rate of 30 frames per second (fps) with resolution lower than that of broadcast TV (17). A successor standard, H.263, replaced the variable-length lossless encoding in H.261 with arithmetic encoding and reduced the data rate to 20 kbps.

The second-generation of image and video codings uses more efficient image representation in the form of objects because HVS is now a fundamental part in the coding chain. Often, images must be segmented to identify the objects they contain. The new techniques utilize three main approaches: (i) segmentation-based schemes, (ii) model based schemes, and (iii) fractal-based schemes (18). The emerging standard, MPEG-4, is designed to provide content-based interactivity with meaningful objects in an audio-visual scene, content-based manipulation and bitstream editing through the MPEG-4 syntactic description language (MSDL), natural and synthetic data coding, temporal random access, improved coding efficiency, robustness, and scalability. Some of the first- and second-generation techniques are reviewed below.

IMAGE COMPRESSION BY LEARNED VECTOR QUANTIZATION WITH NEURAL NETWORKS

Vector quantization is a joint quantization of a block of data, as opposed to the quantization of a single signal or parameter value (scalar quantization) (6). For example, vector quantization can compress a $512 \times 512 \times 8$ black-and-white image by a ratio of more than 50 : 1, and its decompressed form may still be acceptable to the human eye (5).

An image vector quantizer based on Kohonen's self-organizing feature map (SOFM) has been developed. Gray level images ($512 \times 512 \times 8$) are compressed by 16 : 1 and transmitted at 0.5 bits per pixel (bpp) while maintaining a peak signal-to-noise ratio (PSNR) of 30 dB. The SOFM learns a codebook of prototype vectors by performing vector quantization on a set of training images. While not only being representative of the training set, the prototype vectors also serve as a basis for any other histogram-similar image. Hence, these codebooks quantize other images not in the training set. Various optimization techniques have been studied, for example, the sequential, random, or Gaussian presentation of training vectors produce different codebooks. The Gaussian method biases certain objects in the image that have been deemed significant, such as facial features, by treating all vectors in the image as Gaussian random variables, and setting the mean value to the selected object's center of gravity. Simulated annealing is applied to the SOFM network. By injecting impulses of high temperature at increasing intervals, codebooks learn more quickly. Competitive learning (CL), frequency-sensitive competitive learning (FSCL), and Kohonen's neighborhood learning (SOFM) have been analyzed and compared. It was found that while CL and FSCL achieve slightly higher PSNR than SOFM, the latter is superior at generalizing. This technique may also be important in feature extraction.

Wavelet Image Compression

Wavelet compression provides many opportunities for end-user image processing. A wavelet compression scheme has been developed (11) for complex gray-scale images such as fingerprints guided by multifractal measures (17 : 1), head-and-shoulder images of persons (271 : 1), and severe weather patterns derived from three-dimensional radar data.

Fractal Image Compression

Fractal data compression has attracted a great deal of interest since Barnsley's introduction of iterated functions systems (IFS), a scheme for compactly representing intricate image structures (19). Although the applicability of IFSs to the compression of complicated color or gray-scale images is hotly debated, other researchers have applied the concept of self-similarity (fundamental to fractals) to image compression with promising results. A block-oriented fractal coding technique has been developed for still images (20) based on the work of Arnaud Jacquin, whose technique has a high order of computational complexity, $O(n4)$. A neural network, known as frequency-sensitive competitive learning (FSCL) (21) has been used to assist the encoder in locating fractal self-similarity within a source image, and a judicious development of the proper neural network size for optimal time performance was provided. Such an optimally chosen network has the effect of reducing the time complexity of Jacquin's original encoding algorithm to $O(n3)$. In addition, an efficient distance measure was developed for comparing two image segments independent of mean pixel brightness. This measure, not provided by Jacquin, is essential for determining the fractal block transformations prescribed by Jacquin's technique.

Modeling Digital Image and Video Sources

Highly compressed images follow the pattern of digital data already described. Uncompressed images exhibit patterns due to correlations among their individual subregions.

Modeling of a video source depends on both the coding scheme used (such as the H.261 and MPEG-2) and the video sequence itself. The bit rate profile of a video sequence often exhibits a high autocorrelation, and the measured covariance often follows an exponentially decreasing function. This is true for frame level bit rate. For block-level bit rate, the autocorrelation function is oscillatory. Several plausible models for video traffic include (i) autoregressive models, (ii) discrete-time discrete-state Markov models, (iii) continuous-time discrete-state Markov models, and (iv) self-affine models, as described at the end of this section.

Although very simple, the first model does not represent regions of low probability because the parameters are derived from a typical sequence. The second model uses an M-state Markov model in which the number of states is proportional to the number of bits or packets generated from a single video source. Since transitions can occur only between adjacent states, the model does not capture the high rates produced at scene changes. This feature can be modeled by adding new states. Scene changes can be modeled using a modulating process, such as Markov modulation. The model can be adapted to a multiplex of N channels. The third model uses Markov states to represent the quantized instantaneous bit rate. Rapid scene changes can be accommodated by extending the process to two dimensions. We believe that the fourth approach based on fractality and multifractality has potential to deal with the burstiness of the traffic very well.

TRAFFIC ANALYSIS

With the invention of the telephone network, traffic engineering was born. An understanding of the calling patterns was required in order to design the network. In 1903, Malcolm C. Rorty (AT&T) began to study calling processes. His work was later refined by Edward C. Molina (Bell Laboratories). The model they developed was based on the following assumptions:

1. An average holding time would be the minimum time a call is held in the system, regardless of whether a call was blocked.
2. If the number of calls is greater than the resources available, blocking would occur.

Molina developed a model that was applied in practice by AT&T during the 1920s. This model predicts the probability that a call would be blocked in the system that neither rerouted nor queued. It assumes that the user would retry at a later time. The holding times were not fully considered. It was pointed out to Molina that a similar result had been published a century earlier by S. D. Poisson (a French mathematician), and Molina decided to give full credit to Poisson. The Molina–Poisson model is today known as the Poisson process.

A. K. Erlang (Danish engineer–mathematician) carried out independent research in telephony and in 1909 published two models based on the following assumptions:

1. *Erlang B Model.* Calls that arrive at the system when all the resources are in use would be cleared from the system and would not return. In practice, "cleared" means rerouted to a traffic facility with available capacity.
2. *Erlang C Model.* Calls that arrive at the system when all the resources are in use would be placed in an infinite queue in which calls cannot be abandoned until they are served by the system.

Measurement Units

Telephone traffic is measured in erlangs and in centum (hundred) calls seconds (CCS). An erlang is the mean number of arrivals over an interval whose length is the mean service time. It is the amount of traffic that will keep a traffic equipment busy for 1 h. Thus, a 1 h telephone call will generate 1 erlang. Erlangs are given by the following formula:

$$\text{Traffic load (in erlangs)} = (\text{number of calls/hour}) \times (\text{average holding time in seconds})/3600$$

A CCS is the amount of traffic that keeps a piece of equipment busy for 100 s. The traffic load in CCS is given by the following formula:

$$\text{Traffic load (in CCS)} = (\text{number of calls/hour}) \times (\text{average holding time in seconds})/100$$

The number of calls per unit time (arrival rate) is generally denoted by λ. It is evident that 1 CCS = 36 erlangs. The average holding time (AHT) is given by the following formula:

$$\text{AHT} = (\text{total number of seconds of call activity in an hour})/(\text{number of calls for the same hour})$$

The holding time for a call is the sum of the service time plus the time in queue. There are two generic types of traffic loads: carried load and offered load. The carried load is the load served by the system, and the offered load is the demand presented to the system. The carried load is less than or equal to the offered load due to possible loss.

