EXHIBIT 18

>>> <edward.reines@weil.com> 09/15/05 12:58AM >>>
Gentlemen:

I have three requests to enhance the coordination and conduct of discovery.

First, please confirm that defendants' may exchange with each other Telcordia's information designated under the protective order or the "attorneys eyes only" local rule so long as everyone involved complies with their relevant confidentiality obligations.

Second, please agree to serve all documents served in any case on the parties in all three cases. We will do the same.

Third, please agree to copy counsel for parties in all three cases on all case related corrospondance. Again, we will do the same.

We appreciate your cooperation.

Best,

Ed

The information contained in this email message is intended only for use of the individual or entity named above. If the reader of this message is not the intended recipient, or the employee or agent responsible to deliver it to the intended recipient, you are hereby notified that any dissemination, distribution or copying of this communication is strictly prohibited. If you have received this communication in error, please immediately notify us by email (postmaster@weil.com), and destroy the original message. Thank you

EXHIBIT 19

United States Patent [19]
Chao et al.

[11] **Patent Number: 4,893,306**
[45] **Date of Patent: Jan. 9, 1990**

[54] **METHOD AND APPARATUS FOR MULTIPLEXING CIRCUIT AND PACKET TRAFFIC**

[75] Inventors: **Hung-Hsiang J. Chao, Madison; Sang H. Lee, Bridgewater; Liang T. Wu, Gladstone, all of N.J.**

[73] Assignee: **Bell Communications Research, Inc., Livingston, N.J.**

[21] Appl. No.: **118,977**

[22] Filed: **Nov. 10, 1987**

[51] Int. Cl.[4] **H04J 3/16; H04J 3/26**
[52] U.S. Cl. **370/94.2; 370/84; 370/99; 370/112**
[58] Field of Search **370/94, 60, 84, 99, 370/111, 112, 82, 110.1, 89**

[56] **References Cited**

U.S. PATENT DOCUMENTS

4,321,703	3/1982	Schwärtz et al.	370/89
4,516,240	5/1985	Kume et al.	370/94
4,594,708	6/1986	Servel et al.	370/94
4,685,105	8/1987	Shikama et al.	370/89
4,706,246	11/1987	Kume	370/89
4,763,319	8/1988	Rozenblit	370/89
4,764,921	8/1988	Graves et al.	370/110.1
4,771,425	9/1988	Baran et al.	370/110.1

OTHER PUBLICATIONS

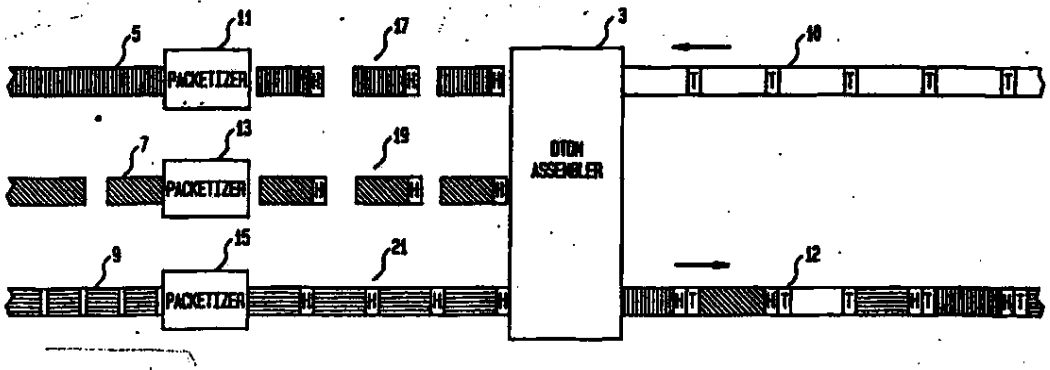
R. W. Muise, et al., "Experiments in Wideband Packet Technology", Proc. 1986, International Zurich Seminar on Digital Communications, pp. 136–138.
W. W. Chu, "A Study of Asynchronous Time Division Multiplexing for Time Sharing Computer Systems", Proc. AFIPS, vol. 35, pp. 669–678, 1969.
A. Thomas, et al., "Asynchronous Time Division Techniques: An Experimental Packet Network Integrating Video Communication", Proc. International Switching Symposium, May 1984.

Primary Examiner—Douglas W. Olms
Assistant Examiner—Min Jung
Attorney, Agent, or Firm—James W. Falk

[57] **ABSTRACT**

A data transmission technique referred to herein as Dynamic Time Division Multiplexing (DTDM) is disclosed along with a set of multiplexers and demultiplexers required to apply DTDM in an actual telecommunications network. The DTDM technique uses a transmission format which is compatible with the existing digital circuit transmission format and the packet transmission format so that DTDM is able to handle the transmission of circuit and packet traffic. Thus, DTDM provides a flexible migration strategy between present circuit networks and future broadband packet networks.

7 Claims, 10 Drawing Sheets



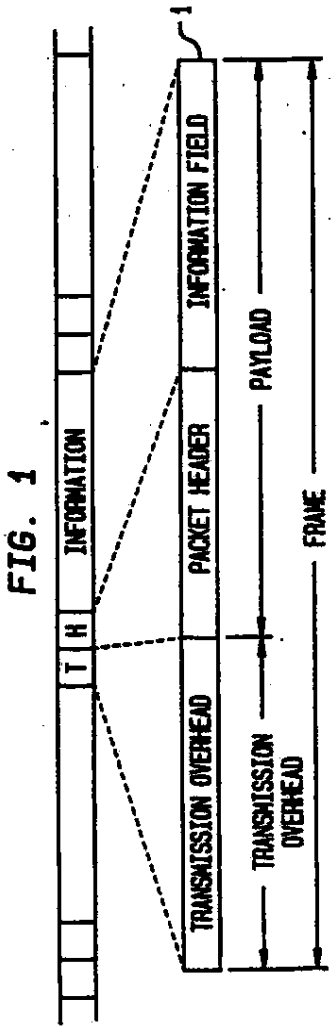

1
INFORMATION
T
H
INFORMATION FIELD
PACKET HEADER
TRANSMISSION OVERHEAD
PAYLOAD
TRANSMISSION OVERHEAD
FRAME

FIG. 1

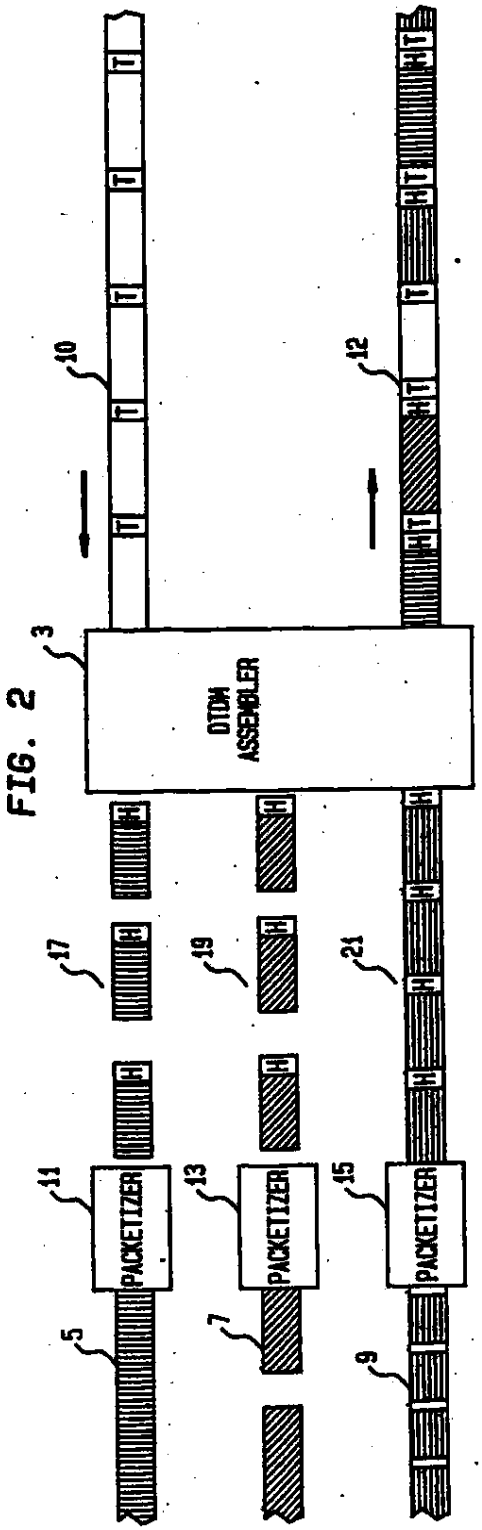
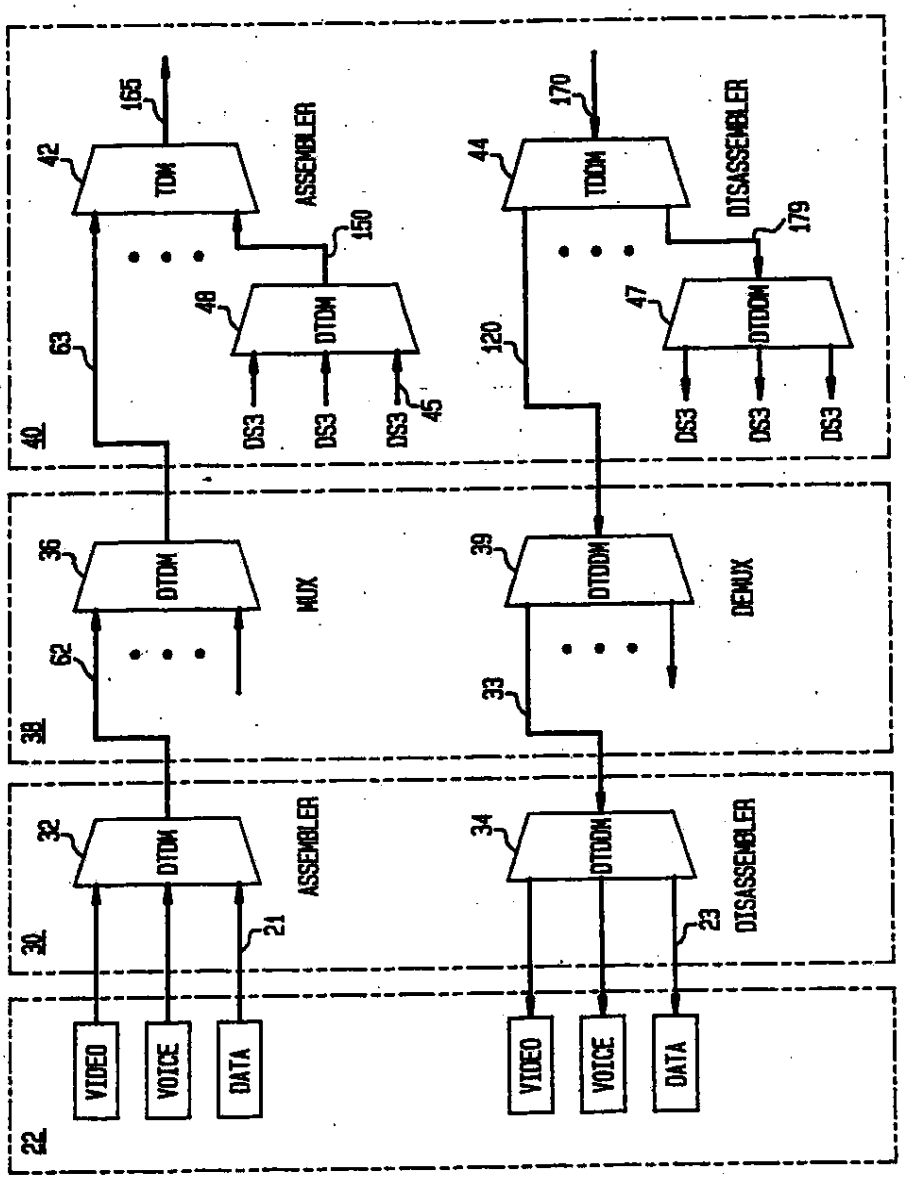

10
12
3
DTDM ASSEMBLER
17
19
21
11
13
15
PACKETIZER
PACKETIZER
PACKETIZER
5
7
9

FIG. 2

FIG. 3


20
165
42
TDM
ASSEMBLER
150
48
DTDM
63
DS3
DS3
DS3
45
40
170
44
TDM
DISASSEMBLER
179
47
DTDM
120
DS3
DS3
DS3
36
DTDM
MUX
62
38
39
DTDM
DEMUX
33
32
DTDM
ASSEMBLER
21
30
34
DTDM
DISASSEMBLER
23
VIDEO
VOICE
DATA
VIDEO
VOICE
DATA
22

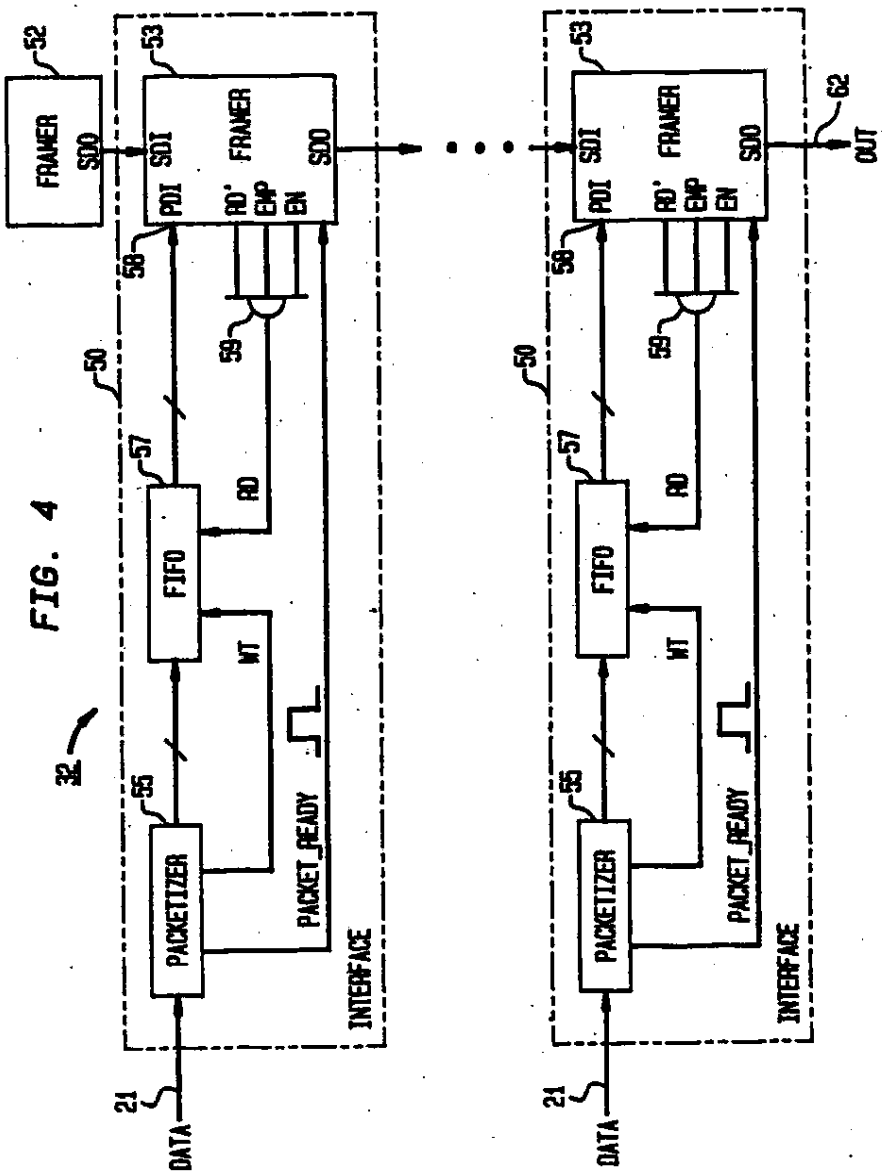

52
FRAMER
SDO
53
SDI
FRAMER
SDO
PDI
RD'
EMP
EN
58
59
50
57
FIFO
RD
WT
55
PACKETIZER
PACKET_READY
INTERFACE
21
DATA
62
OUT

FIG. 4

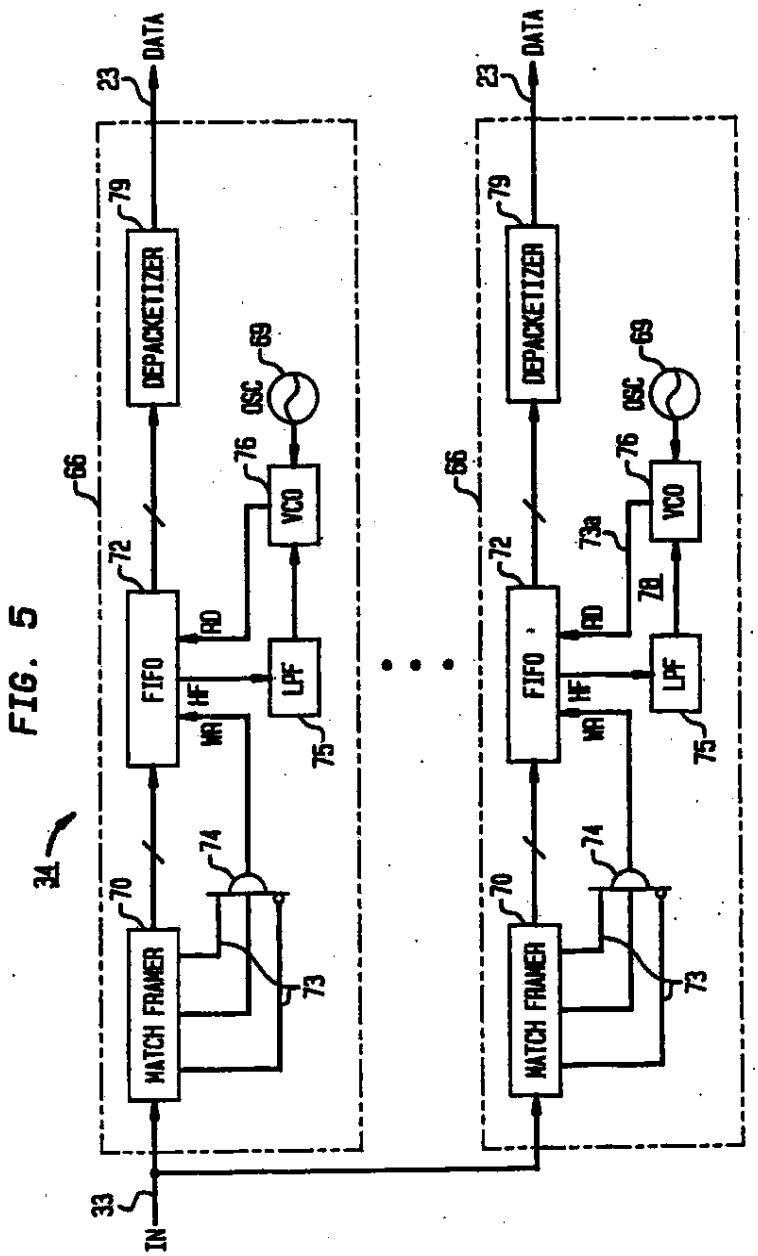

DATA
23
DEPACKETIZER
79
OSC
69
VCO
76
66
72
FIFO
RD
HF
WR
LPF
75
74
70
MATCH FRAMER
73
34
33
IN
73a
78

FIG. 5

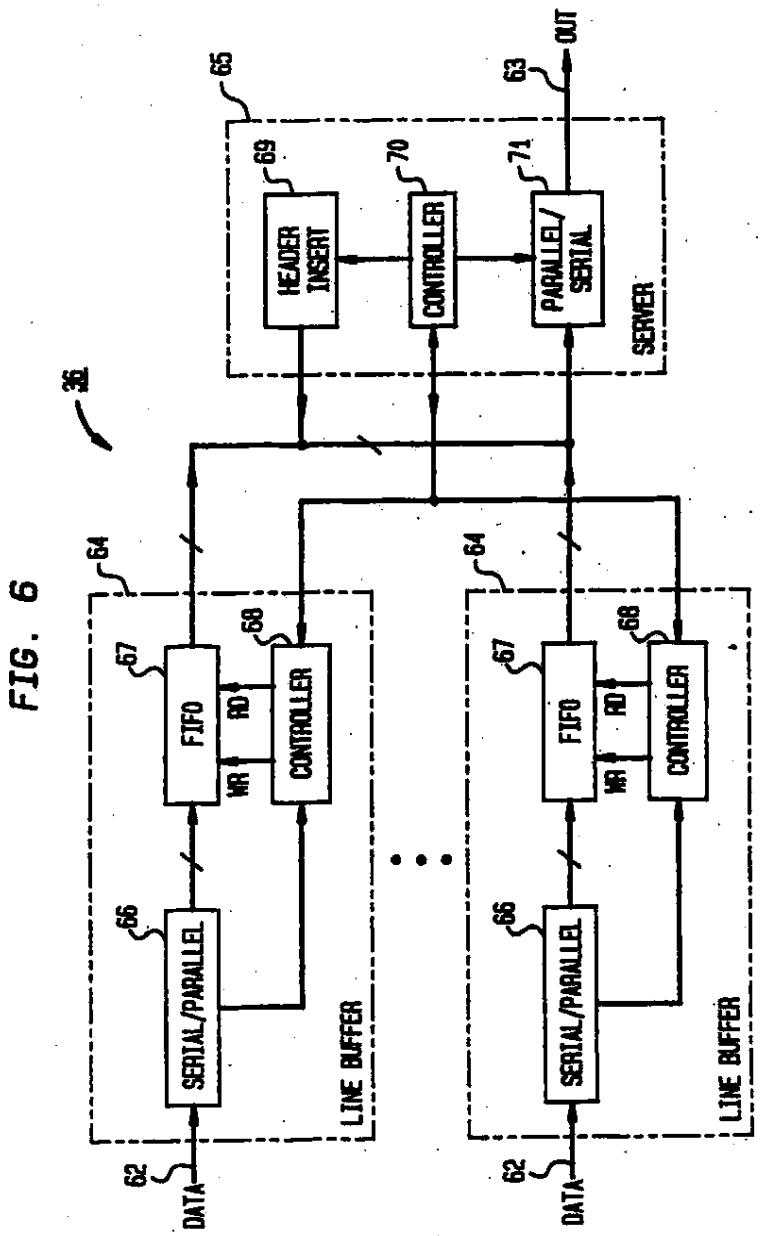

OUT
63
65
69
70
71
HEADER INSERT
CONTROLLER
PARALLEL/ SERIAL
SERVER
36
64
64
67
68
FIFO
RD
WR
CONTROLLER
66
SERIAL/PARALLEL
LINE BUFFER
62
DATA

FIG. 6

FIG. 7

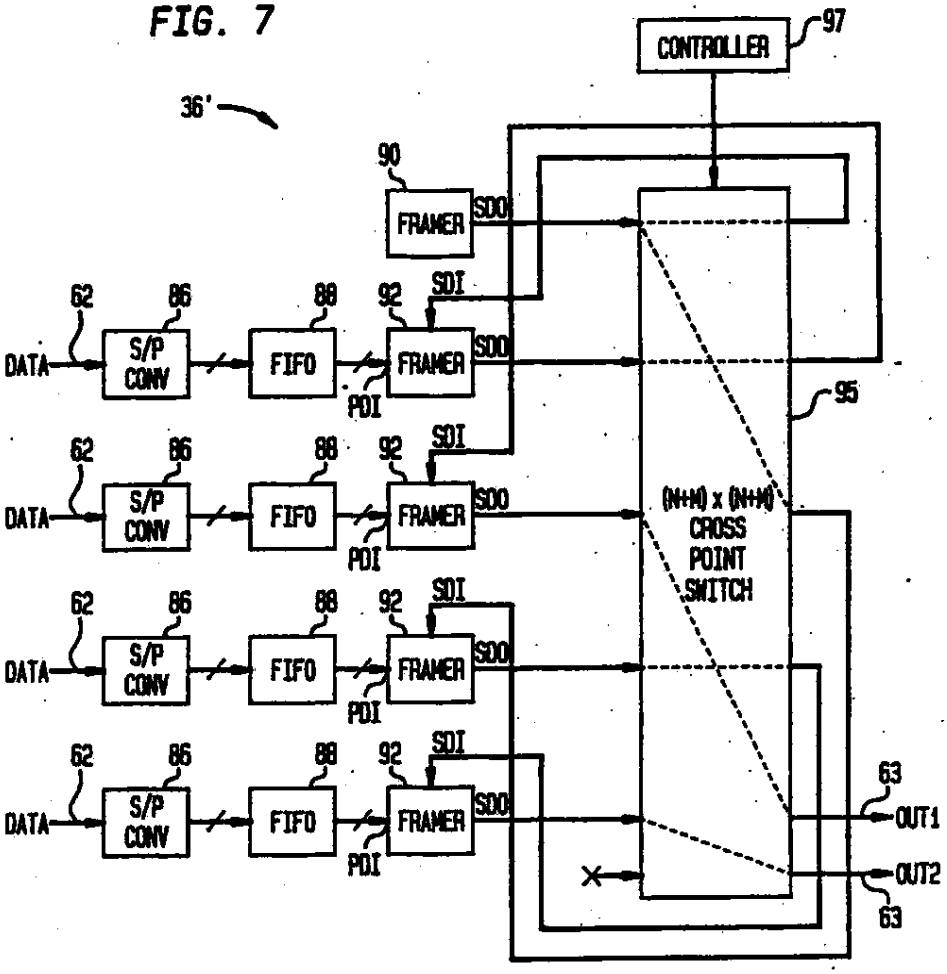
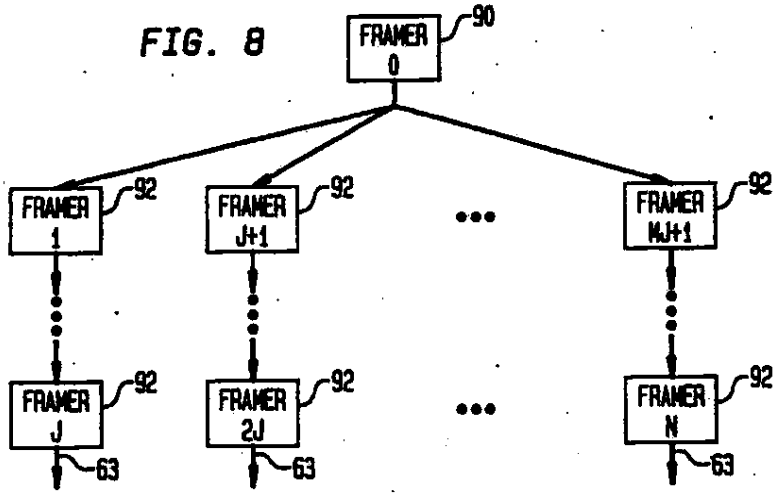

CONTROLLER
97
36'
90
FRAMER
SDO
SDI
62
86
88
92
DATA
S/P CONV
FIFO
FRAMER
PDI
95
(N+H) x (N+H) CROSS POINT SWITCH
63
OUT1
OUT2

FIG. 8


FRAMER 0
90
FRAMER 1
FRAMER J+1
FRAMER MJ+1
FRAMER J
FRAMER 2J
FRAMER N
92
63

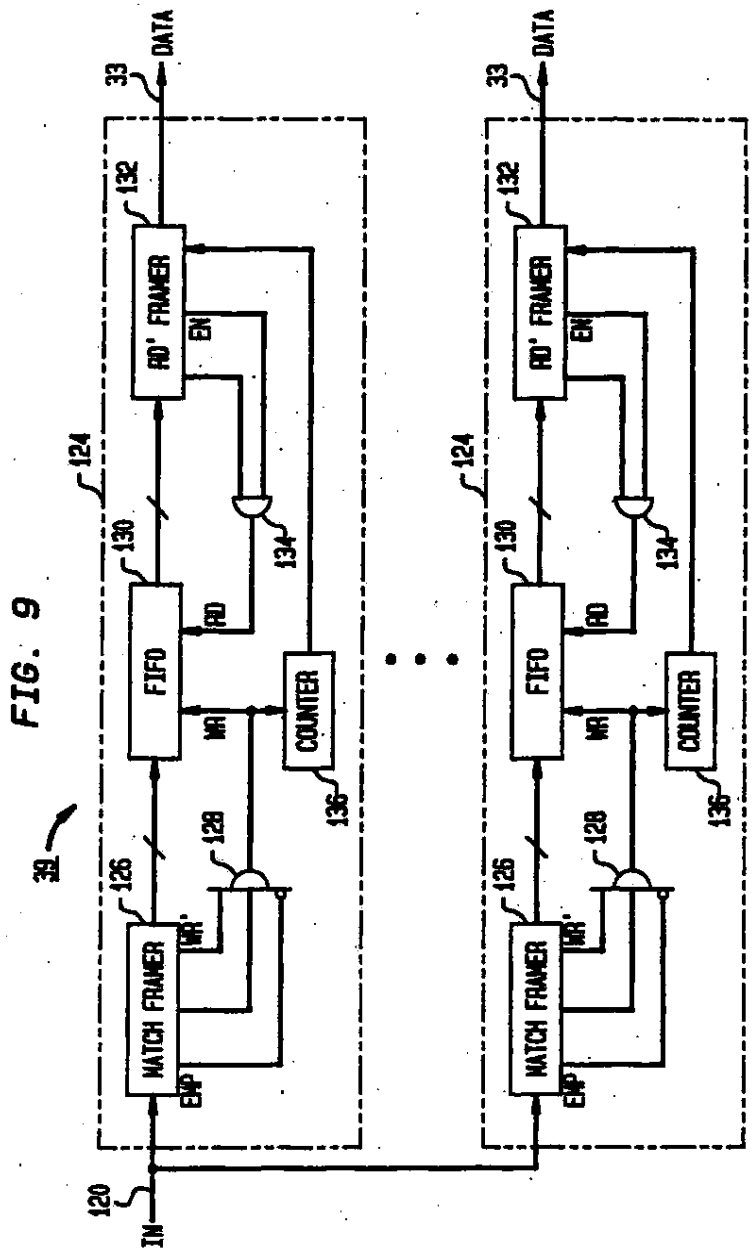

DATA
33
132
RD' FRAMER
EN
124
130
FIFO
RD
WR
134
COUNTER
136
128
126
WR'
MATCH FRAMER
EMP
120
IN
39

FIG. 9

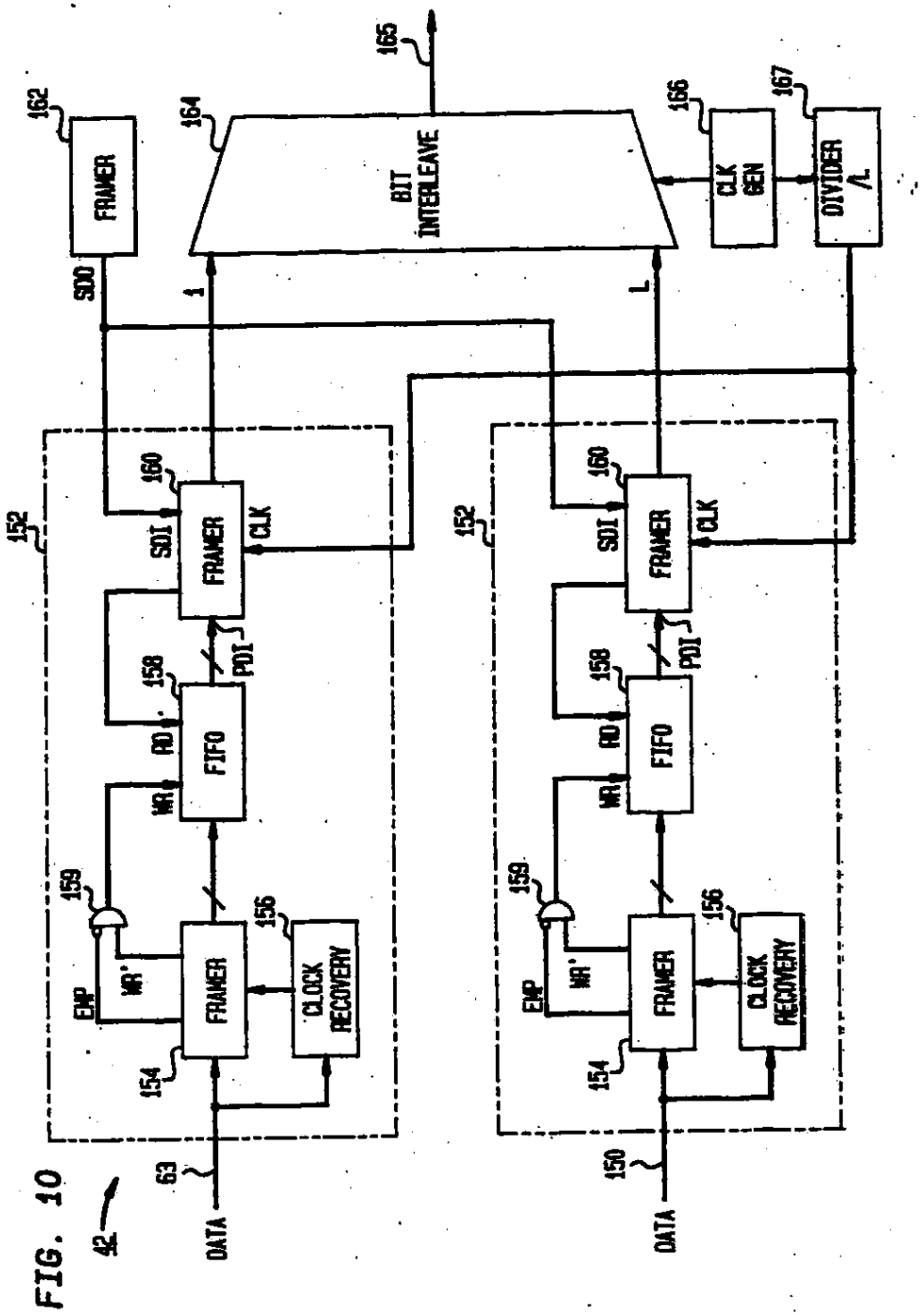

165
162
FRAMER
164
BIT INTERLEAVE
166
CLK GEN
167
DIVIDER N
SDO
1
L
152
160
SDI
FRAMER
CLK
158
PDI
RD
FIFO
WR
159
EMP
WR'
FRAMER
156
CLOCK RECOVERY
154
63
DATA
150
DATA

FIG. 10

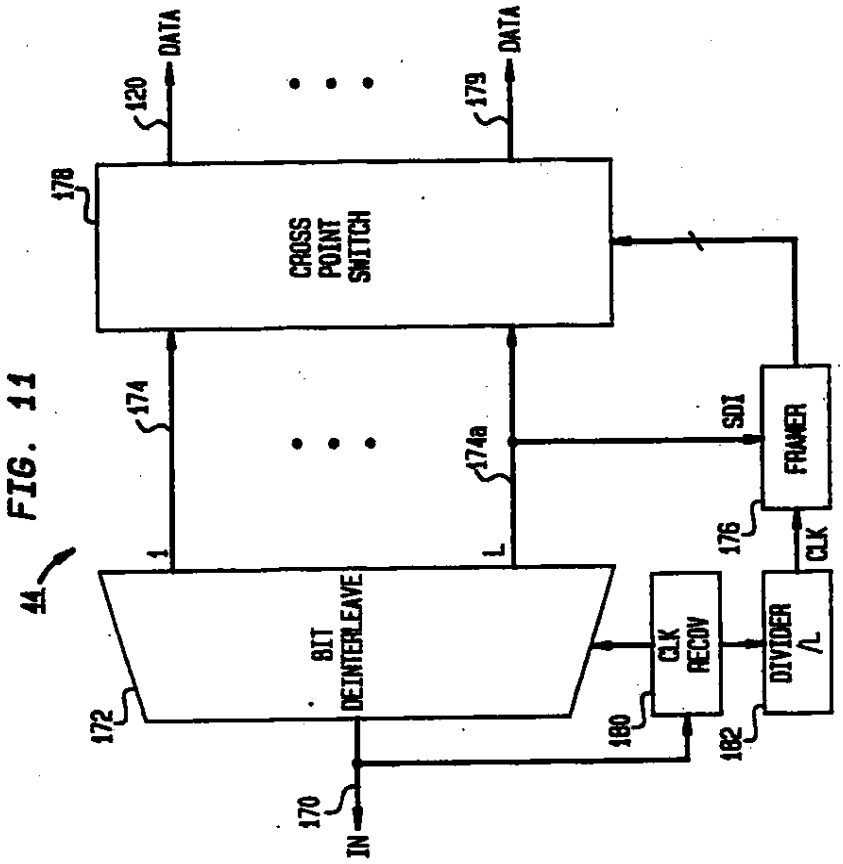

DATA
120
DATA
179
178
CROSS POINT SWITCH
174
174a
SDI
FRAMER
176
CLK
1
L
44
BIT DEINTERLEAVE
CLK RECOV
DIVIDER /L
172
180
182
170
IN