In data networks, the traffic is a mix of traffic classes such as real-time variable bit rate (VBR), non-real-time VBR, and constant bit rate (CBR), and it is measured in kilobits per second or megabits per second (kbps or Mbps). In addition, there are measures for peak and averages rates, burstiness, and delay tolerance. Traffic measures vary between architectures.

Traffic Sampling

In telephony, there are tables and traffic charts which are given in erlangs or CCS. The standard time unit is the hour. Thus, it is common to collect statistics based on hourly measurements. In telephony it is recommended to monitor the traffic for at least 30 of the busiest hours. The International Telecommunications Union (ITU) recommends an automatic process of statistics collection for a significant part of each day for a whole year. Bell Laboratories recommends that traffic measurements be taken for the 30 busiest days in the year, excluding special holidays, identify the busiest hour in each day's measurements, and then use the averages for traffic studies. Measurements such as the average number of hourly call requests, the average number of blocked requests per hour, and the average holding time are essential.

In data networks, there are no standard recommendations for measurements for all architectures. Statistics dictates the minimum size of the sample for a certain type of analysis. In asynchronous transfer mode networks (ATM), for example, a sample can consist of three to five measurements of 1 h to 3 h during peak hours and similarly for off-peak hours. The measurements should be long enough for rare events to be captured.

Traffic Measurements

In general, a call is a connective request to the network, whether it be a telephone network or a data network. The status of the system can be determined from a set of measurements. Some of the basic measurements to be performed on nodes or trunks are the following:

1. Number of extraordinarily long connections
2. Number of extraordinarily short connections

3. Total number of connections
4. Holding time for carried connections
5. Number of blocked connection requests
6. Number of queued connection requests
7. Number of calls abandoned after having been queued
8. Total time in queue
9. Time that the nodes or trunks were all busy

Traffic Calculations

In the case of systems that block, clear, or queue requests, there are a number of standard calculations that can be made to gauge the state of the system.

The number, N, of customers (connections) in the system is given by

$$N = N_q + N_s$$

where N_q is the number of customers in the queue and N_s is the number of customers in service. The expected (average) number L of customers in the system is given by

$$L = L_q + L_s$$

which is equivalent to $E[N] = E[N_q] + E[N_s]$.

The time in queue, q, plus the time in service, τ, is the holding time, w—sometimes referred to as the sojourn time—that is, $w = q + \tau$; and the expected holding time, W, is

$$W = W_q + W_s$$

which is equivalent to $E[w] = E[q] + E[\tau]$. The times are typically given in hours.

The expected number of customers in the system is given by Little's equation (after J. D. C. Little)

$$L = \lambda W$$

where W is the expected (average) holding time and λ is the average arrival rate of customers to the system. This equation is known as Little's rule or Little's law. In particular it applies to the expected number of customers in the queue, thus

$$L_q = \lambda W_q$$

The expected time in the queue W is given by

$$W = W_q + \frac{1}{\mu}$$

where $1/\mu$ is the expected service time. In terms of λ and μ, the expected number of customers is given by

$$L = L_q + \frac{\lambda}{\mu}$$

Let $a = \lambda/\mu$ be the offered load and a' the carried load in a system. The carried load is given by

$$a' = \lambda/\mu$$

which is equivalent to $a' = a(1 - P[blocking])$. In a system that does not block, the carried load coincides with the offered load. The server occupancy ρ is given by

$$\rho = \frac{\lambda}{c\mu}$$

which is equivalent to $\rho = a/c$, and c is the number of servers. In a blocking system we have

$$\rho = \frac{\lambda'}{c\mu}$$

This is equivalent to $\rho = a'/c$, and λ' is the average arrival rate for customers that are not blocked.

Traffic Models

Traditionally, the telephone network is modeled using exponential distributions for call interarrival times and call service times. It follows that call sojourn times are also exponential. In words, the model assumes that the call times are short most of the time.

Consider call interarrival times. The exponential distribution is given by the following equation:

$$P(T \leq t) = 1 - e^{-\lambda t}$$

where λ is the arrival rate, and $(T \leqslant t)$ is the event that the interarrival time T does not exceed t.

The mean arrival, interarrival, holding, or service time is the inverse; for example, the mean arrival time is $1/\lambda$. The variance is the inverse square; for example, the variance among arrival times is $1/\lambda^2$. Variance is a measure of dispersion in terms of deviation from the average. The percentage of the number of calls that fall within a certain percentile delimited at time T is given by the following equation:

$$\pi(r) = \frac{1}{\lambda}\ln\left(\frac{100}{100 - r}\right)$$

where r is the percentile.

The exponential distribution describes continuous-time processes. Events such as call arrivals are discrete in nature. A discrete-time version of the exponential distribution is the geometric distribution. For the Poisson process it is the binomial. This model represents the probability that n call arrivals occur by time t; it is governed by the following equation:

$$P_n(t) = \frac{(\lambda t)^n}{n!}e^{-\lambda t} \qquad \text{for } n = 0, 1, 2, \ldots$$

Using the Poisson model, the proportion of offered load that is blocked given c servers and a offered load (in erlangs) is given by

$$P(c, a) = 1 - \sum_{i=0}^{c-1} \frac{a^i}{i!} e^{-a}$$

Using the Erlang B model, the probability that a call would be blocked, as a function of the number of servers and the offered load, is given by the following equation:

$$B(c,a) = \frac{\dfrac{a^c}{c!}}{\displaystyle\sum_{i=0}^{c} \frac{a^k}{k!}}$$

A recursive version of the above equation is useful in building tables:

$$B(c,a) = \frac{aB(c-1,a)}{c + aB(c-1,a)}$$

Using the Erlang C model, the probability that a call would be queued is calculated with the following equation:

$$C(c,a) = \frac{\dfrac{a^c}{c!\left(1-\dfrac{a}{c}\right)}}{\displaystyle\sum_{i=0}^{c-1} \frac{a^k}{k!} + \frac{a^c}{c!\left(1-\dfrac{a}{c}\right)}}$$

for $0 \leq a < c$. A recursive version of the above equation, useful in building tables, is:

$$C(c,a) = \frac{cB(c,a)}{c - a(1 - B(c,a))}$$

Enhanced Models

Three models have evolved from the Erlang B model: the Engset model, the Equivalent Random Theory model, and the Retrial model. The main assumption for the Erlang B model is that calls that could not be handled by the system would be cleared and would not return. Another assumption is that there could be an infinite number of traffic sources. When the number of sources is finite, the predicted number of servers from the Erlang B model is too high.

The Engset model was developed by T. Engset in 1918. The probability that a call would be blocked given c servers and s sources is:

$$E(c,s,a) = \frac{\binom{s-1}{c}\hat{a}^c}{\displaystyle\sum_{i=0}^{c-1}\binom{s-i}{i}\hat{a}^i}$$

where a is the total offered load and $\hat{a}$ is the offered load per idle server. A recursive version of the formula above, useful in generating tables, is given by

$$E(c,s,\hat{a}) = \frac{(s-c)\hat{a}E(c-1,s,\hat{a})}{c + (s-c)\hat{a}E(c-1,s,\hat{a})}$$

The offered load per idle server can also be calculated as:

$$\hat{a} = \frac{a}{s - a(1 - E(c,s,\hat{a}))}$$

The Equivalent Random Theory model is based on the concept that for every peaked traffic load there is an equivalent random load that yields the same amount of overflow traffic when it is offered to a number of trunks. The model was developed by R. Wilkinson of Bell Laboratories in 1955 and was refined by S. Neal (also of Bell Laboratories) in 1972. This model is also known as the Neal–Wilkinson model. The assumption is that when a route is too busy, calls are redirected to another route. A quantity called the peakedness factor z was introduced to characterize the burstiness of the traffic. The Erlang B tables were updated in 1982 by H. Jacobsen of AT&T, who introduced comparative costs and blocking factors in what is called the EART and EARC tables. The Equivalent Random Theory states that "for any mean offered load, a, and variance, v, describing a nonrandom load (variance > mean), there is a random load a' which, when offered to a group of c trunks (c is not necessarily an integer), would produce an overflow traffic with the same parameters a and v."