FIG. 11

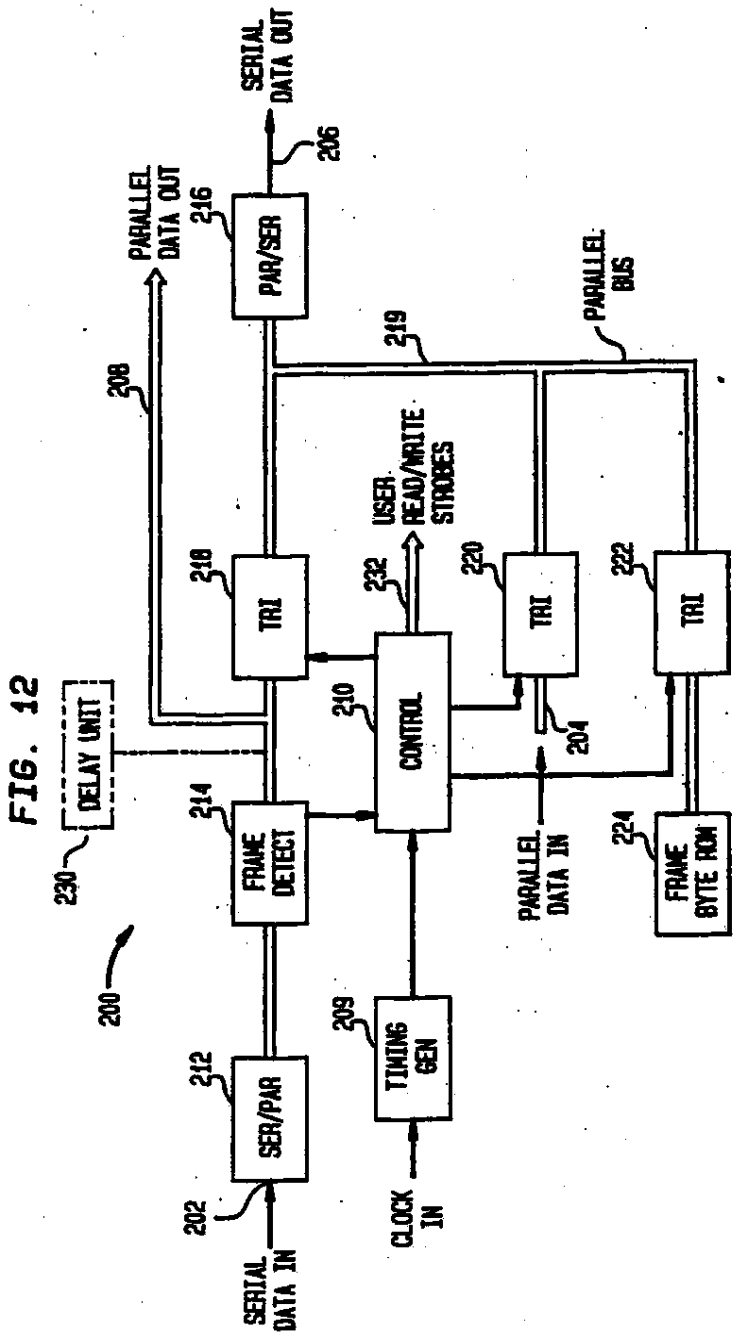

SERIAL DATA OUT
206
PAR/SER
216
PARALLEL DATA OUT
208
PARALLEL BUS
219
USER READ/WRITE STROBES
232
TRI
218
TRI
220
TRI
222
CONTROL
210
204
DELAY UNIT
230
FRAME DETECT
214
PARALLEL DATA IN
FRAME BYTE ROM
224
200
TIMING GEN
209
SER/PAR
212
202
CLOCK IN
SERIAL DATA IN

FIG. 12

METHOD AND APPARATUS FOR MULTIPLEXING CIRCUIT AND PACKET TRAFFIC

RELATED APPLICATIONS

The following applications contain subject matter related to the subject matter of the present application, are assigned to the assignee hereof and have been filed on the same date as the present application.

1. J. J. Chao, "DTDM Multiplexer With Cross-Point Switch", Ser. No. 118,979, now U.S. Pat. No. 4,855,999, issued Aug. 8, 1989
2. M. W. Beckner, F. D. Porter, K. Shu, "DTDM Multiplexing Circuitry", Ser. No. 118,897, now U.S. Pat. No. 4,833,671, issued May 23, 1989
3. H. J. Chao, S. H. Lee, "Time Division Multiplexer for DTDM Bit Streams", Ser. No. 118,978, now U.S. Pat. No. 4,833,673, issued May 23, 1989
4. M. W. Beckner, J. J. Chao, T. J. Robe, L. S. Smoot, "Framer Circuit", Ser. No. 118,898, now U.S. Pat. No. 4,819,226, issued Apr. 4, 1989.

FIELD OF THE INVENTION

This invention relates to the transmission of data in telecommunications networks. More particularly, the present invention relates to a data transmission technique referred to herein as Dynamic Time Division Multiplexing (DTDM), and a set of multiplexers and demultiplexers required to apply DTDM in an actual telecommunications network. DTDM is capable of effectively handling both circuit and packet traffic and thus provides a migration strategy between the present circuit switched telephone network and the future broadband packet switched network.

BACKGROUND OF THE INVENTION

Presently, there are significant uncertainties when it comes to predicting the future demand for broadband telecommunications services such as high definition video and interactive data communications. This uncertainty in the future demand for broadband telecommunications services has a significant impact on the design of public telephone networks. First, to satisfy the unknown growth pattern in future service demands, it is desirable to have a robust network design that can be easily modified in response to changes in demand for particular telecommunications services. Second, the network must be able to handle vastly different types of traffic ranging from low speed data and voice to full motion video. Third, depending on the demand for wideband services, a network design must be capable of providing a migration strategy from existing copper wires and circuit transmission and switching facilities to optical fibers and the succeeding generations of high speed packet transmission and switching facilities, which packet facilities are used in connection with the delivery of wideband telecommunications services. These three criteria determine the selection of the three major components of a network design: network topology, transmission systems and switching systems. Here, the concern is primarily with transmission systems and transmission techniques which meet the foregoing criteria.

Two important types of commercially used transmission systems are circuit systems and packet systems. Typically, circuit systems utilize time division multiplexing (TDM) as a transmission technique. When TDM is used, each data stream comprises frames which are subdivided into slots. Corresponding slots in each frame are allocated to specific connections. For example, the first slot in each frame is allocated to one specific connection and the second slot in each frame is allocated to a second connection, etc. Each frame also includes a field which contains transmission overhead information including frame synchronization words and control words. This traditional circuit transmission format can be extended to multiple bit rate services by allocating multiple slots in each frame to high bandwidth services. In such circuit transmission systems, a combination of space division switching and time division switching is utilized at the network switches to swap time slots between various bit streams so that connections to and between specific subscribers are established.

Historically, the first digital circuit transmission systems were introduced during the 1960's. These first digital circuit transmission systems were introduced in inter-office trunking applications to carry 24 voice channels by a single 1.544 Mb/sec digital stream. This is known as the DS-1 signal. Subsequently, the wide deployment of digital channel banks in the public telephone network required the multiplexing of several DS-1 signals into a higher speed bit stream to efficiently utilize available transmission links. As the network grew further, continuing efforts to effectively multiplex tributaries having different bit rates into a common bit stream resulted in the well-known hierarchical multiplexing plan comprising the DS-1 (1.544 Mb/sec), DS-1C (3.152 Mb/sec), DS-2 (6.312 Mbit/sec), DS-3 (44.736 Mb/sec) and DS4 274.176 Mb/sec signals.

Conventional circuit transmission systems suffer from a number of shortcomings. Perhaps the most important problem is the multiplexing hierarchy itself. An important result of the hierarchy is an inherent lack of flexibility. Since the network can only transmit the set of signals in the hierarchy, every telecommunications service has to meet the stringent interface requirement of given hierarchical signal bit rates, instead of the particular service being able to transmit at its own natural bit rate. Therefore, the packet mode of transmission which is inherently bit rate flexible is favored for future broadband networks which are to be adapted to deliver enhanced telecommunication services such as high definition video and interactive data communications.

In contrast with circuit transmission systems which transmit data in frames subdivided into slots, packet transmission systems transmit data in discrete blocks or packets, with each packet having an address header at the front thereof. At the network switches, packets are routed from a specific input line to a specific output line, based on address information contained in the packet header. In this way data packets can be routed from a particular subscriber location, through a telecommunications network, to another subscriber location. Packet transmission techniques and especially fast packet transmission techniques (see e.g., R. W. Muise et al., "Experiments in Wideband Packet Technology", Proc 1986 International Zurich Seminar on Digital Communications, pp. 136–138 are inherently bandwidth flexible (i.e. the number of packets generated by a given service per unit time is flexible) and thus are suitable for wideband enhanced communications services. Accordingly, it is desirable to introduce packet transmission technology

into the public telephone network, which up to now is based primarily on circuit transmission technology.

The commonly-held view as to how to introduce packet technology into the public network is to deploy a packet overlay network because the existing network is optimized for circuit transmission and is therefore incompatible with packet transmission techniques. Accordingly, many deployment strategies recommend constructing an overlay packet network for a set of wideband services and hope that the migration of new services to the packet overlay network will allow the existing circuit transmission network to be phased out slowly. The main advantage of a packet overlay network is the quick realization of an end-to-end network for new services. However, the approach requires a large initial capital investment and increases operational cost by requiring the management of multiple separate networks.

It is an object of the present invention to provide an alternate approach for introducing packet transmission technology into the public telephone network, which approach requires the replacement of existing transmission components but not the implementation of an entirely new network. Thus, it is an object of the invention to provide a digital data transmission system capable of handling both existing hierarchical circuit traffic and packet traffic.

With regard to the above-identified objects of the invention, it should be noted that recent advances in network switch designs have blurred the distinction between packet networks and circuit networks. A typical switch for use in a telecommunications network has three major components: control processor, switch interfaces, and interconnection network. The control processor handles call set-up and tear-down, maintenance and administrative functions. The switch interfaces convert transmission formats (i.e., the format data has when transmitted between switching nodes) to switch formats (i.e., the format data has when processed within switching nodes). The interconnection network routes information blocks from specific input lines to specific output lines of the switch. For the existing digital circuit systems used in the public telephone network, the information in a specific time slot on an incoming line is transferred, via the switch, to a specific time slot on an outgoing line. Thus, the interconnection network serves as a crossconnect for the incoming signals on a slot-by-slot basis.

It has recently been shown (see e.g., Day-Giacopelli-Huang-Wu, U.S. patent application Ser. No. 021,664 entitled Time Division Circuit Switch, filed on Mar. 4, 1987, now U.S. Pat. No. 4,782,474, issued Nov. 1, 1988, and assigned to the assignee hereof) that a switch for use in a circuit network can be built using a self-routing packet interconnection network. An example of such a self-routing packet network is the Batcher-banyan network. Based on the address headers associated with fixed sized packets, the Batcher-banyan network routes a plurality of packets in parallel to specific destination addresses (i.e., specific output lines) without internal collisions. Thus, to mimic the operation of the conventional time-space-time switches used in circuit networks, switch interfaces are provided which perform the time slot interchange function and which are able to insert headers in front of circuit slots to convert such slots into packets for routing through the self-routing interconnection network and able to remove headers from packets leaving the self-routing interconnection network to reconvert packets back into conventional circuit time-slot format.

In addition to circuit and packet transmission, another mode of digital transmission is known as Asynchronous Time Division Multiplexing (ATDM). See e.g., W. W. Chu "A Study of Asynchronous Time Division Multiplexing for Time Sharing Computer Systems" Proc AFIPS Vol 35, pp. 669–678, 1969 and A. Thomas et al. "Asynchronous Time Division Techniques: An Experimental Packet Network Integrating Video Communication" Proc International Switching Symposium, May 1984. ATDM is used in connection with continuous and bursty data traffic. ATDM uses channel identifiers with actual data to allow on-demand multiplexing of data from subscriber terminals with low channel utilization. The channel identifiers and associated data form time slots. However, ATDM is bit rate flexible since the appearance of packets can be asynchronous. Slot timing is obtained from a special synchronization pattern which is inserted into unused time slots. Since the synchronization pattern appears only in unused time slots, ATDM cannot be used to carry existing high speed hierarchical signals wherein the loading is close to one hundred percent.

In short, the situation is that the present public telephone network utilizes circuit transmission technology and the associated time division multiplexing transmission techniques, while future broadband services, the demand for which is presently uncertain, are best offered using packet transmission technology. It is therefore an object of the invention to provide a transmission system which is capable of integrating present circuit traffic with future packet traffic so as to provide a flexible migration strategy from the existing copper wire based circuit network to succeeding generations of high bandwidth packet transmission networks.

SUMMARY OF THE INVENTION

The digital network transport system of the present invention, referred to herein as Dynamic Time Division Multiplexing (DTDM), is a flexible network transport system capable of effectively handling both circuit and packet traffic. By combining conventional time division multiplexing techniques and packet transmission techniques, DTDM enables a flexible transition from the existing circuit type networks to future broadband packet transmission networks.

In a network utilizing DTDM, each transmission bit stream is divided into frames. These frames are the fundamental unit of data transport in DTDM. Each such frame comprises two fixed length fields: overhead and payload. The overhead field includes, for example, a frame alignment word for frame timing and the empty/full status of the frame. The payload field of each frame may be filled with a data packet including header or a slot from a circuit transmission stream. Before a slot from a circuit transmission stream can be inserted into the payload field of a DTDM frame, it must first be converted into a packet-like form with a header at its front. Viewed another way, each occupied DTDM frame comprises a transmission overhead field, a header field, and a data field. Thus, the DTDM transmission format is a combination of the circuit transmission format and the packet transmission format.

In the DTDM system, packet and circuit traffic can be multiplexed through the same multiplexer. Thus, such a multiplexer can have continuous circuit type tributaries and bursty packet tributaries. To multiplex

such diverse traffic, a train of DTDM frames with empty payload fields is generated. This train has a bit rate which defines a basic backbone transmission rate for the DTDM transmission system. Data in the form of packets or circuit slots with headers attached are inserted into the empty frames to form the DTDM bit stream.

An appropriate analogy is as follows. The stream of empty DTDM frames may be analogized to a train of empty freight cars. The empty freight cars are then filled with data from the various tributaries which may have been in circuit or packet format.

Illustratively, a DTDM multiplexer may be used to merge traffic from three different communications sources or tributaries into a single DTDM bit stream. These tributaries may be a digital phone generating 64 Kilobits/sec PCM voice, a graphics terminal sending bursty data at 1 Megabit/sec, and a circuit transmission stream operating at the DS3 rate of about 45 Megabits/-sec. Illustratively, the bit rate of the backbone DTDM bit stream is 150 Megabits which yields 144,000 frames per second given a 130-byte frame size. The available frames are shared by the three tributaries by giving higher priority to the circuit tributary, and allowing the voice and graphics tributaries to contend on a first-come, first-served basis. The circuit tributary seizes one out of every three empty frames passing by. Thus the regularity of the circuit transmission will be maintained throughout the DTDM transmission link. Illustratively, the voice source is packetized by accumulating up to 15 milliseconds worth of voice samples before inserting this information into an empty DTDM frame along with a header. In this case the voice tributary will on average seize one out of every 2,160 frames. Similarly, at a rate of 1 Megabit per second, the graphics tributary will fill one frame out of 150. In this way, three diverse data streams are multiplexed into a single bit stream.

As a second example, DTDM can be used as a replacement transmission technology to carry existing inter-office traffic. More specifically, consider the need to multiplex and transmit three hierarchical signals at the DS1, DS2, and DS3 rates, respectively, for point-to-point transmission between two offices. The traditional TDM approach would utilize a step-by-step hierarchical approach to multiplex and to subsequently demultiplex these signals. The conventional hierarchical multiplexing scheme requires line conditioning and synchronization circuitry at each level of the hierarchy as well as hardware for bit interleaving.

In contrast, using a DTDM multiplexer, time slots from each of the three signals would be inserted into the empty frames in a basic DTDM backbone signal. If the backbone signal is 150 megabits per second and comprises 144,000 frames per second, the DS3 signal would require one out of every three DTDM frames, the DS2 signal would require approximately one out of every twenty-one DTDM frames and the DS1 signal would require approximately one out of every eighty-four of the empty DTDM frames.

In an actual network, the above-described DTDM streams at the basic backbone bit rate generally contain empty frames; thus DTDM streams may be multiplexed into more densely populated DTDM bit streams at the same bit rate. These more densely populated basic backbone rate bit streams may then be multiplexed into higher bit rate streams for point-to-point inter-office transmission.

Details of the assemblers needed to form the basic DTDM bit streams, the disassemblers needed to disassemble the basic DTDM bit streams, and the set of multiplexers and demultiplexers needed to implement DTDM in an actual network are described in detail below along with a framer circuit which plays a significant role in particular implementations of the assemblers/disassemblers and multiplexers/demultiplexers.