This model is now in use for determining the last-choice route in networks that use alternate routing. The model is given by the following equations:

$$\begin{aligned}
a &= a'E(c,a) \\
z &= a\left(-a + \frac{a'}{c' + 1 + a - a'}\right) \\
\alpha &= a'E(c'+c, a') = ba \\
z' &= a\left(-\alpha + \frac{a'}{c' + 1 + \alpha - a'}\right)
\end{aligned}$$

where a' is the equivalent random load, z' is the peakedness factor associated with the overflow, b is the blocking factor, c is the size of the trunk for the equivalent load, α is the overflow load, and $E(x, y)$ is the Erlang B model.

The Retrial model was developed by R. Wilkinson of Bell Laboratories around the time the Equivalent Random Theory model was being proposed. This model was later enhanced by H. Jacobsen at AT&T in 1980. The model is based on the assumption that when calls are blocked, the callers do not give up, but they hang up and try again. It was developed as an attempt to account for the discrepancy between reality and the Poisson model. The assumptions are that new calls arrive following an exponential interarrival time distribution, blocked calls retry, and completed calls leave the system. The Retrial model was used to generate tables which are given in terms of the following variables: the number of calls in the system, k, the number of calls being retried, l, the average service rate (per unit of time), μ, the number of servers, c, and the arrival rate, λ.

The Erlang C model was used until about the end of the 1970s. The Erlang C model assumes that blocked calls wait in an infinite queue for service for an indefinite amount of time. It was used to produce a set of Erlang C Infinite Queueing tables which consist of CCS Capacity tables, Probability of Delay tables, Average Delay of Delayed Calls in Multiples of Holding Times tables, and Server Occupancy tables. These tables assume exponential arrivals and exponential holding times.

The variables used to construct the above-mentioned tables are similar to the previous models. In addition, let k be the total capacity of the system given by the number of servers plus the number of queue slots, let b be the blocking probability, let d be the probability of delay for nonblocked calls, and let W_q be the average delay in the queue. The CCS Capacity table is constructed from the following equation:

$$\mu W_q = \frac{a^c}{(c-1)(c-a)}\, p_0$$

where

$$p_0 = \left[\sum_{n=0}^{c-1} \frac{a^n}{n} + \frac{a^c}{c\left(1-\dfrac{a}{c}\right)}\right]^{-1}$$

The Probability of Delay tables are generated by the equation

$$\text{Delay probability} = C(c,a) = \frac{a^c}{c\left(1-\dfrac{a}{c}\right)}\, p_0$$

The Average Delay of Delayed Calls in Multiples of the Average Holding Time tables are generated by using the expression:

$$E[q|q>0] = uW_q\left[\frac{1}{1-d}\right]$$

The Server Occupancy table is calculated from the expression:

$$\rho = \frac{a}{c}$$

The Erlang C model was adjusted once it became clear that queued calls do not wait for service indefinitely and that finite queues in real equipment make a significant difference. The set of Erlang C Finite Queueing tables consist of tables similar to the four tables of the Infinite Queueing case plus a Queue Length table.

In these tables a grade of service indicating the percentage of calls to be blocked is required. This value is either given or calculated. In general, the number of servers is obtained from the CCS Capacity tables, and the other tables are used to obtain the mean delays, the mean queue size, and the percentage of time that the server is busy. The variables are the same as in the Infinite Queueing tables.

The CCS Capacity tables are calculated from the equation:

$$b = \frac{1}{c(k-c+1)}\, a^k p_0$$

where for $a/c \neq 1$, p_0 is given by

$$p_0 = \left[\sum_{n=0}^{c-1} \frac{a^n}{n} + \frac{a^c\left(1-\left(\dfrac{a}{c}\right)^{k-c+1}\right)}{c\left(1-\dfrac{a}{c}\right)}\right]\frac{a}{c}$$

and for $a/c = 1$, p_0 is given by

$$p_0 = \left[\sum_{n=0}^{c-1} \frac{a^n}{n} + \frac{a^c(k-c+1)}{c}\right]\frac{a}{c}$$

The Probability of Delay tables are calculated based on the following equation, where d is the probability of delay for nonblocked calls:

$$d = \left[\sum_{n=c}^{k} \frac{1}{c^{n-1}c!}\, a^n p_0\right] + 1 - b$$

The Average Delay of Delayed Calls in Multiples of the Average Holding Time table and the Server Occupancy table are calculated with exactly the same formulas as in the Infinite Queueing case. The Length of the Queue required is given by the following expression, where L is the length of the queue:

$$L = \frac{p_0 c a^{c+1}}{c!(1-ca)^2}[1-(ca)^{k-c+1}-(1-ca)(k-c+1)(ca)^{k-c}]$$

The Kendal Notation

In 1953, D. G. Kendal proposed a notation for describing telecommunications traffic models. In the Kendal notation a traffic model is represented by a string of the form $A/B/c/k/s/Z$, where A represents the type of arrival process, B indicates the type of service process, c is the number of servers (or channels), k is the capacity of the system (queue size plus number of servers), s is the size of the source population, and Z indicates the queueing mode. A, B, and c are always specified; if k and s are not infinite, they are specified too; Z by default specifies queueing discipline of the type first-in, first-out (FIFO; also called first-come, first served, FCFS).

The arrival process A and the service-time process B are described by the following symbols representing interarrival time distributions: M for Markovian (exponential), E_k for Erlang type k, H for hyperexponential, h for hypoexponential, and G for general. The numbers c, k, and s are integers greater than zero (or infinity). The queueing discipline Z can be FCFS, LCFS (last-come, first-served), SIRO (service in random order), GD (general discipline), and RR (round robin).

The notation for the arrival process A was developed by Erlang based on an older notation that classified this process as smooth, random, or peaked. The notation can also include a superscript as in M^k, where k indicates group arrivals. The subscript k in E_k represents the number of stages in the Erlang process. H arrivals were called peaked in the older notation, while the h arrivals were called smooth. Both can have a subscript to represent the number of stages in the process. If the arrival process does not match any of the other patterns, then it belongs to the general category.

It is assumed that the servers in the system always work in parallel. In the case that the system capacity k is infinite, the Erlang C and Erlang C Infinite Queueing tables can be used. If k is finite and larger than the number of servers, the Erlang C Finite Queueing model can be used. If the system capacity is equal to the number of servers, the Erlang B model is used.

Most queueing systems assume that the source population is infinite. If the model requires a finite source population, the Engset model would be appropriate.

TELEPHONY TRAFFIC

In the design of telephone networks, it is important to consider issues such as how many people want to use it, for how long, and how often. These issues are represented as calling rates and traffic intensity.

A set of lines connecting one major point to another—for example, two cities—are called "trunks" in North America and "junctions" in Europe. A subscriber is connected to the network through a local "exchange." A "local area" defines a service area containing a number of local exchanges. A "toll area" is a long-distance area that consists of several trunks and exchanges that connect local areas.

In long-distance networks there are two transmission effects that must be addressed. They are "echo" and "singing." The echo effect is the return of the sender's voice signal to the sender's telephone. This is caused in part by electric impedance mismatches in the network. The singing effect refers to the oscillations in the signal. This is typically caused by positive feedback in the amplifiers.

In telephone networks, the traffic models discussed above apply in the modeling of trunk groups, attendant groups, hunt groups, modem pools, ringing supply circuits, recorded announcement slots, time slots, call progress tone detectors, touch-tone detectors, speech-synthesizer circuit packs, dial-pulse registers, callback queues for trunks, and many more systems.