BRIEF DESCRIPTION OF THE DRAWING

FIG. 1 schematically illustrates the DTDM transmission format, in accordance with an illustrative embodiment of the invention;

FIG. 2 schematically illustrates the formation of a backbone DTDM, bit stream, in accordance with an illustrative embodiment of the invention;

FIG. 3 schematically illustrates an end-to-end network using DTDM, in accordance with an illustrative embodiment of the invention;

FIG. 4 illustrates an assembler for combining diverse tributary data streams into a single DTDM stream, in accordance with an illustrative embodiment of the invention;

FIG. 5 illustrates a disassembler for separating a DTDM bit stream into diverse tributary data streams, in accordance with an illustrative embodiment of the invention;

FIG. 6 illustrates a multiplexer for combining a plurality of DTDM bit streams into a single more densely occupied DTDM bit stream having the same bit rate;

FIG. 7 illustrates an N:M multiplexer for combining a plurality of DTDM bit streams;

FIG. 8 illustrates how the input lines in the multiplexer of FIG. 7 are grouped;

FIG. 9 illustrates a demultiplexer for separating a densely occupied DTDM bit stream into a plurality of less densely occupied DTDM bit streams;

FIG. 10 illustrates a multiplexer for point-to-point transmission.

FIG. 11 illustrates a demultiplexer for use in connection with point-to-point transmission; and

FIG. 12 illustrates a framer circuit.

DETAILED DESCRIPTION

1. DTDM Transmission Format

DTDM is an approach to data transport which can handle both TDM hierarchical signals and packet traffic in a common integrated structure, while allowing complete bit rate flexibility. As illustrated in FIG. 1, the transmission bit stream is divided into frames 1. The DTDM frame is the fundamental unit of information transport in the DTDM transmission scheme. The frames come one after the other so as to form a continuous chain or train.

Each frame 1 comprises two fixed length fields designated transmission overhead (T) and payload in FIG. 1. Illustratively, each frame comprises 130 bytes with 10 bytes being allocated to the transmission overhead field. Typically, the bit rate of the DTDM bit stream illustrated in FIG. 1 is about 150 Megabits/sec. The following information may be available in the overhead field of every DTDM frame; frame alignment word for frame timing, empty/full status of the frame, and span identification.

As shown in FIG. 1, the payload field of each frame may be filled with a data packet including a header (H) or a slot from a circuit transmission stream. However,

before a slot from a circuit transmission stream can be inserted into the payload field of a DTDM frame, it must first be converted to packet-like form by the insertion of a header (H) at its front. Viewed another way, each occupied DTDM frame comprises a transmission overhead field, a header field, and an information field. Thus, the DTDM transmission format is a combination of the circuit transmission format and the packet transmission format. The packet header provides information such as channel number, line number, error detection, etc. In general, only the information required in every frame gets permanent bandwidth allocation in the transmission overhead field.

FIG. 2 schematically illustrates the formation of a DTDM bit stream. The DTDM bit stream assembler 3 can combine into a single bit stream both continuous circuit tributaries and bursty packet tributaries. Three such tributaries are illustrated in FIG. 2. They are: a digital phone tributary 5 generating 64 Kilobits/sec PCM voice, a tributary 7 from a graphics terminal sending bursty data at one megabit per second, and a circuit transmission stream 9 operating at the DS3 rate of about 45 Megabits/sec. Each of the three tributaries has a characteristic shading in FIG. 2 so that it is possible to follow how data from the three tributaries is combined to form the DTDM bit stream.

To multiplex such diverse traffic, a train 10 of DTDM frames with empty payload fields is generated. This train 10 has a bit rate which defines a basic backbone transmission rate for the DTDM system. Each of the frames in the train 10 has an occupied transmission overhead field (T).

Illustratively, the train of frames has a bit rate of about 150 Megabits per second and comprises 144K blocks/sec. The assembler 10 serves to insert data from the tributaries 5,7,9 into the payload fields of the DTDM frames in the stream 10. To accomplish this, the tributaries 5, 7, 9 are first packetized using packetizers 11, 13, 15, respectively to form the packetized streams 17, 19, 21. Each packet comprises a header (H) and an information field. In the case of the tributary 5, up to 15 milliseconds of speech samples are accumulated to form a packet. In the case of the circuit tributary each slot is converted to packet form by placing a header at the front thereof.

To form the DTDM stream 10, the packets comprising the streams 17, 19, 21 are inserted into the empty payload fields of the empty frames in the stream 10. The empty frames are shared by the three tributaries by giving higher priority to the circuit tributary 9 and allowing the voice and graphics tributaries 5, 7, to contend for empty frames on a first-come, first-served basis. Thus, the circuit tributary seizes one out of every three frames so that the regularity of the circuit transmission is maintained throughout the DTDM transmission link. Similarly, the voice tributary will seize one out of every 2,160 frames and the graphics tributary will seize on average one out of every 150 frames. It should be noted that the bit stream 12 is not 100% occupied and that some frames remain empty. In this way, three diverse tributaries are multiplexed into a single DTDM bit stream.

2. A Network Utilizing DTDM

FIG.3 schematically illustrates an end-to-end network 20 utilizing DTDM. The network 20 connects to customer premises equipment (CPE) 22, of which three types are illustrated, namely video, voice and data.

In the network 20, three multiplexing stages are required to support end-to-end transport. In the user-network interface stage 30, an assembler 32 receives data streams on lines 21 from the customer premises equipment 22 and combines these streams into a basic backbone DTDM stream of the type discussed in connection with FIGS. 1 and 2. Similarly, disassembler 34 tears apart a basic DTDM bit stream arriving on line 33 and distributes the data to the appropriate customer premises equipment 22 via lines 23.

As indicated above, the DTDM bit stream formed by the assembler 32 is not 100% occupied. Thus the multiplexer 36 in the remote electronics stage 38 is used to combine several DTDM bit streams arriving on lines 62 into a more densely occupied DTDM bit stream of the same bit rate to achieve greater transmission efficiency. Similarly, the demultiplexer 39 separates a densely populated DTDM bit stream arriving on line 120 into less densely populated DTDM bit streams transmitted via lines 33, so that the data contained therein can ultimately be routed to the correct customer premises equipment.

In the point-to-point stage 40, a plurality of DTDM bit streams arriving via lines 63, 150 are time division multiplexed by means of time division multiplexer 42 for high speed point-to-point transmission via line 165 to a network switch (not shown). For example, the multiplexer 42 receives one DTDM stream via a line 63 from multiplexer 36 and another DTDM stream via line 150. The DTDM bit stream transmitted via line 180 is formed by DTDM assembler 43 and contains the data of three DS3 tributaries 45.

Time division demultiplexer 44 receives a high speed bit stream from a switch (not shown) via line 170 and demultiplexes this stream into a plurality of DTDM streams. One DTDM stream containing data for customer premises equipment goes to demultiplexer 39 via line 120 and another DTDM stream comprising DS3 slots goes to disassembler 47 via line 179.

3. DTDM Assembler and Disassembler

The function of the DTDM bit stream assembler 32 of FIG. 3 is to packetize each incoming data stream associated with one particular customer service or transmission channel and then embed these packets into the basic DTDM transmission frames. The assembler 32 is shown in greater detail in FIG. 4.

The assembler 32 comprises a plurality of interface units 50. Each interface unit 50 serves to interface an associated data input 21 with the DTDM bit stream. A DTDM bit stream comprising empty frames with empty payload fields is generated by framer unit 52. A detailed description of the framer unit is provided below.

Each interface unit includes a framer unit 53. The framer units 52, 53 are connected together in a daisy chain fashion. The frames comprising the DTDM bit stream are passed along the daisy chain from one framer unit to the next. More particularly, the DTDM bit stream leaves the serial data output (sdo) of the framer unit 52 and enters the serial data input (sdi) of the topmost framer unit 53. The DTDM bit stream leaves the topmost framer 53 via its serial data output (sdo). The DTDM bit stream then enters the serial data input (sdi) of each succeeding framer unit and leaves via the serial data output (sdo) of each framer unit. The DTDM bit stream leaves the serial data output of the lowermost framer via line 62. As shown in FIG. 3, line 62 serves to

transmit the DTDM bit stream to the DTDM multiplexer 36. If the DTDM frame currently located at the framer unit 53 of a particular interface 50 is empty, that interface may insert a packet into the payload field of the DTDM frame.

The data inputs 21 to the assembler 32 are connected to the customer premises equipment 22 of FIG. 3 and may have a wide range of bit rates; for example, the data inputs 21 can be video, voice, data, or different digital hierarchical transmission signals (DS-1, DS-2 and DS-3). Therefore, the assembler architecture must be capable of efficiently accommodating different input bit rates and be flexible enough to allow for future expansion or for the changing of particular input connections to different services. The architecture shown in FIG. 3 provides the capability to easily add or drop a particular input service.

Each input 21 is connected to a packetizer 55 which forms part of the associated interface unit 50. The packetizer 55 puts the incoming data into a packet structure by adding a packet header at the beginning of appropriate segments of the input bit stream. The packet header carries information about the packet, such as packet occupancy, channel identification number, line identification number, check sum and so on. Illustratively, the channel identification number is used to identify the input service from which the packet originated. After the data is put into a packet structure, it is stored in a FIFO 57 with byte wide format. The framer unit 53 then reads the data from the FIFO 57 into its parallel data input (pdi) 58 and generates properly framed data bits which are inserted into an empty payload field of a DTDM frame currently at the particular framer unit 53.

However, a framer unit 53 will not read the data from the FIFO 57 unless two conditions are met. One is that the "packet-ready" pulse signal from the packetizer 55 is asserted, indicating one packet is completely stored in the FIFO. The other condition is that the incoming DTDM frame on the serial data input (sdi) of the framer 53 is not already occupied by a valid packet, i.e. the incoming DTDM frame is empty. Thus, an empty or "emp" signal is transmitted from the framer 53. The "packet-ready" signal triggers an enable signal, "en", in the framer unit to be asserted for the whole frame transmission period allowing the data packet to be moved from the FIFO 57 through the framer 53 and into the DTDM bit stream. Using the "emp" and "en" signals, control logic 59 controls the reading of a packet out of the FIFO 57 and into the framer 53.

Since the framer units 53 are daisy-chained together, the contention for empty DTDM frames is automatically resolved in favor of input services having positions closer to the empty frame generator.

In order to simplify the assembler 32 of FIGS. 3 and 4 and hence reduce the building cost, one practical assumption may be utilized; the total traffic of all inputs at any given time is less than the bit rate of the basic backbone DTDM stream.

The topmost framer 52 in FIG. 4 does not have any input service connected to it. It generates the chain of empty DTDM frames which are sent to the following framers 53. If none of the interfaces 50 insert a packet into a particular frame, an empty frame is finally sent out through the serial data output (sdo) of the bottommost framer unit on lead 62.

After the DTDM bit stream has traveled through the entire communications network 20 of FIG. 3, which network includes multiplexers, switches, and demultiplexers, etc., the DTDM bit stream is disassembled and the data distributed to the appropriate customer services equipment. In the network 20 of FIG. 3, the disassembler 34 is used for this purpose. The disassembler 34 is shown in greater detail in FIG. 5. Illustratively, the disassembler 34 removes both the transmission overhead and packet header field from each incoming DTDM frame and distributes the data contained in the frame to the desired customer premises device.

More particularly, the assembler 34 comprises a plurality of interfaces 66. Each interface 66 receives the incoming DTDM bit stream via line 33 (see FIG. 3) and is illustratively connected to one customer premises device via an output lines 23 (see FIG. 3). Each incoming DTDM frame is simultaneously received by the framer unit 70 in each interface 66. However, only packets containing data to be transmitted to the associated customer premises equipment are transferred from the framer 70 to the associated FIFO 72. To accomplish this, the packet occupancy and channel identification number are examined by the framer 70. The framer 70 in turn generates proper control signals via lines 73, which, along with control logic 74, determine whether or not the packet carried in the payload field of the particular DTDM frame will be written into the FIFO 72 of the particular interface unit so that the data contained in the packet can be transmitted to the associated customer premises equipment.

Recovering the correct frequency from the incoming data is a very challenging task. Although for each kind of customer premises equipment or service the frequency is known, the difference between the local reading clock used to read data out of the FIFO 72 and the clock which was used to load data into empty frames at the transmit end may result in overflow or underflow of the FIFO 72. Illustratively, a phase locked loop 78 is used to modify the local reading clock in order to cancel this difference in clock rates.

As shown in FIG. 5, the local reading clock signal (line 73*a*) used to read data out of the FIFO 72 is phase locked with the incoming data so that the data can be read out correctly from the FIFO 72 without overreading or underreading. The rate at which data is read out of the FIFO is determined by the frequency of the voltage controlled oscillator 76 in the phase locked loop 78.

The packet is written into the FIFO 72 with the network clock rate, but read out at a rate dependent on the particular equipment to which the data is transmitted. An "hf" signal which indicates that the FIFO 72 is half full is smoothed out by a low-pass filter 75 whose output is used to control the output frequency of a voltage-controlled oscillator 76. If information is read out of the FIFO 72 faster than information is written into the FIFO 72, then the "hf" signal will not be asserted. This causes the voltage output from the low-pass filter 75 to decrease, reducing the output frequency produced by the voltage controlled oscillator and reducing the rate at which data is read out of the FIFO 72. Similarly, if information is read out of the FIFO more slowly than it is being written into the FIFO 72, the "hf" signal will be asserted and the voltage controlled oscillator frequency will be increased so that the read clock signal frequency is larger. The same interface unit 66 can be used for different customer premises devices by choosing a proper frequency for oscillator 69.

Data packets read out of FIFO units 72 are depacketized by means of depacketizer circuits 79 which serve to remove the headers. The resulting data is then trans-

mitted via lines 23 to the appropriate customer premises equipment.

4. DTDM Bit stream Multiplexer and Demultiplexer

The function of the DTDM bit stream multiplexer 36 of FIG. 3 is to concentrate a plurality of relatively sparsely occupied incoming DTDM streams into at least one more densely occupied DTDM stream of the same bit rate, resulting in more efficient use of the transmission facility. There is more than one architecture for implementing the DTDM multiplexer 36 of FIG. 3.

One embodiment of such a DTDM bit stream multiplexer is illustrated in FIG. 6. The DTDM multiplexer 36 of FIG. 6 comprises N input lines 62 (see FIG. 3) and one output line 63 (see FIG. 3). Line buffers 64 recognize and queue incoming DTDM frames. The server 65 looks for newly arrived DTDM frames in the line buffers 64, adds a proper line number in the header field, and sends the frames out in a more densely occupied DTDM bit stream.

The primary functions of the line buffers 64 are recognition and queuing of incoming DTDM frames. Each line buffer contains a serial/parallel converter 66 for converting incoming serial DTDM frames into parallel form and a first-in, first-out buffer 67 with capacity for multiple frames. A timing and control circuit 68 operates the line buffer and interfaces it with the server.

The main functions of the server 65 are to look for newly arrived DTDM frames in the line buffers, to modify the header field to include a line number, and to place the DTDM frame in a more densely occupied DTDM bit stream. The server comprises a header insert circuit 69 for modifying the header field of the DTDM frames, a controller circuit 70 for interfacing with the line buffers 64, and a parallel to serial converter 70. The operations of the server are pipelined; while the server reads a DTDM frame from a line buffer and places it in an outgoing DTDM stream, it continues searching line buffers for DTDM frames. It should be noted that the multiplexer of FIG. 6 is useful for multiplexing packets in non-DTDM transmission formats in addition to being useful for DTDM bit streams.

Another possible architecture for a multiplexer capable of combining several relatively sparsely occupied DTDM bit streams into a more densely occupied DTDM bit stream of the same bit rate builds on the architecture of the DTDM bit stream assembler 32 of FIG. 4. Each input to an interface unit 50 of FIG. 4 is replaced by a serial data link on which a DTDM bit stream arrives. The data packets contained in the frames comprising the incoming DTDM bit stream contend for output frames in an outgoing DTDM bit stream. The frames comprising the outgoing DTDM bit streams are generated by the framer 52 and passed along the chain of interconnected framers 53. The interface units 50 insert data packets from incoming DTDM frames into the frames of the outgoing bit stream to form a more densely occupied DTDM bit stream. The contention for output frames is resolved automatically by the daisy-chained connection of the framer units. Note that no packetizer is needed in the interface units, and the length of each FIFO is preferably more than two frames to prevent data packets contained in incoming frames from being lost.

FIG. 7 schematically illustrates an alternative DTDM bit stream multiplexer for combining a plurality of relatively sparsely occupied DTDM bit streams into a smaller number of more densely populated DTDM bit streams of the same bit rate. The multiplexer 36' of FIG. 7 has the flexibility to receive N input DTDM bit streams and to transmit M output DTDM bit streams, which allows M output lines to be shared by N input lines. It is known that both the probability of buffer overflow and the average delay for bursty traffic can be significantly decreased by increasing the number of outputs.

Using the multiplexer architecture 36 shown in FIG. 6, it is difficult to build an N:M multiplexer, because the service order is determined by a single central server. However, it is possible to provide a multiplexer system comprising M separate multiplexers of the type shown in FIG. 6, each having N/M input lines and line buffers and one server and associated output line. In contrast, the service order in the multiplexer 36' of FIG. 7 is determined locally, which results in the flexibility of reassigning input lines to different output lines based on the input traffic statistics.

The N:M multiplexer 36' of FIG. 7 comprises a plurality of input lines 62 (see FIG. 3) and a smaller number of output lines 63 (see FIG. 3). DTDM frames arriving on the input lines 62 are converted into a byte wide stream by means of the serial-to-parallel converters 86 and stored in the associated buffers (FIFOs) 88. The operation of the framer units 90, 92 is similar to those in the DTDM bit stream assembler of FIG. 3. Each framer 90, 92 has a parallel data input (pdi), a serial data input (sdi) and a serial data output (sdo). Framer 90, the head-end framer unit, doesn't have any input lines connected to it. In normal operation, it continuously sends out a chain of empty frames. The remaining framers 92 take data comprising occupied DTDM frames in the buffers 88, and insert this data into the empty frames generated by the framer 90, so as to combine a plurality of sparsely occupied DTDM bit streams into a smaller number of more densely populated DTDM bit streams.

The N:M multiplexer 36' of FIG. 7 comprises an $(N+M)\times(N+M)$ broadcasting cross point switch network 95. The serial data output (sdo) of each framer 90, 92 is connected to an input of the switch, and the serial data input (sdi) of each framer 9 is connected to an output of the switch as shown in FIG. 7. The connections through the switch network are controlled by a dedicated controller 97.