There are different kinds of trunks. In stand-alone trunks the users hang up and try again when the line is busy. There is no queueing or rerouting. Depending on the users' tolerance to redial and costs, a grade of service and size of trunk can be selected. These trunks are analyzed with the Retrial model. Alternate routing trunks reroute blocked calls. The size of these trunks depend on the volume of calls, grade of service, traffic load during peak hours, and costs. These trunks are modeled with the EART and Neal–Wilkinson models for the cases of primary/intermediate trunks and last-choice trunks, respectively. Queueing trunks hold their blocked calls for a finite time until they can be serviced. The size of the trunk is determined by the tolerance of users to wait in the queue, desired speed of service, and costs. These trunks are modeled by Erlang C Finite Queueing models.

A summary of traffic models with their Kendal notation and suitable applications is given below

1. Erlang B ($M/M/c/c$): Touch-tone telephones.
2. Engset ($h/M/c/c/s$): Statistical multiplexers and concentrators.
3. Retrial ($H/M/c/k$): Stand-alone trunk groups. Blocked requests are retried.
4. Neal–Wilkinson ($H/M/c/k$): Final trunk groups. Blocked requests are retried.
5. EART/EARC ($H/M/c/c$): First-choice trunk groups. Requests are rerouted. Typically the least costly.
6. Erlang C, finite servers, infinite queue ($M/M/c$): Callback queues. Requests are queued.
7. Erlang C, single server, infinite queue ($M/M/1$): Switch activity monitors. Requests are queued.
8. Erlang C, finite servers, finite queue ($M/M/c/k$): Automatic call distribution groups. Blocked requests are retried.

TRAFFIC IN COMPUTER NETWORKS

Computer networks carry traffic different from telephony networks. In most cases, the traffic is packet based. A packet is a unit of information consisting of payload and overhead (header and/or trailer) that travels as one unit in the network.

Traffic in computer networks is typically organized in a layered architecture. The International Organization for Standardization (ISO) introduced the Open Systems Interconnection (OSI) reference model, which consists of seven layers, each layer specifying functions and protocols required to establish communication between two points. The layers are: physical, data link, network, transport, session, presentation, and application. The layers interact with each other to establish a physical communication link, to send/receive data, to find the destination point, to retransmit if necessary, to establish a communication session, to format the data in a way comprehensible to a user, and to carry out an end-to-end application, respectively.

There are two main methods for transmitting packets in computer networks: synchronous and asynchronous. In the former, a computer sends a packet synchronized by a clock. In the latter, a computer sends a packet at any time.

There are two basic types of call requests: connection oriented and connectionless. In the former, the sender requests a connection; and if enough resources exist, the network establishes a channel which will be used by all the packets from the source. In the latter, every packet or group of packets travels independently. Transmission Control Protocol/Internet Protocol (TCP/IP) is an example of a connectionless service, while ATM is an example of a connection-oriented service.

In the case of computer networks such as those based on ATM, it was proposed that traffic be modeled with $M/M/1$ and $M/G/1$ models. However, as it will be described below, the need for more appropriate models has been identified. Even though the evolution of the telephony network has provided a wealth of traffic models, in the world of packet networks fewer results exist that can be applied in practice.

One of the major computer networks in the world is the Internet. Initially, it was used for non-real-time data communications. The evolution of the Internet to real-time services requires that Internet protocols be furnished with quality of service (QoS) capabilities. As this evolution takes place, it becomes necessary to guarantee continuous QoS support across the Internet, intranets, and backbone networks. A backbone network is a computer network that operates at the lower layers of the OSI model. ATM standards are supervised by the ATM Forum. This organization produces specifications for standards. The user network interface (UNI) specifications for ATM have evolved to include support for a mapping between the Internet and ATM.

In ATM packets are called cells, and consist of 53 bytes; five of these constitute a cell header, and the remaining 48 are the payload. Thus, a message from a higher-layer protocol is broken into groups of 48 bytes. One of the main characteristics of ATM is the capability of establishing virtual circuits and virtual paths. A virtual circuit resembles a line in telephony and a virtual path resembles a trunk.

If ATM networks are to be integrated with new Internet services, mapping the ATM Forum's UNI QoS specifications to the emerging Internet technologies becomes a key issue. Currently, Internet services are being delivered partly through ATM backbones. Since ATM supports QoS by design and the demand of Internet services is typically non-real-time, the integration is done via best effort services. Thus, the capabilities of ATM backbones are not fully utilized. In addition, there is a general agreement in the telecommunications community that increasing bandwidth alone cannot solve these problems.

The Internet uses a four-level reference model. It is similar to the OSI model, except that the upper five layers are combined into two layers. The main protocols are the Internet Protocol (IP) for routing and the Transmission Control Protocol (TCP) for flow control and error recovery. Combined, they are known as TCP/IP.

Examples of real-time applications over the Internet include Internet telephony, video, and whiteboards. Many service providers carry part of their traffic over standard telephone networks which were not designed for long call holding times. As the Internet itself is being re-engineered, it is becoming evident that one of the main contributions of the present telecommunications revolution is real-time services.

Four service classes for Internet traffic can be identified. They are guaranteed service, predictive service, controlled delay service, and best effort service. In the Internet, protocols such as the Resource Reservation Protocol (RSVP) are evolving to provide a way to guarantee a QoS on the Internet. RSVP allows a user to reserve resources for real-time applications.

Quality of Service in ATM

Quality of service in ATM is a term which refers to a set of performance parameters that characterize a transmission quality over a given connection.

The ATM Forum has specified performance objectives depending on the QoS class. The objectives have only been defined for three classes (1, 3, and 4). Additional classes (2 and Unspecified) have objectives that are derived from the other classes. The performance parameters used to define QoS classes 1, 3, and 4 are the cell loss ratio (CLR), cell transfer delay (CTD), and cell delay variation (CDV).

QoS class 0 (or class U): This class is intended for services in which no performance objectives need to be specified. It is intended to include services from the unspecified bit rate (UBR) category.

QoS class 1 (or class A): This class is intended to meet stringent cell loss requirements for applications including circuit emulation of high capacity facilities.

QoS class 2 (or class B): The objectives are similar to those of class 3, with the exception that its CLR objective applies to all cells rather than to just high-priority cells.

QoS class 3 (or class C): This class is intended to meet connection-oriented or connectionless data transfer applications that have minimal delay needs. It is appropriate for connections such as non-real-time variable bit rate (VBR) traffic. An available bit rate (ABR) service category can also be served in this class.

QoS class 4 (or class D): This class is intended for low latency, connection oriented or connectionless data transfer applications. It is also intended to provide interoperability with IP.

A very important issue in guaranteed QoS specification over the Internet is that these requirements are maintained by backbone networks. Thus, QoS mapping becomes a key issue.

SONET/SDH

The Synchronous Optical Network (SONET) is a technology used to carry traffic over wide area networks (WANs). Its electrical equivalent is the Synchronous Digital Hierarchy (SDH). These types of transmissions are measured in erlangs and CCS units. In other words, a SONET line busy for 1 h has an activity of 1 erlang associated with it.

Traffic Standards

The cell transfer performance parameters of a broadband switching system (BSS) are based on the following standards:

ITU-T (International Telecommunications Union—Telecommunications Sector) Recommendation I.356, B-ISDN ATM Layer Cell Transfer Performance, July 1993; plus Draft Revised Recommendation I.356R, May 1996 (definition of the QoS classes).

NSI T1.511, B-ISDN ATM Layer Cell Transfer Performance Parameters, 1994.

Types of Performance Measurements

There are two types of performance measurements: the call processing performance, which relates to the establishment and teardown of connections for connection-oriented services; and the cell transfer performance, which relates to information transport performance on established ATM connections. Call processing performance is measured based on the connection setup denial probability, the connection setup delay, and the connection clearing delay.

The connection setup denial probability is the fraction of connection attempts that are unsuccessful. The connection setup delay is the delay between incoming and outgoing signaling messages associated with the establishment of a call. These measurements are typically done during peak hours. The connection clearing delay is the delay between a release message and a release complete message during the clearing of a connection. These measurements are typically done during peak hours.

Performance of ATM cell transfer is based on the cell loss ratio (CLR), cell transfer delay (CTD), and cell delay variation (CDV).

The CLR is the ratio between the number of cells lost (cells not delivered to their destination port) and the total number of incoming cells (lost + delivered). These measurements are taken at the input and output interfaces. The CTD is the elapsed time from the moment the first bit of a cell enters the ATM network to the moment the last bit of the cell leaves the output port. The delay has several components. The network contribution to the delay for all services includes cell construction delay, propagation delay, transmission delay, and node queueing and switching delays. The CDV is caused by contention among connections for cell slots at multiplexers and switches. CDV impacts service performance for constant bit rate (CBR) traffic.

Following is an approximation to two-point CDV in terms of its quantifier Q_α:

$$Q_\alpha = \max[\mu - \delta_1, \mu - \delta_2]$$

where μ is the mean cell transfer delay of a connection, and δ_1 and δ_2 satisfy

$$P[\text{cell delay} \le \delta_1] < \alpha$$
$$P[\text{cell delay} \ge \delta_2] < \alpha$$

Common rates for ATM are: DS1 at 1.544 Mbps, DS3 at 44.736 Mbps, STS-1 at 51.84 Mbps, OC-3 or STS-3c at 155.52 Mbps, OC-12 or STS-12c at 622.08 Mbps. The transfer capacity of OC-3 is 353,207 cells/s, and the average emission time for an ATM cell is about 2.83 μs.

Reference Traffic

Reference traffic loads are defined for testing. Let N_i and N_o be the number of input and output ports, and let *n_vc* be the number of traffic sources. A cell-level test source is associated with each virtual circuit, and traffic sources should be equally distributed among the input and output ports.

It is necessary to preserve the initial cell sequence on each virtual circuit used in the test, and for this reason the ATM cell multiplexer used during the test should have an FIFO buffer in each input.

For performance tests of a virtual circuit that supports CBR traffic the QoS is assumed to be of class 1. For sources supporting VBR traffic, the QoS is assumed to be of class 3 or 4. There are four types of CBR sources and three types of VBR. The CBR test sources have PCR values specified in the standards. For example, for OC-3 the PCR value is 353,207 cells/s (135.53 Mbps including operation and maintenance cells OAM); for DS3 the PCR is 96,000 cells/s (36.86 Mbps including OAM cells); and for DS3 the PCR value is 104,268 cells/s (40.04 Mbps including OAM cells).

The types of CBR test sources are the following. In CBR test source I the PCR is 4140 cells/s. The phases of *n_cbrI* CBR I test sources should be randomized uniformly on connection establishment, over an interval of 241.5 μs. CBR test source II is defined with a PCR of 16,556 cells/s. The phases of *n_cbrII* CBR I test sources should be randomized uniformly on connection establishment, over an interval of 60.4 μs. CBR test source III is defined with a PCR of 119,910 cells/s. The phases of *n_cbrIII* CBR I test sources should be randomized uniformly on connection establishment, over an interval of 8.34 μs. In CBR test source IV, a PCR of 173 cells/s is used. The phases of *n_cbrIV* CBR I test sources should be randomized uniformly on connection establishment, over an interval of 5780 μs.

The Bellcore Recommendations specify that VBR sources can be characterized by a two-state Markov process consisting of (a) an active state during which the source generates payload cells and (b) a silent state during which cells are not generated. Other more appropriate models have been proposed and some of them are covered later in this article.

For a given reference load, the duration of the active states of all VBR test sources should be identically independently distributed (IID) random variables. The duration of an active phase has an integer number of cell slots with a geometric distribution of mean M_a.

In the active state a VBR source produces a synchronous burst of cells with period P, where P is an integer number of cell slots. The mean burst size is given by

$$B = \left[\frac{M_a}{P}\right] \text{cells}$$

where $[x]$ denotes the least integer function of x, also known as the floor function $\lfloor x \rfloor$.

The silent state lasts an integer number of cell slots, which is also geometrically distributed with mean M_s.

The occupancy of an output link due to a single VBR test source is given by

$$LO_{vbr} = \frac{B}{M_a + M_s}$$

The mean cell rate of a VBR test source on an OC-3 link is $CR_{vbr} = LO_{vbr} \cdot 353{,}207$ cells/s. The mean cell rate of a VBR test source on an DS3 is $CR_{vbr} = LO_{vbr} \cdot 96{,}000$ cells/s.

The mean occupancy ratio of N_o output links is

$$\rho_{vbr} = \frac{n\_vbr \cdot LO_{vbr}}{N_o}$$

where *n_vbr* is the total number of VBR traffic sources.

There are three types of VBR test sources, all carried on QoS class 3 or 4. For VBR I, M_a is 240 cell slots, P is 6 cell slots, and M_s is 720 cell slots. For VBR II, M_a is 500 cell slots, P is 25 cell slots, and M_s is 2500 cell slots. For VBR III, M_a is 210 cell slots, P is 1 cell slots, and M_s is 2500 cell slots.

Performance Objectives Across the Network

The performance target of an ATM connection from classes 3, 4, and U can be met with class 1 parameters. But this can result in inefficient transmission. A network element that supports classes 1 and 3 should guarantee class 3 performance objectives but not necessarily those for class 1. A network element that supports classes 1, 3, and 4 should guarantee performance objectives for class 3, but not necessarily those for classes 1 or 4. Connections of class 4 should guarantee performance objectives for class 4, but not necessarily those for class 1. The same applies for a network element that supports classes 1, 3, 4, and U, and the connections of class U do not need to meet the performance objectives of the other classes.

In the case of OC-3 and OC-12 the performance objectives are a CLP $\le 10^{-10}$ for QoS class 1, $\le 10^{-7}$ for QoS class 3, and $\le 10^{-7}$ for QoS class 4. The CTD is 150 μs for QoS classes 1 and 4, and not specified for QoS class 3. The CDV is 250 μs for QoS classes 1 and 4, and not specified for class 3.

A network element should also support different QoS classes in each direction of a virtual circuit connection. Traffic shaping is a methodology used to minimize the CDV. The idea is to smooth out the traffic before it enters the network. This is also known as traffic policing.

Traffic Load Specification

The mean output link occupancy ratio, ρ, corresponds to the maximum traffic load of a particular type. For example, for *n_cbrI* test sources used on N_o OC-3 output ports, the mean occupancy ratio is given by

$$\rho_{cbrI} = \frac{n\_cbrI \cdot 4{,}140}{N_o \cdot 353{,}207}$$

Typically a network element is tested with 2 CBR, 3 VBR and 2 mixed bit rate sources (MBR), and the equipment supplier provides the performance measurements.

For CBR sources, the *n_cbr*, N_i, N_o, ρ_{cbr}, the capacity of the input ports (bandwidth), and QoS parameter values are provided. For VBR sources, the *n_vbr*, N_i, N_o, ρ_{cbr}, and QoS parameter values are provided. For mixed sources the *n_vbr*, N_i, N_o, ρ, the capacity of the input ports, and QoS parameter values for CBR and VBR are provided. The network element supplier must also specify the number of VBR test sources operating with 40 CBR sources for OC-3 and 160 CBR sources for OC-12.

Call-Level Models

Typically call arrivals are assumed to occur independently following a Poisson model in an ATM virtual circuit. However, the requested capacity (bandwidth) of ATM calls can vary, depending on the specific application supported. More sophisticated models will be discussed later.

Call holding times are typically assumed to be exponentially distributed, while ATM call holding times vary with specific applications.