Illustratively, when the system is initialized, the N input lines 62 are divided into M groups, $(1,2\ldots,J)$, $(J+1,J+2,\ldots,2J),\ldots,(MJ+1,\ldots,(M+1)J)$, where $J=N \bmod M$. The J input lines in each group are logically connected as shown in FIG. 8. Each group of input lines is associated with one output line. Thus, all of the DTDM frames arriving at the inputs of one group are merged into a single DTDM bit stream which leaves via the associated output. The topmost framer unit 92 in each group receives empty frames broadcast from the framer 90. Each frame is then passed through the switch 95 from one framer in the group to the next framer in the group. If a particular FIFO 88 has data comprising a DTDM frame and the associated framer 92 receives an empty frame, the data is inserted into the empty frame. Thus, within each group service priority is ranked in descending order with the higher priorities near the top. Ultimately, M relatively densely occupied DTDM bit streams leave the multiplexer of FIG. 6 via the outputs 63.

Thus, with the addition of the cross point switch, more than one framer 92 receives empty frames from the framer 90 at the same time. This achieves the N:M

multiplexing function automatically and with minimal complexity.

If the input lines are not grouped so as to distribute output traffic evenly, the input lines can be regrouped easily by changing the connections within the switching network 95. For example, a particular input in the first group of inputs may be assigned to any other group, e.g., the second group of inputs, to spread out traffic evenly. The controller 97 must know the traffic statistics of each input line and follow some algorithm to rearrange the inputs and decide the ordering (priority) within each group.

The DTDM multiplexer of FIG. 7 may route DTDM frames arriving on the same input line to different output lines. For example, n frames arriving on an input line have been sent to output #1. But the (n+1)th frame may be switched to output #2 because reconfiguration took place to balance traffic among the output lines. This may cause an out of sequence problem if the (n+1)th frame arrives at the receive end before the nth frame does. The cost to reorder the frame sequence at the output end may be high. Illustratively, to avoid this problem, one rule may be followed: the input lines carrying services with high bit rate information, such as video, will not be switched from one input line group to another during the service period. For a low bit rate service, such as voice at 64 Kb/s, even if two consecutive frames containing data are dispatched onto two different output lines, the two frames from such a bursty service will be separated by more than several hundred frame intervals. Hence, it is unlikely for there to be an out of sequence problem in this case.

It should be noted that multiplexer architecture of FIG. 7 may be used to multiplex other types of traffic besides DTDM traffic. For example, streams of data packets may be multiplexed together to form more densely occupied streams.

Turning now to FIG. 9, the DTDM bit stream demultiplexer 39 of FIG. 3 is illustrated in greater detail. The function of the DTDM bit stream demultiplexer 39 is to separate a relatively densely occupied incoming DTDM bit stream into a plurality of relatively sparsely occupied outgoing DTDM bit streams of the same bit rate so that the user data in the frames can ultimately be transported to the proper customer premises devices.

The demultiplexer 39 of FIG. 9 has one input line 120 (see FIG. 3) and a plurality of output lines 33 (see FIG.3). Each output line 33 is connected to the input line 120 by means of an associated interface 124. Any incoming DTDM frame is simultaneously received by the framer unit 126 in each interface 124. The frame occupancy and line identification number of each incoming DTDM frame are examined by the framers 126. If the frame is not empty and the line number is matched, the packet contained therein will be written into the FIFO 130 under the control of logic 128 and then read out of the FIFO 130 by the framer 132 at the output end of the interface 124 under the control of logic 134. Otherwise, the packet is simply discarded. In this manner, the data from each incoming frame is routed to the correct output line. A counter 136 in each interface is used to count the number of bytes written into the FIFO 130 and generates a signal when a full packet is stored in the FIFO. This signal will inform the output framer 132 to start reading the packet in the FIFO. The framer 132 will assert the "en" signal during the reading of the entire packet. The framers 132 generate sequences of DTDM frames. These sequences of frames leave the framer 132 via the serial data outputs and form the outgoing DTDM bit streams on the lines 33. When data packets are present in the FIFOs 130 they are inserted into the frames generated by the framers 132. In particular embodiments of the demultiplexer, the functions of the framer units 126, 132 may be performed by a single framer unit.

5. Time Division MUX/DEMUX for DTDM Bit Stream

After relatively sparse DTDM bit streams are concentrated into more densely populated DTDM bit streams of the same bit rate using for example, the DTDM multiplexer 36 of FIG. 3, a plurality of such more densely populated bit streams may be time division multiplexed into a higher speed data stream using, for example, the time division multiplexer 42 of FIG. 3. Such high speed data channels may be used for communications to and from central offices.

Usually, the most challenging work in a time division multiplexing system is to synchronize all incoming bit streams so that they have a common bit rate before they are interleaved into a higher bit rate stream. Typically, the input bit streams have the same nominal center frequency but drift independently a small amount from the center frequency. The conventional way to overcome the asynchronization among the input bit streams is positive bit or byte stuffing. The frequency of the high speed output bit stream is made greater than the product of the nominal center frequency and the number of input tributaries. There is usually a bit or byte position reserved for the occasional stuffing of a dummy bit or byte. Also, there is some control overhead used to indicate if the bit or byte at the stuffing position is valid.

By taking advantage of the fact that the frames comprising each input DTDM bit streams are not 100% occupied, the frequency of the higher speed output bit stream can be made exactly equal to the nominal center frequency of the input tributaries times the number of the input tributaries. In the case of a DTDM system, this can be accomplished through the positive and negative stuffing of DTDM frames. Since the frequency of each input tributary signal can be adjusted in the positive or negative direction through the insertion or removal of an empty DTDM frame, it is possible to make the frequency of the high-speed multiplexed bit stream exactly an integer multiple of the nominal input tributary frequency.

A time division multiplexer 42 (see FIG. 3) for multiplexing a plurality of DTDM bit streams is illustrated in FIG. 10. Each input 63, 150 (see FIG. 3) is connected to an interface unit 152. Each interface unit 152 comprises a framer 154 which is clocked by a clock signal derived from a clock recovery circuit 156. The derived clock, which is the actual frequency of the tributary, may differ slightly from the nominal tributary frequency as discussed above. This difference between the nominal and actual frequencies is eliminated in the interface unit. Each incoming DTDM frame will be examined by the framer 154 in the associated interface 152 for its occupancy. The data packets contained in the occupied frames will be written into the FIFO 158 under the control of logic 159 and read out later by the framer 160 at the output end of the interface unit 152. Empty frames are discarded.

The reading of the data packets from the FIFOs 158 to the parallel data inputs (pdi's) of the framers 160 is synchronized. The serial data input (sdi) of each framer

160 is connected to the serial data output of a framer 162. The framer 162 serves to broadcast empty frames to the framers 160 so that each framer 160 receives a synchronous chain of empty frames at the nominal tributary frequency. The empty frames received by each framer unit 160 are filled with data packets from the associated FIFO 158 to produce synchronized tributary bit streams at the nominal tributary frequency.

If the actual frequency of a particular tributary is less than the nominal center frequency then on occasion, the associated FIFO 158 will not have a packet to insert into an empty DTDM FRAME. The net effect is that an empty DTDM frame is added so that the tributary acquires a frequency equal to the nominal frequency. However, if the actual frequency of the tributary is larger than the nominal center frequency the net effect is that empty DTDM frames are dropped so that the tributary acquires a frequency equal to the nominal frequency. Illustratively, the difference between the actual and nominal tributary frequencies is on the order of ten parts per million. In this case, a two frame capacity FIFO 158 is sufficient as long as each input tributary has one empty frame in 10^5.

All of the framers 160 send out frames at the same time, with frame alignment being automatically achieved. The aligned frames are then bit interleaved using bit interleaving circuit 164 to produce a single high bit rate bit stream at output 165 (see FIG. 3). Note that the clocks of the framers 160 are connected together so that data bits coming from the framers 160 are phase aligned and can be bit interleaved directly. The clock for the framers 160 is provided by the clock generator 166 and frequency divider 167. In an alternative embodiment of a time division multiplexer, instead of bit interleaving, frame or byte interleaving may also be used. If the frame interleaving is used then the multiplexed output bit stream has the same DTDM frame structure, thereby allowing the flexible single transport architecture to grow as the technology advances.

A time division demultiplexer 44 (see FIG. 3) for demultiplexing the high speed bit stream is illustrated in FIG. 11. The high speed data stream arrives on input line 170 (see FIG. 3) and is bit deinterleaved by means of bit deinterleave circuitry 172 into several lower speed tributary bit streams which are transmitted outward on lines 174. In order to dispatch the bits to correct tributaries, a predetermined span identification (SP ID) is inserted for each tributary before they are multiplexed at the transmit side. The tributary present on line 174*a* is connected to a framer unit 176, which will detect the frame boundary and determine by examining the span identification whether or not the bit deinterleave circuitry has correctly aligned the incoming bit stream so that appropriate data goes to appropriate output tributaries. If not, either a skip pulse is generated to rotate the bit sequence or a signal is generated by the framer 176 and sent to a cross point switch 178 to reassign the order of the bit stream. The bit streams with correct bit assignments appear at outputs 120, 179 (see FIG. 3). Alternatively, instead of the crosspoint switch 178, a barrel shifter may be used. It should be noted that the clock for the bit deinterleave circuit 171 and framer 176 is provided by clock recovery circuit 180 and frequency divider 182. Demultiplexers which operate according to similar principles are disclosed in R. J. Boehm et al. "Standardized Fiber Optic Transmission Systems - A Synchronous Optical Network View" IEEE Journal on Selected Areas in Communications VOL SAC-4 No. 9 pp 1424–1431 Dec. 1986 and L. R. Linnell "A Wide-band Local Access System using Emerging-technology Components" IEEE Journal on Selected Areas in Communications" VOL SAC-4 No. 4 pp 612–618 July 1986.

6. The Framer Circuit

The framer unit is an important component for the implementation of specific embodiments of the assemblers, disassemblers, multiplexers and demultiplexers which comprise the DTDM network discussed above.

The framer unit performs a number of functions in the DTDM network, including generating trains of empty DTDM frames, enabling the writing of data packets into specific DTDM frames, and the examination of header data in specific DTDM frames to generate signals for the control of peripheral circuits (e.g. in a DTDM demultiplexer to determine if data in a particular DTDM frame belongs to particular customer premises equipment or a particular less densely occupied DTDM bit stream). All of these functions may be carried out by the framer unit discussed below.

A framer unit 200 is schematically illustrated in FIG. 12. Illustratively, the framer unit 200 is formed as a single chip. The framer unit 200 has a serial data input 202, a parallel data input 204, a serial data output 206 and a parallel data output 208. Timing information for the framer unit 200 is provided by timing generator 209. The framer 200 operates under control of a control unit 210 which illustratively comprises one or more finite state machines.

As indicated above, a plurality of framer units may be connected in a daisy chain fashion and DTDM frames may be passed from one framer to the next (see e.g., framers 53 of FIG. 4). Data may be written into an empty DTDM frame as follows. A DTDM frame is received at the serial input 202. The DTDM frame is converted to parallel form by serial-to-parallel converter 212 and is detected by frame detector 214. The frame detector 214 is in communication with the control 210 and illustratively communicates to the control 210 information such as whether or not the frame is empty. Illustratively, the DTDM frame leaves the framer unit via the serial output 206 after conversion to serial form by way of parallel-to-serial converter 216. However the frame cannot reach the parallel-to-serial converter 216 unless the control 210 applies a signal to the tristate device 218.

The data to be written into the frame is received at the parallel data input 204 (illustratively from a FIFO 57 in the DTDM bit stream assembler 32 of FIG. 4). If the particular DTDM frame is empty and data is available at the parallel input 204, a signal is applied by the control 210 to the tristate device 220 to enable the data to be inserted into the particular DTDM frame via bus 219 before it leaves the framer unit. However, if the DTDM frame is already full the control does not provide such a signal to the tristate 220. In particular framer units additional information such as span identification may be inserted into specific DTDM frames by means of an additional tristate unit not shown.

The framer unit 200 may also be utilized to generate a chain of empty DTDM packets (see e.g., framer 52 in FIG. 4). In this case the serial input 202 and associated serial-to-parallel converter 212 are not utilized. Instead, the control 210 applies a periodic signal to tristate 222 so that a frame alignment word is periodically read from frame byte ROM 224 and transmitted via bus 219 to

parallel-to-serial converter 216 and serial output 206 so as to define a train of empty DTDM frames. Other information comprising the transmission overhead (T) field of the DTDM frame may also be stored in ROM 224 or provided by other sources connected to the bus 219 via a tristate device operative under the control of the control unit 219.

In particular situations (see e.g., framers 70 of FIG. 5 and 126 of FIG. 9), a framer unit receives occupied DTDM frames and the header (H) or transmission overhead (T) fields have to be examined to control peripheral circuit operations such as the reading of data into a FIFO. In this case, a multiple byte delay unit 230 may be included in the path between the serial input 202 and the parallel and serial outputs 208, 206. Typically a frame arrives at the serial input 202 and is converted to parallel form by the serial-to-parallel converter 212. The frame detector detects the frame and supplies necessary information from the header or transmission overhead fields to the control unit 210 which issues appropriate control signals via lines 232 such as user read/write strobes. Illustratively, the user read/write strobes control the writing of data from DTDM frames in the framer unit into associated FIFOs or other buffers. If the FIFO has byte wide format, the parallel output 206 may be used for this purpose. The delay unit 230 is used to insure that the necessary signal processing takes place before the DTDM frame leaves the framer unit.

7. Conclusion

A data transmission technique known as Dynamic Time Division Multiplexing (DTDM) has been disclosed along with an end-to-end network utilizing DTDM.

Finally, the above described embodiments of the invention are intended to be illustrative only. Numerous alternative embodiments may be devised by those skilled in the art without departing from the spirit and scope of the following claims.

What is claimed is:

1. A method for simultaneously transmitting data from sources having different bit rates in a telecommunication network comprising the steps of:

generating a bit stream comprising a sequence of frames, each of said frames including a transmission overhead field containing frame timing information and an empty payload field, and

filling the empty payload fields in said frames with data in packetized format from a plurality of sources which have access to the bit stream including circuit or packet sources, such that data in packetized format from any of said sources is written into any available empty payload field of any of said frames for transmitting data from each of said sources at its own desired bit rate via said bit stream and for transmitting data from said plurality of sources simultaneously via said bit stream.

2. The method of claim 1 wherein prior to filling said frames with slots from a circuit transmission stream, said slots are converted to said packetized format by placing a header in front of each of said slots.

3. A method for generating a bit stream capable of transporting data originating from both circuit transmission and packet sources comprising

generating a bit stream comprising a sequence of frames, each of said frames including a transmission overhead field containing frame timing information and an empty payload field,

packetizing data from a plurality of sources having different bit rates and which have access to said bit stream including circuit transmission sources or customer premises equipment to produce data packets, and

inserting said packets from said sources into the empty payload fields of said frames such that a packet from any of said sources is inserted into any available empty payload field of any of said frames for transmitting data from each of said sources at its own desired bit rate via said bit stream and for transmitting data from said plurality of sources simultaneously using said bit stream.

4. An apparatus for assembling a dynamic time division multiplexing bit stream comprising,

generating means for generating a train of frames wherein each frame includes a transmission overhead field containing timing information and an empty payload field,

processing means for processing data from a plurality of sources into packet format, and

inserting means for receiving said train of frames and for inserting each of said packets comprised of data from one of said plurality of sources into any empty payload field of any of said frames available to said inserting means to form said bit stream so that data from each of said sources can be transmitted at its own desired bit rate via said bit stream and so that data from said plurality of sources can be transmitted simultaneously via said bit stream.

5. The apparatus of claim 4 wherein said sources include circuit transmission bit streams or customer premises equipment.

6. An apparatus for assembling a bit stream for transmitting data from a plurality of sources comprising:

means for generating a train of frames, each of said frames including a transmission overhead field and an empty payload field, and

a plurality of interfaces, each of said interfaces serving to interface one of said sources with said train of frames, each of said interfaces comprising:

packetizing means for converting data into data packets,

memory means for storing at least one of said packets formed by said packetizing means, and

circuit means for inserting a packet stored in said memory means into any empty payload field of any available one of said frames so that data from each one of said sources can be transmitted at its own desired bit rate via said bit stream and so that data from said plurality of sources can be transmitted simultaneously via said bit stream.

7. The apparatus of claim 6 wherein said interface units are connected to one another serially and wherein said frames are passed sequentially to each of said interface units to receive said packets in said empty payload fields.

* * * * *

EXHIBIT 20

US00RE36633E

United States Patent [19]

Fleischer et al.

[11] E Patent Number: Re. 36,633

[45] Reissued Date of Patent: Mar. 28, 2000

[54] **SYNCHRONOUS RESIDUAL TIME STAMP FOR TIMING RECOVERY IN A BROADBAND NETWORK**

[75] Inventors: **Paul E. Fleischer**, Little Silver; **Chi-Leung Lau**, Marlboro, both of N.J.

[73] Assignee: **Telcordia Technologies, Inc.**, Morristown, N.J.

[21] Appl. No.: **08/555,196**

[22] Filed: **Nov. 8, 1995**

Related U.S. Patent Documents

Reissue of:

[64] Patent No.: **5,260,978**
Issued: **Nov. 9, 1993**
Appl. No.: **07/969,592**
Filed: **Oct. 30, 1992**

[51] Int. Cl.[7] **H04L 7/00**

[52] U.S. Cl. **375/354**; 375/362; 375/364; 370/509; 370/394; 370/516

[58] Field of Search 375/354, 355, 375/362, 365, 366, 371, 364; 370/503, 509, 510, 511, 512, 516, 394, 517, 519

[56] **References Cited**

U.S. PATENT DOCUMENTS

4,489,421	12/1984	Burger .
4,530,091	7/1985	Crockett .
4,759,014	7/1988	Decker et al. .
4,961,188	10/1990	Lau .
5,115,431	5/1992	Williams et al. .
5,255,291	10/1993	Holden et al. .

FOREIGN PATENT DOCUMENTS

3-114333	5/1991	Japan .