A supplier of network elements should specify the capacity (bandwidth) and number of input ports, the capacity and number of output ports, and the maximum number of independently occurring ATM call attempts per hour for each of the PCR values 1.5 Mbps, 6 Mbps, 10 Mbps, 135 Mbps, and 620 Mbps. Connection denial probabilities for these values will be discussed below for exponentially distributed call holding times with a mean of 3 min.

For prescribed connection setup delay (point-to-point) and connection clearing delay objectives, the supplier must state the type of signaling message processor used in the test, any information necessary for proper interpretation of the results, and the maximum number of independently occurring ATM call attempts per hour.

During testing of a network element it is also important to identify the maximum call-level traffic load under which a virtual circuit meets the connection setup delay, connection clearing delay, and connection denial probability objectives.

The connection setup delay (point-to-point) must be 150 ms or less. Since traffic analysis techniques are not as advanced as the telephony techniques, a network element should include protective measures against a large number of call requests in a short time (a burst) that could reduce its call processing capability. The connection clearing delay (point-to-point) must be less than 90 ms. The connection denial probability (CDP) depends upon the peak cell rate (PCR) that is contained in the signaling message that carries the connection setup request. For a CDP of 10^{-6}, the bandwidth of the call should be less than 1.5 Mbps, and the PCR should be less than 4140 cells/s. For a CDP of 10^{-5}, the bandwidth of the call should be less than 6 Mbps, and the PCR should be less than 24,840 cells/s. For a CDP of 10^{-4}, the bandwidth of the call should be less than 10 Mbps, and the PCR should be less than 41,400 cells/s. For a CDP of 10^{-3}, the bandwidth of the call should be less than 135 Mbps, and the PCR should be less than 353,207 cells/s. For a CDP of 10^{-2}, the bandwidth of the call should be less than 620 Mbps, and the PCR should be less than 1,412,828 cells/s.

Performance Objectives for Multipoint Connections

Multipoint connections should provide the same QoS performance as point-to-point connections. Multipoint connections include: point-to-multipoint, multipoint-to-point, and multipoint-to-multipoint. In the case of point-to-multipoint virtual paths or virtual circuits, independent (asymmetric) QoS classes must be supported.

TCP/IP Over ATM

ATM cells are grouped into frames of variable size which match the size of TCP/IP datagrams (packets). There are different types of ATM frames, the most common for VBR services is the ATM Adaptation Layer 5 (AAL5). An AAL5 frame is used for each IP packet. The segmentation and reassembly of frames adds some variation to the transfer delay experienced by TCP/IP datagrams. QoS class 3 is typically the one used for TCP/IP over ATM.

In a TCP/IP over ATM, transmission frames are sometimes lost due to missing cells. This is because incomplete frames fail the cyclic redundancy check (CRC) and TCP discards and retransmits incomplete frames. If the last cell of a frame is lost, the remaining cells will be considered as part of the next frame, and both frames will be dropped because the combined frame will fail the CRC. The last cell of a frame contains a special bit in its header called the ATM-layer-user to ATM-layer-user (AUU) bit; the purpose of this bit is to notify the receiver that the end of a frame has been reached. In ATM, cells are dropped sometimes when bit errors cause problems in frame assembly. By its very nature, ATM is a statistical multiplexing technique that sometimes exceeds its capacity and thus drops cells.

Thus, the loss of a single bit in a cell could cause an entire TCP/IP frame (or equivalently an AAL5 frame) to be discarded. However, the TCP/IP protocol is designed to handle these problems.

Usage Measurements

Usage measurements provide a way to measure the amount of data transported on an ATM connection. Typically, the ingress total cells, ingress high-priority cells, egress total cells, and egress high-priority cells are measured. The minimum measurement interval is 1 h, and the maximum is 1 day. Sometimes intervals of 15 min are used.

In some cases it is also desirable to include service parameters such as the pack cell rate (PCR), quality of service (QoS), and sustained cell rate (SCR). These are measured at the input and output ports.

TRAFFIC ENGINEERING AT THE FOREFRONT

Traffic characterization is an important aspect that has to be considered for efficient network management and control. This is especially important for computer networks, because the variety of sources and the nature of multimedia information on these networks complicate resource allocation problems.

Traffic characterization techniques for computer networks can be classified by the nature of the traffic descriptors into the following categories: autoregressive moving average (ARMA) models, Bernoulli process models, Markov chain models, neural network models, self-similar models, spectral characterization, autoregressive modular models such as transform expand sample (TES) and quantized TES (QTES) models, traffic flow models, and wavelet models.

The traditional traffic descriptors are the mean, peak, and sustained rates, burst length, and cell-loss ratio. These values capture only first-order statistics, and a need has been identified for descriptors that provide more information in order to describe highly correlated and bursty multimedia traffic.

The natural approach is the use of traditional traffic models which have been used in the modeling of nodes. Other concepts such as packet-trains have also been applied.

It is widely accepted that short-term arrival processes in telecommunication networks can be accurately described by Poisson processes—for example, a file transfer protocol (FTP) control connection which can be modeled as a Markov modulated Poisson process (MMPP). However, it has been identified that long-range dependencies found in certain multimedia traffic can be better described using the concept of self-similarity and also using autoregressive integrated moving average (ARIMA) models. For example, FTP arrivals can be modeled using an ARIMA model.

The concept of self-similarity in communications systems, also known as fractality, was introduced by B. Mandelbrot in the mid-1960s. Mandelbrot also introduced telecommunications-related models such as fractional Brownian noise. Since then, these concepts have played a key role in compression techniques, in signal processing, and more recently they have played an important role in the analysis of network traffic. These models can capture long-term dependencies in traffic, which admit higher-order statistical measures as descriptors.

Self-similarity refers to the property of an object to maintain certain characteristics when observed at different scales in space or time. The concepts of long-term dependence (LRD) and self-similarity have been extensively studied in various universities and research laboratories such as Bell Laboratories, Bellcore, and the Telecommunications Research Laboratories (TRLabs). Models have been proposed that use the term fractality in the sense that the autocovariance of the traffic exhibits self-similarity. Other self-similar models include fractional ARIMA processes, fractional Gaussian noise, renewal reward processes and their superposition, renewal processes, and the aggregation of simple short-range dependent models (On/Off).

Self-similar models have been applied in the analysis of VBR video, LAN traffic, traffic generation, progressive image coding for packet-switching communications, and estimation from noisy data. A bibliographic guide on self-similarity techniques in the context of telecommunications was presented by Willinger et al. (22) in 1996.

Another approach suitable for modeling VBR video is based on TES models. This approach fits a model of the empirical distribution (histogram) and empirical autocorrelation function simultaneously. This approach is especially suitable for traffic generation, while QTES models, which are a discrete state variant of TES, are suitable for queueing analysis. For a comprehensive review of these types of processes see the excellent review by Melamed (23) and references therein.

Neural networks have also been applied in traffic modeling for their ability to classify and implement nonlinear mappings. Neural networks are especially suitable for prediction and control.

Frequency domain techniques like spectral analysis has also been applied to model wide-band input processes in ATM networks. In addition, wavelet coding has also been explored. Wavelets provide a convenient way to describe signals in the time-frequency domain. These have been applied with techniques such as weighted finite automata, vector quantization, self-organizing maps, and simulated annealing.

Self-Similar Traffic

It is generally accepted that certain traffic on packet networks exhibits long-range-dependent (LRD) properties. One of the main contributions to this discovery was presented by Leland from Bellcore for Ethernet traffic in 1994. A clear discrepancy was found between predictions of traditional models and empirical measurements in networks. One of the manifestations of the property of LRD is self-similarity in traffic.