OTHER PUBLICATIONS

CCITT Study Group XVIII/8—Contribution D. 1123 entitled "Proposed method to provide the clock recovery function for circuit emulation", (Dec. 1990).

Letter from P. Adam of CNET dated Aug. 26, 1991 concerning discussion on TS/SFET.

Memorandum from R. Lau and B. Kittams of Bellcore dated Oct. 11, 1991 re A Compromise of SFET and TS.

Memorandum from T. Houdoin to B. Kittams & R. Lau dated Oct. 14, 1991 re Compromise SFET/TS.

CNET Memorandum from T. Houdoin to R. Lau and B. Kittams re Compromise SFET/TS (undated).

Y. Matsuura, S. Kozuka, K. Yuki, "Jitter Characteristics of Pulse Stuffing Synchronization," pp. 259–264, Jun. 1968.

D.L. Duttweiler, "Waiting Time Jitter," *Bell System Technical Journal*, vol. 51, No. 1, pp. 165–209, Jan. 1972.

CCITT Recommendations G. 707, G. 708, G. 709 (1988).

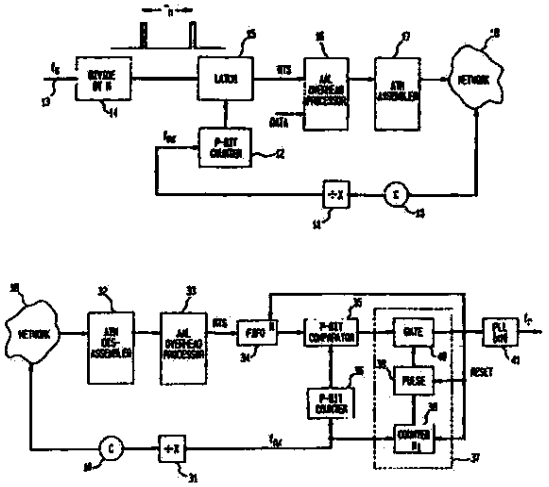
(List continued on next page.)

Primary Examiner—Don N. Vo
Attorney, Agent, or Firm—Joseph Giordano

[57] **ABSTRACT**

A Residual Time Stamp (RTS) technique provides a method and apparatus for recovering the timing signal of a constant bit rate input service signal at the destination node of a synchronous ATM telecommunication network. At the source node, a free-running P-bit counter counts cycles in a common network clock. At the end of every RTS period formed by N service clock cycles, the current count of the P-bit counter, defined as the RTS, is transmitted in the ATM adaptation layer. Since the absolute number of network clock cycles likely to fall within an RTS period will fall within a range determined by N, the frequencies of the network and service clocks, and the tolerance of the service clock, P is chosen so that the 2_P possible counts, rather than representing the absolute number of network clock cycles an RTS period, provide sufficient information for unambiguously representing the number of network clock cycles within that predetermined range. At the destination node, a pulse signal is derived in which the periods are determined by the number of network clock cycles represented by the received RTSs. This pulse signal is then multiplied in frequency by N to recover the source node service clock.

33 Claims, 3 Drawing Sheets



OTHER PUBLICATIONS

CCITT Recommendations G. 823, G. 824 (1988).
CCITT Recommendations D.294, "Timing for CBO Services," Jun. 1989.
R.C. Lau, "Synchronous Frequency Encoding Technique for B–ISDN Circuit Emulation," SPIE vol. 1179 Fiber Networking and Telecommunications, pp. 160–171, Sep. 1989.
H.M. Ahmed, "Adaptive Terminal Synchronization in Packet Data Networks," Globecom, pp. 20.6.1–20.6.5, 1989.
CCITT SG XVIII Contribution, D.1020, "Timing Recovery for CBR Circuit Emulation." Dec. 1990.
CCITT Study Group XVIII—Contribution D. 1451 entitled "Performance Comparison of Timing Recovery Methods for CBR Services," Jun. 1991.
Submission to T1S1 entitled "Broadband Aspects of ISDN," T1S1.5/91–382, Nov. 4, 1991.
Summary Minutes of T1 Services, Architecture, and Signaling Technical Subcommittee (T1S1) Session, Nov. 8, 1991.
CCITT Study Group XVIII entitled "Synchronous Residue–Time Stamp: A Combination of SFET/TS", Dec. 1991.
CCITT Study Group XVIII entitled "Report of the meeting of SWP XVIII/8–3 (Services, IVS and AAL types 1 and 2)", Dec. 1991.
CCITT Study Group XVIII entitled "Revised text of I.363 Section 2 (AAL type 1)", Dec. 1991.
CCITT Recommendation I.363 entitled "B–ISDN ATM Adaptation Layer (AAL) Specification," (1991).
Appendix 5 (to Annex 7) for COM XVIII–R 70–E entitled "Descriptions of the SFET Method" (Jul. 1991).
Appendix 6 (to Annex 7) for COM XVIII–R 70–E entitled "Descriptions of the TS Method" (Jul. 1991).
R.C. Lau and P.E. Fleischer, "Synchronous Techniques for Timing Recovery in BISDN," *IEEE Globecom*, pp. 814–820, Dec. 1992.
R.C. Lau and P.E. Fleischer, "Synchronous Techniques for Timing Recovery in BISDN," Communications Switching of the IEEE Communications Society, IEEE Transactions on Communications, Apr. 1995.
Kozuka, "Phase Controlled Oscillator for Pulse Stuffing Synchronization System," Review of the Electrical Communications Laboratory, vol. 17, Nos. 5–6, May–Jun. 1969.
R.P. Singh et al., "Jitter and Clock Recovery for Periodic Traffic in Broadband Packet Networks," Bell Communications Research, Inc., presented at IEEE Globecom, Florida, Dec. 1988.
J. Gonzales et al., Proceedings ICC '91, "Jitter Reduction in ATM Networks," 1991, pp. 274–279.
R.P. Singh et al., "Adaptive Clock Synchronization Schemes for Real–Time Traffic in Broadband Packet Networks," Presented–8th European Conference on Electrotechnics, Stockholm, Sweden, Jun. 1988.
American National Standard, ANSI T1.105–1988, "Digital Hierarchy Optical Interface Rates and Formats Specification," Sep. 1988.
"Jitter Reduction in ATM Networks," Julio Gonzales and Jean–Paul Le Meur Proceedings ICC '91, 9.4.1–9.4.6.

FIG. 1

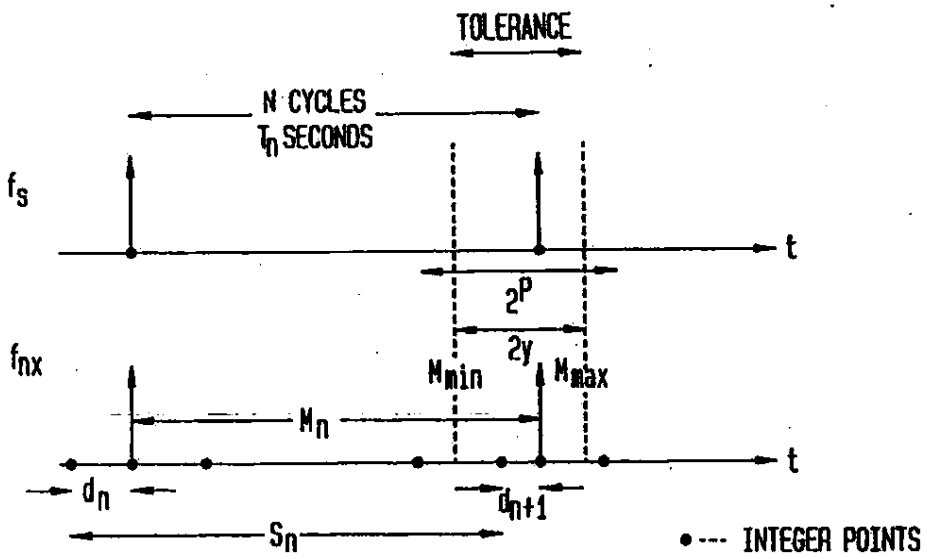
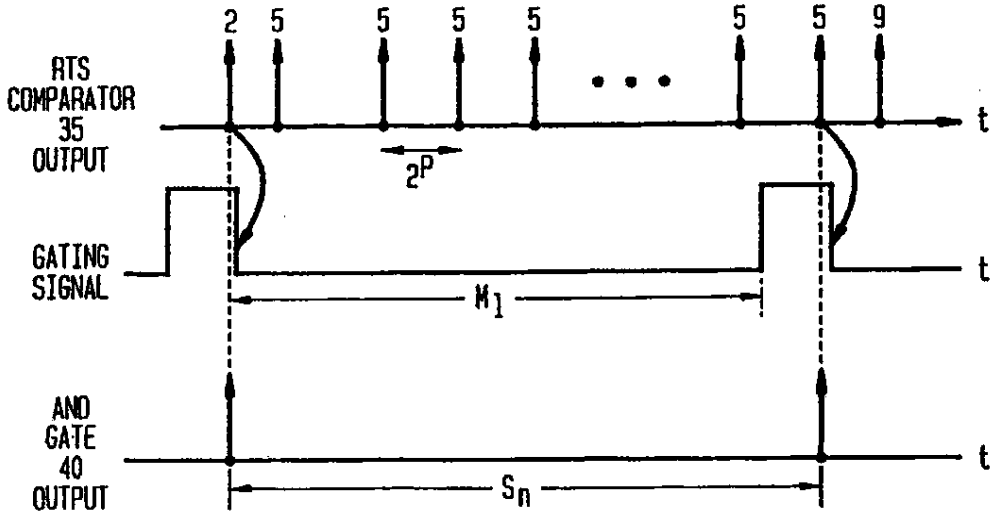

TOLERANCE
N CYCLES
T_n SECONDS
f_s
t
2^P
2y
f_{nx}
M_{min}
M_{max}
M_n
t
d_n
d_{n+1}
S_n
INTEGER POINTS

FIG. 4


2 5 5 5 5 5 5 9
RTS COMPARATOR 35 OUTPUT
t
2^P
GATING SIGNAL
t
M_1
AND GATE 40 OUTPUT
t
S_n

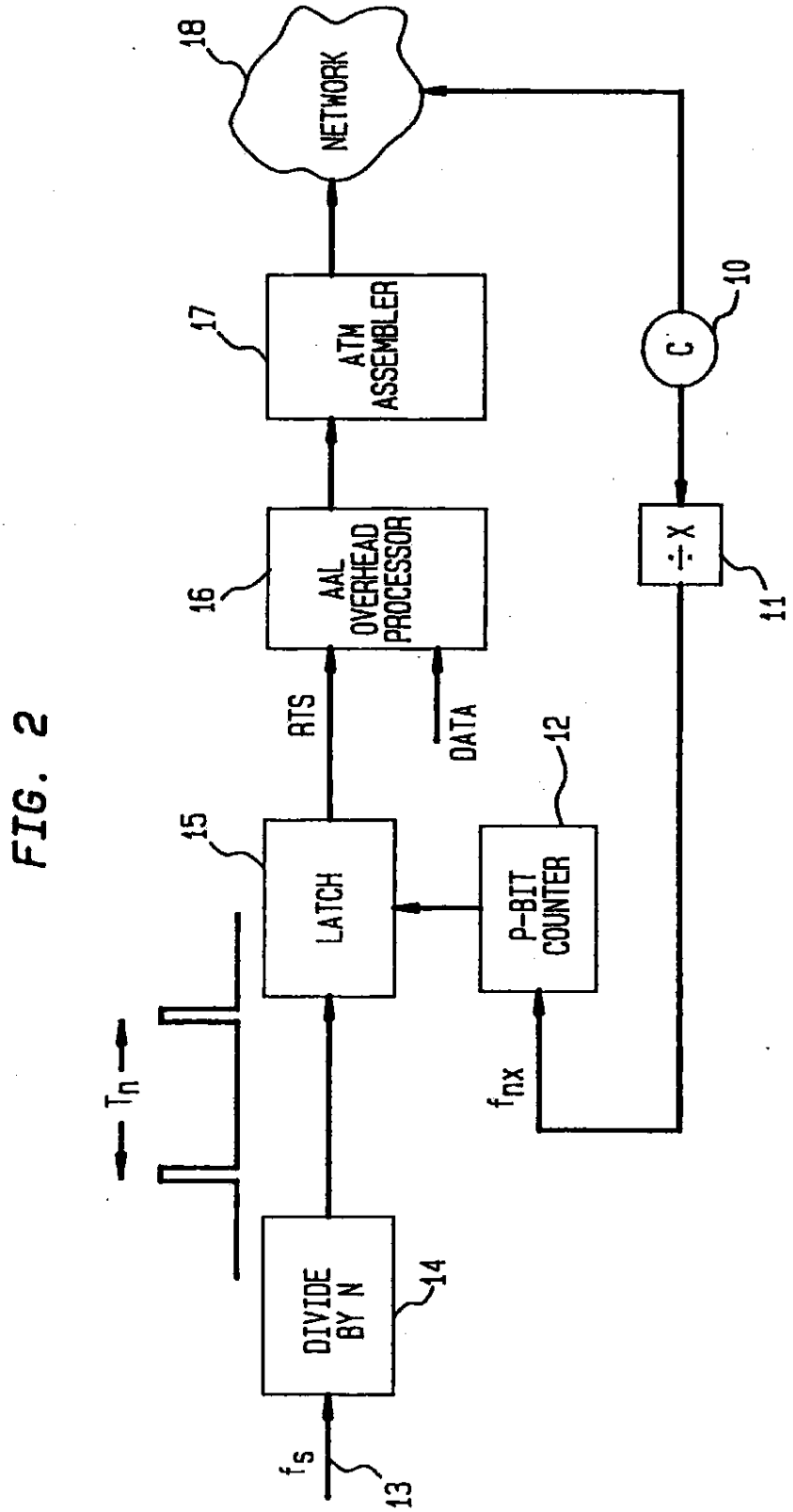

NETWORK
18
ATM ASSEMBLER
17
AAL OVERHEAD PROCESSOR
16
RTS
DATA
LATCH
15
P-BIT COUNTER
12
DIVIDE BY N
14
f_s
13
T_n
f_{nx}
C
10
÷ X
11

FIG. 2

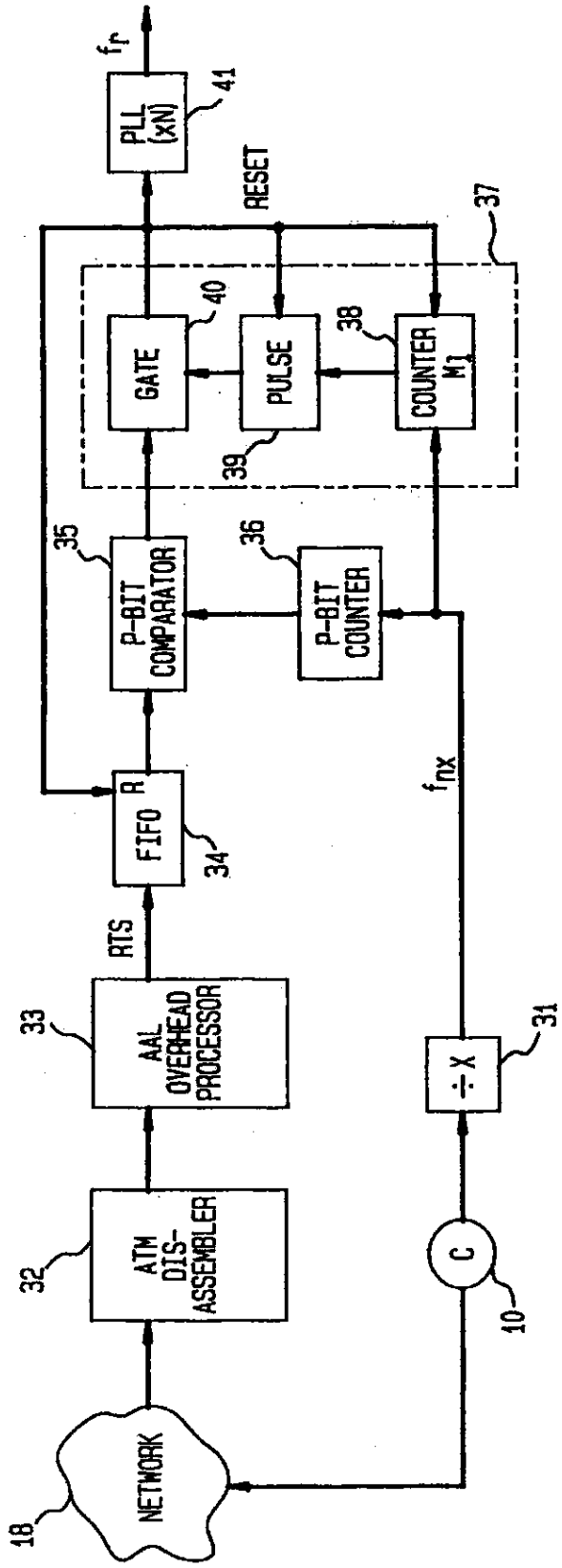

f_r
PLL (xN)
41
RESET
37
GATE
40
PULSE
38
COUNTER M1
39
35
P-BIT COMPARATOR
36
P-BIT COUNTER
R
FIFO
34
f_{nx}
RTS
33
AAL OVERHEAD PROCESSOR
÷X
31
32
ATM DIS-ASSEMBLER
C
10
NETWORK
18

FIG. 3

SYNCHRONOUS RESIDUAL TIME STAMP FOR TIMING RECOVERY IN A BROADBAND NETWORK

Matter enclosed in heavy brackets [ ] appears in the original patent but forms no part of this reissue specification; matter printed in italics indicates the additions made by reissue.

This application is the parent application of reissue application 09/292,668 filed Apr. 16, 1999.

BACKGROUND OF THE INVENTION

This invention relates to timing recovery of a source node service clock frequency at a destination node in a broadband asynchronous transfer mode (ATM) network where the source and destination nodes receive reference timing signals derived from a single master clock.