There is evidence that the self-similarity observed in traffic has practical implications for network design and management. Traditional short-range dependent (SRD) input streams (e.g., Markovian) to queues affect queueing performance, and many techniques are available to analyze the implications in network management and control. In contrast, few theoretical results exist for queueing systems with LRD inputs.

Network planners have observed that capacity estimations are inaccurate when trying to take advantage of statistical multiplexing. This is due mostly to the fact that the models used do not account for the self-similarity characteristics of the observed traffic. The quest for an "Erlang"-type model for ATM is the driving force for research in ATM traffic characterization. Standards will eventually be updated, and network design will enjoy better theoretical support.

Second-order statistics have been used in modeling traffic—for example the index of dispersion for counts (IDC), or equivalently the variance-time curve. For an MMPP model, the IDC graph increases and saturates, indicating short-range correlation. In the case of a self-similar model, the IDC curve increases monotonically.

In the 1960s, a study on river flow changes by H. E. Hurst resulted in a model which has been successfully used in several disciplines, including ATM traffic analysis. The technique models the variance–time curve in terms of a power law that is characterized by the Hurst parameter. This parameter is related to the slope of the variance–time curve. In fact, it is related to the slope of the asymptote of the curve, but in practice a simple linear regression ignoring the points indicating short-term correlation is employed. This technique has been used to measure burstiness changes in ATM. A similar technique that has been used is R/S analysis. For more information on other techniques the reader is referred to the surveys of traffic characterization techniques for packet networks in the reference section of this article.

In the standards for ATM, Markovian models have been recommended. Clearly, these models are too simplistic for describing a process as complex as a packet network with statistical multiplexing.

It is of great interest in network planning to have accurate models for traffic and performance characterization. Some descriptors do not appropriately capture the observed behavior. For example, the peak to average ratio, generally used to describe burstiness, does not provide information about the rate of change of the burstiness.

The following provides some mathematical details on the structure of a self-similar process. A self-similar process has been defined as a covariance-stationary stochastic process $X = \{x_t\}$, with mean $\mu = E\{x_t\}$, finite variance $\sigma^2 = E[(x_t - \mu)^2]$, and autocorrelation function

$$r(k) = \frac{E[(x_t - \mu)(x_{t+k} - \mu)]}{E[(x_t - \mu)^2]}$$

If $\sigma = \infty$ then $r(k)$ is not well-defined. Recall that in a covariance stationary process, the statistics depend only on the distance of the points in a time series, not on the time index.

Some of the differences between traditional models and self-similar models are as follows. Consider a covariance-stationary (short-range-dependent) traffic process $X = \{x_k\}$. Its autocorrelation function decays exponentially fast according to ("$\sim$" means asymptotically proportional)

$$r_x(k) \sim a^{|k|} \qquad \text{as } |k| \to \infty, 0 < a < 1$$

This implies a summable autocorrelation function $0 < \sum_k r_x(k) < \infty$. Such processes can also be characterized by $\mathrm{var}(X^m) \sim m^{-1}$ as $m \to \infty$.

It is generally accepted that the autocorrelation function for packet traffic processes has the form

$$r_x(k) \sim |k|^{-\beta} \qquad \text{as } |k| \to \infty, 0 < \beta < 1$$

This implies that $\sum_k r_x(k) \to \infty$. Such processes can also be characterized by $\mathrm{var}(X^m) \sim m^{-\beta}$ as $m \to \infty$. The Hurst parameter is used to measure the degree of LRD and is given by $H = 1 - \beta/2$, implying that $1/2 < H < 1$.

For short-range dependent processes the slope of the variance-time curve is -1, and for LRD it is between -1 and 0. H can be estimated from the slope of the variance–time curve in a log–log plot. Notice that for short-range dependent processes $H = 1/2$. It has been observed that for Ethernet traffic $H \approx 0.8$.

In summary, the LRD property has the following statistical manifestations: The traffic is self-similar, the associated autocorrelation function decays hyperbolically (for $1/2 < H < 1$), and the variance of the aggregated processes decays more slowly than traditional models. Another characteristic is that the densities of packet interarrival times and burst lengths have a heavy tail. Additionally, the spectral densities exhibit the $1/f$-noise phenomenon described by Erramilli et al. (24) in 1996. In the frequency domain, LRD manifests itself in a spectral density of the form $s_x(k) = \sum_k r_x(k)e^{ikw}$ that obeys the following power law near the origin:

$$s_x(k) \sim |\omega|^{-\gamma} \qquad \text{as } \omega \to 0, 0 < \gamma < 1$$

The Hurst parameter is given by $H = (1 + \gamma)/2$.

A covariance stationary process is called asymptotically self-similar with self-similarity parameter $H = 1 - \beta/2$ if, for all sufficiently large m, $r_x^{(m)}(k) \sim r_x(k)$ as $|k| \to \infty$, where $r_x^{(m)}(k)$ denotes the autocorrelation function of the aggregated process (i.e., block averages). The process is exactly second-order self-similar if equality holds for all k and m.

Fractional Gaussian noise (i.e., the increment processes of fractional Brownian motion) is an example of an exactly second-order self-similar process, with self-similarity parameter $1/2 < H < 1$. Fractional ARIMA models are examples of asymptotically self-similar processes with self-similarity parameter $H = d + 1/2$, $0 < d < 1/2$.

All the properties described above are often collectively referred to as traffic fractal properties.

Multifractal Model of Packet Traffic

This section introduces an approach to ATM traffic modeling using generalized entropy concepts. The type of processes considered generalize self-similar processes and are generally known as multifractal processes. The key feature that distinguishes these models is that their behavior is described by higher-order moments. It has been shown experimentally that ATM traffic can be characterized adequately by a fifth moment, rather than a second moment such as the IDC curve. The application of this methodology to the analysis of ATM traffic was presented by Rueda and Kinsner in 1997 (25).

The concept of generalized entropy developed by Hungarian mathematician Alfred Renyi in the 1960s, has been successfully applied in modeling traffic in packet networks, and this is the foundation for multifractal models. These models are also called entropy models. Renyi's work has led to interesting methodologies in image processing. The strength of the technique is in its ability to describe higher-order moments. The formal definition of multifractal processes is still under study, but it is clear that the technique generalizes self-similar processes; that it can be used as an alternative to variance–time analysis; and that it produces an estimate of the Hurst parameter as a special case.

Several measures, such as the correlation dimension and the Hausdorff dimension (or equivalently the Hurst parameter), have been used in the past for packet traffic. Multifractality is essentially a technique to calculate multiple fractal dimensions simultaneously based on interarrival times.

Central to fractal analysis is the concept of volume element (vel). In the case of multifractals for image processing, an object is covered by vels and a counting process is carried out to identify the densities of points in the vels. Multifractal analysis is based on the concept of generalized dimension D_q introduced by Renyi. The generalized dimension measure is defined as

$$D_q = \lim_{N\to\infty,\epsilon\to\infty} \frac{\log(\sum_{j=1}^{N_\epsilon} p_j^q)}{(q-1)\log(\epsilon)}$$

where for ATM traffic analysis, p_j is interpreted as the probability of interarrivals in the volume element j, and ϵ is the radius of the volume elements. The total number of interarrival times is $N = \sum_{j=1}^{N_\epsilon} n_j$, where n_j is the frequency of the volume element j. N_ϵ is the finite approximation to the number of points used. This generalized dimension concept is based on the assumption that the following power law holds:

$$\frac{1}{(q-1)} \sum_{j=1}^{N_\epsilon} p_j^q \sim \epsilon^{D_q}$$

Recalling that the Shannon entropy, which is a measure of disorder, is defined as

$$H_1 = -\sum_{j=1}^{N_\epsilon} p_j \log(p_j)$$

It is evident that the following definition provided by Renyi is more general:

$$H_q = \frac{1}{(q-1)} \sum_{j=1}^{N_\epsilon} p_j^q \qquad \text{for } q \in [-\infty, \infty]$$

The moment order is given by q. The definition of generalized dimension can be equivalently stated in terms of the generalized entropy as

$$D_q = \lim_{N_\epsilon \to \infty, \epsilon \to \infty} \frac{H_q}{\log(\epsilon)}$$

Other measures can be derived from the above definition for specific values of q. The Hausdorff dimension, for example, is obtained for $q = 0$ as $D_H = D_0 = E + 1 - H$, where E is the Euclidean dimension and H is the Hurst parameter. For $q = 1$, the information dimension can be obtained: $D_I = D_1$. For $q = 2$, the generalized dimension is the same as the correlation dimension: $D_C = D_2$. This dimension is defined using the pair correlation function C_k. These properties only hold for asymptotic self-similar fractals and for exact self-similar fractals $D_q = D_0$, the Hausdorff dimension for all q. The generalized dimension D_q is a monotonically decreasing function, and thus $D_i > D_j$ for $i < j$.

The Holder exponent, α_q, given by

$$\alpha_q = \frac{d}{dq}((q-1)D_q)$$

and the spectrum of singularities, f_q, given by

$$f_q = q\alpha_q - (q-1)D_q$$

are two functions that are useful in constructing an interesting graph called the multifractal spectrum based on the function $f_q(\alpha_q)$. The maximum value of the multifractal spectrum for a traffic trace yields the Hurst parameter.

Applications of multifractal techniques to ATM traffic analysis have produced interesting results. There are several issues to be addressed—for example, the meaning and effect of higher moments in terms of telecommunications systems.

The present methodology is based on the assumption that the probability of an arrival in a given volume element is the ratio between the count and the total number of elements. In terms of real-time implementation, the technique provides results in only one pass of the data stream. This motivates hardware implementations.

The information that can be obtained from the application of multifractal techniques to traffic characterization can potentially complement the descriptors defined in the standards, providing a very promising methodology for traffic characterization in packet networks.

BIBLIOGRAPHY

1. N. Jayant, Signal compression: Technology targets and research directions, *IEEE J. Selected Areas Commun.,* **10**: 796–818, 1992.
2. G. Held, *Data Compression: Techniques and Applications, Hardware and Software Considerations,* 2nd ed., New York: Wiley, 1987.
3. T. C. Bell, J. G. Cleary, and I. H. Witten, *Text Compression,* Englewood Cliffs, NJ: Prentice-Hall, 1990.
4. D. Dueck and W. Kinsner, Experimental study of Shannon–Fano, Huffman, Lempel–Ziv–Welch and other lossless algorithms, *Proc. 10th Comput. Netw. Conf.,* San Jose, CA, Sept. 29–30, 1991, pp. 23–31.
5. J. A. Storer (ed.), *Image and Text Compression,* Boston: Kluwer, 1992.
6. A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression,* Boston: Kluwer, 1992.
7. W. Kinsner, *Review of data compression methods, including Shannon–Fano, Huffman, Arithmetic, Storer, Lempel–Ziv–Welch, fractal, neural network, and wavelet algorithms,* Technical Report, DEL91-1, Jan. 1991.
8. T. W. Parsons, *Voice and Speech Processing,* New York: McGraw-Hill, 1986.
9. J. Watkinson, *Compression in Video and Audio,* Oxford: Focal Press, 1995.
10. A. Langi and W. Kinsner, Design and implementation of CELP speech processing system using TMS 320C30, *Proc. 10th Comput. Netw. Conf.,* San Jose, CA, Sept. 27–29, 1991, pp. 87–93.
11. A. Langi, K. Ferens, and W. Kinsner, A wavelet model of LPC excitation for speech signal compression, *9th Int. Conf. Math. Comput. Model. Rec.,* ICMCM'93, Berkeley, CA, July 26–29, 1993.
12. K. Ferens and W. Kinsner, Adaptive quantization in wavelet subbands for wideband audio signal compression, *9th Int. Conf. Math. Comput. Model. Rec.,* ICMCM'93, Berkeley, CA, July, 1993.
13. M. Ghanbari et al., *Principles of Performance Engineering for Telecommunication and Information Systems,* London: IEE, 1997.
14. I. Glover and P. Grant, *Digital Communications,* Upper Saddle River, NJ: Prentice-Hall, 1998.
15. W.B. Pennebaker and J. L. Mitchell, *JPEG Still Image Data Compression Standard,* New York: Van Nostrand Reinhold, 1993.
16. N. Jayant (ed.), *Signal Compression: Coding of Speech, Audio, Text, Image and Video,* Cleveland: World, 1997.
17. R. J. Bates and D. W. Gregory, *Voice and Data Communications Handbook,* New York: McGraw-Hill, 1998.
18. L. Torres and M. Kunt, *Video Coding: The Second Generation Approach,* Boston: Kluwer, 1996.
19. M. F. Barnsley and L. P. Hurd, *Fractal Image Compression,* Wellesley, MA: Peters, 1993.
20. L. Wall and W. Kinsner, A fractal block coding technique employing frequency sensitive competitive learning, *Proc. IEEE Commun., Comput. Power Conf.,* WESCANEX'93, Saskatoon, May 17–18, 1993, pp. 320–329.
21. S. C. Ahalt et al., Competitive learning algorithms for vector quantization, *Neural Netw.,* **3** (3): 277–290, 1990.
22. W. Willinger, M. S. Taqqu, and A. Erramilli, A Bibliographical Guide to Self-Similar Traffic and Performance Modeling for Modern High-Speed Networks, *Stochastic Networks: Theory and Application,* Oxford: Clarendon Press, 1996, pp. 339–366.
23. B. Melamed, An overview of TES processes and modeling methodology, in L. Donatiello and R. Nelson (eds.), *Performance Evaluation of Computer Communication Systems, Lecture Notes in Comput. Sci.,* New York: Springer-Verlag, 1993, pp. 359–393.

24. A. Erramilli, O. Narayan, and W. Willinger, Experimental queueing analysis of long-range dependent packet traffic, *IEEE/ ACM Trans. Netw.,* **4** (2): 209–223, 1996.

25. J. A. Rueda and W. Kinsner, Multifractal ATM traffic characterization, *Proc. Can. Conf. Broadband Res.,* Ottawa, Canada, April 1997, pp. 125–136.

Reading List

D. Bear, *Principles of Telecommunication Traffic Engineering,* London: Institution of Electrical Engineers, 1980.

Bellcore, *Broadband Switching System Generic Requirements* (GR-1110-CORE), Revision 4, NJ, October 1996.

R. L. Freeman, *Telecommunication System Engineering,* New York: Wiley, 1996.

J. Y. Hui, *Switching and TrafficTheory for Integrated Broadband Networks,* Boston: Kluwer, 1990.

W. Kinsner, New text, image, and sound compression techniques for electronic publications, *Proc. 1993 Intern. Conf. Refereed Electron. J.,* 1993, pp. 18.1–18.20.

W. E. Leland et al., On the self-similar nature of Ethernet traffic (extended version), *IEEE/ACM Trans. Netw.,* **2**: 1–15, 1994.

R. R. Martine, *Basic Traffic Analysis,* Englewood Cliffs, NJ: Prentice-Hall, 1994.

D. E. McDysan and D. L. Spohn, *ATM Theory and Application,* New York: McGraw-Hill, 1995.

J. A. Rueda and W. Kinsner, A survey of traffic characterization techniques in telecommunication networks, *Proc. IEEE Can. Conf. Elec. Comput. Eng.,* Calgary, Canada, May 1996, pp. 830–833.

JOSE A. RUEDA
Telecommunications Research Laboratories (TR*Labs*)
WITOLD KINSNER
University of Manitoba

TELECOM SWITCHES. See TELECOMMUNICATION EXCHANGES.