Asynchronous Transfer Mode (ATM) is a packet oriented technology for the realization of a Broadband Integrated Services Network (BISDN). By using ATM, network resources can be shared among multiple users. Moreover, various services including voice, video and data can be multiplexed, switched, and transported together under a universal format. Full integration will likely result in simpler and more efficient network and service administration and management. However, while conventional circuit-switching is optimized for real-time, continuous traffic, ATM is more suitable for the transport of bursty traffic such as data. Accommodation of constant bit rate (CBR) services is, however, an important feature of ATM, both for universal integration and for compatibility between existing and future networks. In the transport of a CBR signal through a broadband ATM network, the CBR signal is first segmented into 47-octet units and then mapped, along with an octet of ATM Type I Adaptation Layer (AAL) overhead, into the 48-octet payload of the cell. The cells are then statistically multiplexed into the network and routed through the network via ATM switches.

It is essential to the proper delivery of such CBR service traffic in a broadband network that the clock controlling the destination node buffer be operating at a frequency precisely matched to that of the service signal input at the source node in order to avoid loss of information due to buffer over- or under-flow. However, unlike the circuit-switched transport of service data wherein the clock frequency at the destination node may be traced directly back to that of the source node by the regular, periodic arrival of the CBR traffic, transport in an ATM network inherently results in cell jitter, i.e. the random delay and aperiodic arrival of cells at a destination node, which essentially destroys the value of cell arrival instances as a means for directly recovering the original service signal input frequency.

Such cell jitter, generally the result of the multiplexing of transport cells in the broadband network and the cell queuing delays incurred at the ATM switches in the network, is substantially unpredictable. Thus, little is known about the cell arrival time beyond the fact that the average cell delay is a constant, assuming that the ATM network provides sufficient bandwidth to ensure against loss of cells within the network. As a means for closely approximating the service signal frequency at the destination node, some consideration had previously been given to utilizing a direct extension of circuit-switched timing recovery practices which rely entirely upon a buffer fill signal as the basis for recovery of the source timing. However, due to the lack of knowledge of statistics of the cell jitter, this approach would have required

a phase-locked loop with very low cut-off frequency (in the order of a few Hz) and would thus have resulted in excessive converging time and degradation of jitter and wander performance.

A number of schemes have been proposed to improve upon such a conventional manner of recovering service timing in the presence of cell jitter, yet none has achieved this end economically and without extensive control systems of notable complexity. Singh et al., for example, in "Adaptive Clock Synchronization Schemes For Real-Time Traffic In Broadband Packet Networks," 8th European Conference on Electrotechnics, Stockholm, Sweden, June 1988, and "Jitter And Clock Recovery For Periodic Traffic In Broadband Packet Networks," IEEE Globecom '88, Florida, December 1988, have proposed algorithms which attempt to more closely estimate cell jitter statistics and derive timing recovery from those indications. These adaptive approaches, suggested to be applicable to both synchronous and non-synchronous networks, rely upon the interaction of increasingly complex algorithms which would require the noted extensive controls for implementation.

These prior art schemes described above can be classified as non-synchronous techniques, which are based on the simple fact that the expected value of the network cell jitter is zero and thus rely on phase filtering. Synchronous techniques, on the other hand, utilize the fact that common timing is available at both the transmitter and the receiver. In a synchronous broadband ATM network, such as the Synchronous Optical Network (SONET) prescribed by American National Standard, ANSI T1.105-1988, "Digital Hierarchy Optical Interface Rates and Formats Specification," Mar. 10, 1988, the network source and destination node control clocks are synchronized to the same timing reference. As a result, there is no necessity for relying upon any extraneous phenomenon such as instants of cell arrival to provide a datum base for determining the relative frequencies of those control clocks. The effect of cell jitter caused by multiplexing and switching delays in the network is therefore of little consequence in any procedure for circuit transporting CBR service, which is based, as is the present invention, on an actual synchrony of node timing. Thus being devoid of concern for cell jitter, this process is free to simply determine the difference in frequency between the CBR service signal input at the source node and the source/destination node timing clock(s).

U.S. Pat. No. 4,961,188 issued on Oct. 2, 1990 to Chi-Leung Lau, co-inventor herein, discloses a synchronous frequency encoding technique (SFET) for clock timing in a broadband network. The SFET takes advantage of the common timing reference at both the source and the receiver. At the source, the asynchronous service clock is compared to the network reference clock. The discrepancy between properly chosen submultiples of the two clocks is measured in units of a preassigned number of slip cycles of network clock. This clock slip information is conveyed via a Frequency Encoded Number (FEN) which is carried in the ATM Adaptation Layer (AAL) overhead. At the receiver, the common network clock and the FEN are used to reconstruct the service clock. This timing recovery process does not rely on any statistics of the cell jitter except that it has a known, bounded amplitude. Therefore, the recovered clock has jitter performance comparable to that of the circuit-switched network.

An alternative proposed approach is known as Time Stamp (TS). In the Time Stamp approach (see, for example, Gonzales et al, "Jitter Reduction in ATM Networks", Proceedings ICC'91, 9.4.1–9.4.6), the network clock is used to

drive a multi-bit counter (16-bits in the proposal), which is sampled every fixed number of generated cells (e.g., 16). Thus, a fixed number, N, of service clocks cycles is used as the measuring yardstick. The sampled value of the 16-bit counter is the TS that inherently conveys the frequency difference information. Because of the size of the TS (2 octets), it has been proposed that the TS be transmitted via the Convergence Sublayer (CS) overhead. Thus the TS is a 16-bit binary number occurring once every N service clock cycles. Differences in successive TSs represent the quantized values of M, where M is the number of network clock cycles during the fixed TS period. At the receiver, the TS period is reconstructed from the received TSs and the network clock. A free-running 16-bit counter is clocked by the network clock and the output of the counter is compared to the received TSs which are stored in a TS FIFO. A pulse is generated whenever there is a match between the TS and the 16-bit counter. The service clock is recovered by supplying the resultant pulse stream as the reference signal to a multiply-by-N phase locked loop (PLL).

A comparison of the SFET approach and the TS approach reveals advantages and disadvantages for each. In the SFET approach there is a relatively stringent requirement on the derived network clock since it must be slightly larger than the service clock. Advantageously, however, a convergence sublayer is not required to transmit the FEN and only small overhead bandwidth is required to transmit the necessary information. On the other hand, the TS approach is more flexible in that it does not require stringent relationships between the service clock and the network derived clock and can therefore support a range of service bit rates. Disadvantageously, however, a rigid convergence sublayer structure is required to transmit the TS, which adds complexity and makes inefficient use of the overhead bandwidth.

An object of the present invention is to achieve synchronous timing recovery with an approach that has the advantages of both the SFET and TS approaches, specifically, the efficiency of SFET and the flexibility of TS.

SUMMARY OF THE INVENTION

As described hereinabove, the TS approach requires a large number of bits (16-bits in the example), to represent the number of network clock cycles within a time interval defined by a fixed number (N) of service clock cycles. In accordance with the present invention, the number of bits required to represent the number of network clock cycles within that time interval is substantially reduced. This is possible through the realization that the actual number of network clock cycles, M (where M is not necessarily an integer), deviates from a nominal known number of cycles by a calculable deviation that is a function of N, the frequencies of the network and service clocks, and the tolerance of the service clock. Specifically, therefore, rather than transmitting a digital representation of the quantized actual number of network clock cycles within the interval, only a representation of that number as it exists within a defined window surrounding an expected, or nominal, number of network clock pulses is transmitted from a source node to a destination node in an ATM network. This representation will be referred to hereinafter as the Residual Time Stamp (RTS). By selecting the number of bits, P, so that all 2_P possible different bit patterns uniquely and unambiguously represent the range of possible numbers of network clock cycles within the fixed interval that is defined by N service clock cycles, the destination node can recover the service clock from the common network clock and the received RTS.

At the source node, a free-running P-bit counter counts clock cycles in a clock signal derived from the network clock. The service clock, which is derived from the incoming data signal to be transmitted over the ATM network, is divided by the factor of N to produce a pulse signal having a period (the RTS period) which defines the time interval for measuring the number (modulo 2^P) of derived network clock pulses. At the end of each RTS period, the current count of the free-running P-bit counter is sampled. That sampled value is the RTS, which is transmitted via the adaptation layer. Since the service clock from which the RTS period is defined and the derived network clock are neither synchronized nor integrally related in frequency, the actual number of derived network clock cycles in a RTS period is unlikely to be an integer. Thus, when sampled at the end of each RTS period, the increment in the count of the P-bit counter is a quantized version of the count (modulo 2^P) of pulses in the RTS interval as modified by any accumulated fractional counts from a previous interval.

At the destination node, after the AAL is processed, the successive RTSs are converted into a pulse signal which has periods between pulses defined by the fixed integral numbers of derived network clock pulses that correspond to the conveyed RTS periods. Specifically, a free-running P-bit counter is driven by the derived network clock. A comparator compares this count with a stored received RTS and produces a pulse output upon a match. Since the count of the P-bit counter matches the stored RTS every 2_P derived network clock cycles, comparator output pulses that do not actually represent the end of the RTS period are inhibited by gating circuitry. This gating circuitry includes a second counter that counts the derived network clock cycles occurring since the end of the previous RTS period. When this second counter reaches a count equal to the minimum possible number of derived network clock pulses within an RTS period, the next comparator pulse output produced upon a match between the RTS and the count of the P-bit counter, is gated-through to the output and resets the second counter. The resultant gated through output pulse stream drives a multiply-by-N phase locked loop to recover the service clock.

BRIEF DESCRIPTION OF THE DRAWING

FIG. 1 are timing diagrams showing the RTS concept of the present invention;

FIG. 2 is a block diagram showing apparatus, in accordance with the present invention, for generating the RTS at the source node of an ATM network;

FIG. 3 is a block diagram showing apparatus, in accordance with the present invention, for reconstructing the service clock at the destination node of an ATM network; and

FIG. 4 are timing diagrams showing the gating function at the apparatus of FIG. 3.

DETAILED DESCRIPTION

The concept of the Residual Time Stamp is described with reference to FIG. 1. In FIG. 1, and in the description hereinafter, the following terminology is used:

f_n—network clock frequency, e.g. 155.52 MHz;

f_{nc}—derived network clock frequency,

$$f_{nc} = \frac{f_n}{x},$$

where x is a rational number;

f_s—service clock frequency;

N—period of RTS in units of the service clock (f_s) cycles;

T_n—the n-th period of the RTS in seconds;

$\pm\varepsilon$—tolerance of the source clock frequency in parts per million;

M_n (M_{nom}, M_{max}, M_{min})—number of f_{nx} cycles within the n-th (nominal, maximum, minimum) RTS period, which are, in general, non-integers.

As can be noted in FIG. 1, during the n-th period, T_n, corresponding to N service clock cycles, there are M_n network derived clock cycles. As aforenoted, since the service clock and the network clock are neither synchronized nor integrally related in frequency, this number of derived network clock cycles is not an integer. Since all practical timing recovery techniques transmit only integer values, the fractional part of M_n must be dealt with. Simple truncation or rounding of the fractional part in each RTS time slot is not permissible, as this would lead to a "random walk" type error accumulation. Rather, it is necessary to accumulate the fractional parts at the transmitter and use the accumulated value to modify the transmitted integer quantity. Since it is most convenient to generate RTS by an asynchronous counter, as will be described hereinafter in conjunction with the description of FIG. 2, a "truncation" operation is natural, reflecting the fact that an asynchronous counter's output does not change until the subsequent input pulse arrives. To formalize these notions, S_n is defined as the truncated value of M_n after accounting for the left over fractional part, d_n, from the (n−1)-th interval, viz.,

$$S_n=[M_n+d_n] \tag{1}$$

and

$$d_{n+1}=d_n+M_n-S_n \tag{2}$$

where [a] denotes the largest integer less than or equal to a. Since for accurate clocks, the range of M_n is very tightly constrained, i.e., $M_{max}-M_{min}=2y<M_n$, the variation in S_n is also smaller than its magnitude. It follows from Equation (1) that

$$[M_{min}+d_n]\geq S_n\geq[M_{max}+d_n] \tag{3}$$

Since the maximum and minimum of d_n are 1 and 0 respectively, S_n is bounded by,

$$[M_{min}]\geq S_n\geq[M_{max}]+1 \tag{4}$$

This implies, that the most significant portion of S_n carries no information and it is necessary to transmit only its least significant portion. This, therefore, is the essential concept of the RTS. The minimum resolution required to represent the residual part of S_n unambiguously is a function of N, the ratio of the network derived frequency to the service frequency, and the service clock tolerance, $\pm\varepsilon$. The maximum deviation, y, between the nominal number of derived network clock pulses in an RTS period, M_{nom}, and the maximum or minimum values of M (M_{max} or M_{min}) is given by,

$$y = N \times \frac{f_{nx}}{f_s} \times \varepsilon \tag{5}$$

where M_{nom} equals

$$N \times \frac{f_{nx}}{f_s}.$$

A specific numerical example can be considered for clarity of understanding. As illustrative derived network clock frequency and service clock frequencies could be given by f_{nx}=155.52 MHZ (for x=1), and f_s=78.16 MHz (nominal), respectively. A typical RTS sampling period (N) is 3008, which corresponds to a period of 8 cells and a 47-octet payload per cell (47 bytes/cell×8 bits/byte×8 cells per RTS period). Using these numbers, M_{nom}=5985.2119. If it is further reasonable to assume that the service clock tolerance is 200 parts per million, i.e., $\pm 200\times10^{-6}$. From equation (5), therefore, y=1.197, which demonstrates that it is superfluous to transmit the full S_n in each RTS sampling period and transmission of the last few (P) bits of S_n is sufficient. This P-bit sample is the Residual-TS (RTS).

FIG. 2 is a block diagram of the source node of an ATM network showing apparatus for generating and transmitting the RTS. The basic network clock, C, shown at 10, serves as the reference for timing of all nodes of the synchronous network being here considered. This clock, having a frequency f_{nx}, is divided in frequency by a rational factor x by a divider 11 to produce a derived network clock having a frequency f_{nx}. Preferably, x would be an integer value. The dividing factor is chosen so that the P bits available can unambiguously represent the number of derived network clock cycles within an RTS period. In the case where

$$\frac{f_{nx}}{f_s}$$

is less than or equal to two, as in the example above, it can be shown that a 3-bit RTS is sufficient.

The derived network clock, f_{nx}, drives a P-bit counter, which is continuously counting these derived network clock pulses, modulo 2^P. The service clock, f_s, on lead 13, which is derived from the service data signal (not shown) to be transmitted over the ATM network, is divided in frequency by N, the desired RTS period in units of f_s cycles, by divide-by N circuit 14. As shown in FIG. 2, the output of divider 14 is a pulse signal in which T_n is its n-th period. At every T seconds (N source clock cycles) latch 15 samples the current count of counter 12, which is then the P-bit RTS to be transmitted. As aforedescribed, this number represents the residual part of S_n and is all that is necessary to be transmitted to recover the source clock at the destination node of the network.

Each successive RTS is incorporated within the ATM adaptation layer overhead by AAL processor 16. The associated data to be transmitted (not shown) is also processed by processor 16 to form the payload of the cells, which are then assembled by an ATM assembler 17, which adds an ATM header for transmission over the network 18.

With reference again to the previous example, a four-bit counter (P=4) can be assumed to be used. Since M_{nom}=5985.2119 and 5985.2119 (modulo 16)=1.2119, a typical RTS output sequence when the source is at nominal frequency will be as follows;

. . . 5,6,7,9,10,11,12,13,15,1,2, . . .

Since the counter 16, in effect, quantizes by truncation, the RTS changes only by integer values. The changes in RTS are

such that their average is exactly equal to M_{nom} (modulo 2^P). In this example, the changes are either 1 or 2 with the change of 2 occurring either every 4 or 5 RTSs in such a way that the average interval is 1/0.2119=4.7198. In general, successive RTSs are related by

$$RTS_{n+1}=RTS_n+S_n=RTS_n+[d_n+M_n](\text{modulo } 2^P) \quad (6)$$

In order to guarantee that no information is lost due to the modulo arithmetic, i.e., that the transmitted RTS represents S_n unambiguously, it can be seen from equation (4) that the number of bits used for transmission must satisfy:

$$2_P \geq [M_{max}]-[M_{min}]+2 \quad (7)$$

Thus, in the example above, the number of bits allocated to the RTS must be 3 or greater. It can be noted that the number of bits necessary to unambiguously represent the number of derived network clock cycles within the RTS period is substantially less than the number of bits that would be required to represent the absolute number of clock cycles within the same interval. In the example above, for example, a 13-bit number would be required to represent M_{nom}.

If equation (7) is satisfied, knowledge of M_{nom} in the receiver at the destination node along with the received RTSs can be used to reproduce the service clock from the synchronous network clock. FIG. 3 shows one receiver implementation for reproducing the service clock from the received RTSs. At the receiver the common network clock 10 is available as it was at the transmitter. As in the transmitter, a divider 31 divides the network clock frequency, f_n, by the same factor of x as divider 11 in the source node, to produce the same derived network clock signal having a frequency f_{nr}, as was used by the transmitter at the source node of FIG. 2.

In a structure paralleling the transmitter in FIG. 2, a disassembler 32 processes the ATM headers received from the network 18 and passes the payload to an AAL processor 33. In addition to extracting the transmitted data (not shown), processor 33 extracts the periodic transmitted RTSs, which are sequentially stored in a FIFO 34, which is used to absorb the network cell jitter. The earliest received RTS in FIFO 34 is compared by P-bit comparator 35 with the count of a free running P-bit counter 36, driven by the derived network clock, f_{nr}. Whenever the output of counter 36 matches the current RTS, comparator 35 generates a pulse. Since counter 36 is a modulo 2_P counter, the RTS in FIFO 34 matches the count of counter 36 every 2_P derived network clock pulses, f_{nr}. The output of comparator 35 thus consists of a train of pulses that are separated, except for the first pulse, by 2_P cycles of the derived network clock. In order to select the output pulse of comparator 35 that corresponds to the end of the fixed period of the transmitted service clocks, which is the period per RTS to be recovered, gating circuitry 37 is employed. Gating circuitry 37, which includes a counter 38, a gating signal generator 39, and an AND gate 40, gates only that pulse output of comparator 35 produced after counting, from the last gated output pulse, a minimum number, M_f, of derived network clock cycles. This minimum number, M_f, is given by:

$$M_f=[M_{min}]-(^{P-2}) \quad (8)$$

This ensures that $[M_{max}]-2_P<M_f<[M_{min}]$, and thus the gating pulse is guaranteed to select the correct RTS.

The gating function is best explained in conjunction with the timing diagrams of FIG. 4. Initially, it can be assumed

that gating signal generator 39 is set to keep AND gate 40 open. Comparator 35 compares the first RTS in FIFO 34 with the free-running count of counter 36. When the count of counter 36 matches this first RTS, shown in FIG. 4 as "2", comparator 35 produces a pulse which is gated through AND gate 40. This gated output pulse resets gating signal generator 39 thereupon turning off AND gate 40, resets the counter of counter 38 to zero, and reads the next stored RTS, "5", in FIFO 34. When counter 36 reaches the count of "5", comparator 35 produces another output pulse. AND gate 40, however, is OFF and remains off until counter 38 counts M_f derived network clock cycles. Therefore, as noted in FIG. 4, all the subsequent matches of the RTS, "5" and the count of counter 36, which occur every 2_P derived network clock cycles, are blocked by AND gate 40. These subsequent pulses are blocked until counter 38 reaches a count of that minimum number of clock cycles that can comprise the fixed interval to be recovered from the RTS. After counting M_f derived network clock cycles, counter 38 generates a pulse which signals gating signal generator 39 to open AND gate 40. The next pulse produced by comparator 35 upon the match between the RTS in FIFO 34 and the count of counter 36 is gated through AND gate 40. This pulse, as before, resets counter 38, resets gating signal generator 39, and reads-in the next stored RTS to the output of FIFO 34. The resultant time difference between output pulses of AND gate 40 is the desired fixed time interval, S_n, to be recovered from the transmitted RTSs. As previously defined in equation (1), S_n is the truncated value in the nth interval, after accounting for a left over portion from the (n−1)-th interval, of the actual number of derived network clock cycles within the fixed interval defined by N source clock cycles. As can be noted, S_n modulo (2^P) is equal to the difference of the RTSs associated with the pulses matched by comparator 35 right before and right after the reset. Thus in FIG. 4, for the n-th period, this is the difference between "5" and "2", or "3", and for the (n+1)-st period, this is the difference between "9" and "5" or "4". The resultant pulse train at the output of gating circuitry 37 can be seen to duplicate the signal at the source node of the network, which is defined by N service clock cycles, as modified by the quantization effect of the RTSs. This pulse stream is input to a multiply-by N phase-locked loop 41 which multiplies the frequency by the factor of N and smooths out the variation of the reproduced periods. The resultant output clock signal, f_r, is the reproduced service timing signal, which can be employed by the circuitry at the destination node.

The above-described embodiment is illustrative of the principles of the present invention. Other embodiments could be devised by those skilled in the art without departing from the spirit and scope of the present invention.

What is claimed is:

1. A method of recovering, at a destination node of a packet-based telecommunications network, the timing clock of a service input at a source node of said packet-based telecommunications network, the destination node and the source node having a common network clock, comprising the steps of:

(a) at the source node, dividing the timing clock of the service input by a factor of an integer N to form residual time stamp (RTS) periods;

(b) at the source node, counting the network clock cycles modulo 2^P, where 2_P is less than the number of network clock cycles within an RTS period and P is chosen so that the 2_P counts uniquely and unambiguously represent the range of possible network clock cycles within an RTS period;

(c) transmitting from the source node to the destination node an RTS at the end of each RTS period that is equal to the modulo 2_P count of network clock cycles at that time;

(d) determining from the RTSs received at the destination node, the number of network clock cycles in each RTS period;

(e) generating a pulse signal from the network clock at the destination node in which the period between each pulse in the pulse signal equals the determined number of network clock cycles in the corresponding RTS period; and

(f) multiplying the frequency of the pulse signal generated in step (e) by the same factor of an integer N used in step (a) to recover the timing clock of the service input.

2. The method of claim 1 wherein the network clock frequency is less than or equal to twice the service clock frequency.

3. A method of recovering, at a destination node of a packet-based telecommunications network, the timing clock of a service input at a source node of said packet-based telecommunications network, the destination node and the source node having a common network clock, comprising the steps of:

(a) at the source node, dividing the timing clock of the service input by a factor of an integer N to form residual time stamp (RTS) periods;

(b) at the source node, dividing the network clock by a rational factor to form a derived network clock;

(c) at the source node, counting the derived network clock cycles modulo 2^P, where 2_P is less than the number of derived network clock cycles within an RTS period and P is chosen so that the 2_P counts uniquely and unambiguously represent the range of possible derived network clock cycles within an RTS period;

(d) transmitting from the source node to the destination node an RTS at the end of each RTS period that is equal to the modulo 2_P count of derived network clock cycles at that time;

(e) at the destination node, dividing the network clock by the same rational factor used at the source node to form a derived network clock equal to the derived network clock at the source node;

(f) determining from the RTSs received at the destination node, the number of derived network clock cycles in each RTS period;

(g) generating a pulse signal from the derived network clock at the destination node in which the period between each pulse in the pulse signal equals the determined number of derived network clock cycles in the corresponding RTS period; and

(h) multiplying the frequency of the pulse signal generated in step (g) by the same factor of an integer N used in step (a) to recover the timing clock of the service input.

4. The method of claim 3 wherein the derived network clock frequency is less than or equal to twice the service clock frequency.

5. Apparatus for recovering, at a destination node of a packet-based telecommunications network, the timing clock of a service input at a source node of said packet-based telecommunications network, the destination node and the source node having a common network clock, comprising at the source node:

dividing means for dividing the timing clock of the service input by a factor of an integer N to form residual time stamp (RTS) periods;

counting means connected to the network clock for counting network clock cycles modulo 2^P, where 2_P is less than the number of network clock cycles within an RTS period and P is chosen so that the 2_P counts uniquely and unambiguously represent the range of possible network clock cycles within an RTS period; and

transmitting means, responsive to the RTS periods formed by said dividing means and the count of said counting means, for transmitting over the telecommunications network an RTS at the end of each RTS period that is equal to the modulo 2^P count of network clock cycles at that time;

and comprising at the destination node:

receiving means for receiving the RTSs transmitted over the telecommunications network by said transmitting means;

converting means responsive to the received RTSs and the network clock for converting the received RTSs into a pulse signal in which the periods between pulses are determined from the numbers of network clock cycles associated with the counts of network clock cycles within said RTS periods; and

means for multiplying the frequency of the pulse signal generated by said converting means by the same factor of an integer N used in said dividing means for recovering the timing clock of the service input.

6. Apparatus in accordance with claim 5 wherein the network clock frequency is less than or equal to twice the service clock frequency.

7. Apparatus in accordance with claim 5 wherein said converting means comprises:

means for sequentially storing the received RTSs;

means for counting network clock cycles modulo 2_P;

comparing means for comparing the modulo 2_P count of network clock cycles with a stored RTS and for generating a pulse each time the count of network clock cycles matches the RTS; and gating means for gating to said multiplying means, for each sequentially received and stored RTS, the pulse produced by said comparing means that occurs after the counting means counts, starting-in-time from the previous gated pulse, a number of network clock cycles that is greater than a predetermined minimum absolute number of network clock cycles that can occur within any RTS period.

8. Apparatus for recovering, at a destination node of a packet-based telecommunications network, the timing clock of a service input at a source node of said packet-based telecommunications network, the destination node and the source node having a common network clock, comprising at the source node:

first dividing means for dividing the timing clock of the service input by a factor of an integer N to form residual time stamp (RTS) periods;

second dividing means for dividing the network clock by a rational factor to form a derived network clock;

counting means connected to the network clock for counting derived network clock cycles modulo 2^P, where 2_P is less than the number of derived network clock cycles within an RTS period and P is chosen so that the 2_P counts uniquely and unambiguously represent the range of possible derived network clock cycles within an RTS period; and

transmitting means, responsive to the RTS periods formed by said first dividing means and the count of said counting means, for transmitting over the telecommu-

nications network an RTS at the end of each RTS period that is equal to the modulo 2^P count of derived network clock cycles at that time;

and comprising at the destination node:

receiving means for receiving the RTSs transmitted over the telecommunications network by said transmitting means;

means for dividing the network clock by the same rational factor used at the source node to form a derived network clock;

converting means responsive to the received RTSs and the derived network clock for converting the received RTSs into a pulse signal in which the periods between pulses are determined from the numbers of derived network clock cycles associated with the counts of derived network clock cycles within said RTS periods; and

means for multiplying the frequency of the pulse signal generated by said converting means by the same factor of an integer N used in said first dividing means for recovering the timing clock of the service input.

9. Apparatus in accordance with claim 8 wherein the derived network clock frequency is less than or equal to twice service clock frequency.

10. Apparatus in accordance with claim 8 wherein said converting means comprises:

means for sequentially storing the received RTSs;

means for counting derived network clock cycles modulo 2_P;

comparing means for comparing the modulo 2_P count of derived network clock cycles with a stored RTS and for generating a pulse each time the count of derived network clock cycles matches the RTS; and

gating means for gating to said multiplying means, for each sequentially received and stored RTS, the pulse produced by said comparing means that occurs after the counting means counts, starting-in-time from the previous gated pulse, a number of derived network clock cycles that is greater than a predetermined minimum absolute number of derived network clock cycles that can occur within any RTS period.

11. Apparatus for generating a representation of the relationship between the timing clock of a service input, at a source node of a packet-based telecommunications network, and a network clock, the apparatus comprising:

(a) means, at the source node, for defining a residual time stamp (RTS) period as an integral number N of source-node service clock cycles;

(b) means, at the source node, for defining a derived network clock frequency f_{nx} from a network frequency f_n where $f_{nx}=f_n/x$, x is a rational number, and f_{nx} is less than or equal to twice the service clock frequency;

(c) means, at the source node, for counting the derived network clock cycles modulo 16 in an RTS period and;

(d) means for transmitting from the source node an RTS that is equal to the modulo 16 count of derived network clock cycles in the RTS period.

12. Apparatus for recovering, at a destination node of a packet-based telecommunications network, the timing clock of a service input at a source node of the packet-based telecommunications network, wherein the destination and source nodes have a common network clock divided network clock and wherein the service node generates a residual time stamp (RTS) signal equal to a modulo 16 count of cycles based on the network clock; the apparatus comprising:

means for receiving the RTS signal;

means for determining the number of network cycles in an RTS period from the RTS signal; and

means responsive to the determining means for generating a clock signal which represents a recovery of the timing clock of the service input.

13. Apparatus for generating a representation of a timing clock of a service input at a source node of a packet-based telecommunications network, wherein a common network clock or divided network clock is provided for the source node and a destination node; the apparatus comprising:

(a) means for defining a time interval by a fixed number of service clock cycles; and

(b) means for generating a digital representation of a quantized difference between an actual number of network clock cycles within the time interval and an expected number of network clock cycles within the time interval, the difference being within a defined time window corresponding to a frequency variation of the source-node service clock.

14. The apparatus of claim 13, wherein the digital representation represents a chosen number of the least significant bits of the quantized actual number, the chosen number being sufficient to represent a range of frequency deviations of the source-node service clock variation.

15. The apparatus of claim 14 wherein the chosen number is 4.

16. Apparatus for recovering, at a destination node of a packet-based telecommunications network, the timing clock of a service input at a source node of said network, wherein a common network clock or divided network clock is provided for the destination node and the source node and a time interval is defined by a fixed rational number of source-node service clock cycles; the apparatus comprising:

means for receiving generating a digital representation of a quantized difference between an actual number of network clock cycles within the time interval and an expected number of network clock cycles within the time interval, the difference being within a defined time window corresponding to a frequency variation of the source-node service clock; and

means for recovering the source-node service clock at the destination node by constructing a timing signal at the destination node based on a received representation of the network cycle difference.

17. Apparatus for reconstructing, at a destination node of a packet-based telecommunications network, a timing clock of a service input at a source node of the network, wherein a common network clock or divided network clock is provided for the destination node and the source node and wherein the reconstruction is based on successive modulo 2^P numerical representations of the number of network clock cycles within corresponding successive predetermined time periods, each of the numerical representations being received from the source node and being less than the actual number of network clock cycles within its corresponding time period; the apparatus comprising:

means for receiving the numerical representations in succession at the destination node;

means for converting the received numerical representations into successive fixed time intervals, wherein each successive interval corresponds to the number of network clock cycles in a corresponding one of the predetermined time periods; and

means for recovering the source-node service clock from the fixed time intervals.

18. The apparatus of claim 17, wherein the converting means further comprises:

means for sequentially storing the successive modulo 2^P numerical representations;

means for comparing the successive numerical representations with a modulo 2^P count of the network clock cycles at the destination node to generate a comparison signal for each match between the numerical representation and the modulo 2^P count at the destination node; and

means for successively selecting a proper comparison signal by waiting until a minimum number of network clock cycles has occurred.

19. A method for generating a signal at a source node for use in recovering a source-node service clock at a destination node in a packet-based telecommunications network, wherein a common network clock or divided network clock is provided for the source and destination nodes; the steps of the method comprising:

defining a time interval by a fixed number of cycles of the source-node service clock;

determining an actual number of cycles of the network clock within the time interval;

determining a numerical deviation of the number of actual network clock cycles from another number of network clock cycles that would occur if the source-node service clock frequency were nominal; and

generating a digital signal representing the numerical deviation for transmission through the network to the destination node.

20. A method for recovering a source-node service clock at a destination node in a packet-based telecommunications network, wherein a common network clock or divided network clock is provided for the source and destination nodes, wherein an actual time interval is defined by a fixed number of cycles of the source-node service clock, and wherein a number of actual cycles of the network clock within the actual time interval and a numerical deviation of the number of actual network clock cycles from another number of network clock cycles known nominally to be within the time interval are determined; the steps of the method comprising:

receiving a digital signal representing the numerical deviation transmitted through the network from the source node; and

generating a timing signal corresponding to the source-node service clock on the basis of the digital signal representing the numerical deviation.

21. A method for recovering, at a destination node of a packet-based telecommunications network, a timing clock of a service input at a source node of the packet-based telecommunications network, wherein a common network clock or divided network clock is provided for the destination node and the source node; the steps of the method comprising:

defining a time interval by a fixed number of cycles of the source-node service clock;

determining an actual number of cycles of the network clock within the time interval;

determining a numerical deviation of the number of actual network clock cycles from another number of network clock cycles that would occur within the time interval if the source-node service clock frequency were nominal;

generating a digital signal representing the numerical deviation;

transmitting the digital signal to the destination node; and

generating a timing signal at the destination node corresponding to the source node service clock on the basis of the digital signal and a signal from the network clock.

22. The method of claim 21 wherein the numerical deviation is determined as a function of the fixed number of source-node service clock cycles, the frequencies of the network clock and the source-node service clock, and a frequency variation of the source-node service clock.

23. The method of claim 21 further including the step of employing a modulo 2^P counter to generate a representation of the numerical deviation.

24. Apparatus for generating a signal at a source node for use in recovering a source-node service clock at a destination node in a packet-based telecommunications network, wherein a common network clock or divided network clock is provided for the source and destination nodes; the apparatus comprising:

means for defining a time interval by a fixed number of cycles of the source-node service clock;

means for determining a number of actual cycles of the network clock within the time interval;

means for determining a numerical deviation of the number of actual network clock cycles from another number of network clock cycles that would occur within the time interval if the source-node clock frequency were nominal; and

means for generating a digital signal representing the numerical deviation for transmission through the network to the destination node.

25. The apparatus of claim 24 wherein the numerical deviation is determined as a function of the fixed number of source-node service clock cycles, and frequencies of the network clock and the source-node service clock, and a nominal frequency of the source-node service clock.

26. The apparatus of claim 24 wherein the numerical deviation determining means includes a modulo 2^P counter which generates the numerical deviation.

27. The apparatus of claim 26 wherein a value of 2^P is 16.

28. Apparatus for recovering a source-node clock at a destination node in a packet-based telecommunications network, wherein a common network clock or divided network clock is provided for the source and destination nodes and wherein a time interval is defined by a fixed number of cycles of the source-node service clock, and wherein a number of actual cycles of the network clock within the time interval and a numerical deviation of the number of actual network clock cycles from another number of network clock cycles that would occur within the time if the source-node service clock frequency were nominal;

means for receiving a digital signal representing the numerical deviation transmitted through the network from the source node; and

means for generating a timing signal corresponding to the source-node service clock on the basis of the digital signal representing the numerical deviation.

29. Apparatus for recovering, at a destination node of a packet-based telecommunications network, a timing clock of a service input at a source node of the packet-based telecommunications network, wherein a common network clock or divided network clock is provided for the destination node and the source node; the apparatus comprising:

means for defining a time interval by a fixed number of cycles of the source-node service clock;

means for determining a number of actual cycles of the network-clock within the time interval;

means for determining a numerical deviation of the number of actual network clock cycles from another number of network clock cycles that would occur within the time interval if the source-node service clock frequency were nominal;

means for generating a digital signal representing the numerical deviation;

means for transmitting the digital signal to the destination node; and

means for generating a timing signal at the destination node corresponding to the source-node service clock on the basis of the digital signal and a signal from the network clock.

30. The apparatus of claim 29 wherein the numerical deviation is determined as a function of the fixed number of source-node service clock cycles, frequencies of the network clock and the source-node service clock, and a nominal frequency of the source-node service clock.

31. The apparatus of claim 29 wherein the numerical deviation determining means includes a modulo 2^P counter which generates the numerical deviation.

32. The apparatus of claim 29 wherein means are provided for carrying any fractional network cycle in any time interval for network cycle counting by the modulo 2^P counter for counting the next time interval.

33. A method for generating a representation of the relationship between the timing clock of a service input, at a source node of a packet-based telecommunications network, and a network clock, the method comprising the steps of:

(a) defining, at the source node, a residual time stamp (RTS) period as an integral number N of source-node service clock cycles;

(b) defining, at the source node, a derived network clock frequency f_{nx} from a network frequency f_n where $f_{nx}=f_n/x$, x is a rational number, and f_{nx} is less than or equal to twice the service clock frequency;

(c) counting, at the source node, the derived network clock cycles modulo 16 in an RTS period; and

(d) transmitting from the source node an RTS that is equal to the modulo 16 count of derived network clock cycles in the RTS period.

* * * * *