PCT WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 5 :		(11) International Publication Number:	WO 91/07038
H04L 29/06	A2	(43) International Publication Date:	16 May 1991 (16.05.91)

(21) International Application Number: PCT/US90/05787

(22) International Filing Date: 10 October 1990 (10.10.90)

(30) Priority data:
431,595 3 November 1989 (03.11.89) US

(71) Applicant: MICROCOM SYSTEMS, INC. [US/US]; 500 River Ridge Drive, Norwood, MA 02062-5028 (US).

(72) Inventors: PEARSON, Gregory ; 12955 Jolette Avenue, Granada Hills, CA 91344 (US). MELHORN, Nathan, Richmond ; 50 Dinsmore Avenue, #516, Framingham, MA 01701 (US). ONARATO, Michael, Francis ; 2 Simon Willard Road, Acton, MA 01720 (US). RICHARDS, Craig, Allen ; 55 Lake Street, Wrentham, MA 02093 (US).

(74) Agent: MIRABITO, A., Jason; Wolf, Greenfield & Sacks, 600 Atlantic Avenue, Boston, MA 02210 (US).

(81) Designated States: AT (European patent), AU, BE (European patent), CA, CH (European patent), DE (European patent), DK (European patent), ES (European patent), FR (European patent), GB (European patent), GR (European patent), IT (European patent), JP, LU (European patent), NL (European patent), SE (European patent).

Published
Without international search report and to be republished upon receipt of that report.

(54) Title: METHOD AND APPARATUS FOR EFFECTING EFFICIENT TRANSMISSION OF DATA

(57) Abstract

A data communications system and method is provided for sending a data stream of characters in distinct data packets between two units of data terminal equipment connected over communications lines. A modem is connected between each unit of data terminal equipment and the communications lines, and one initializing modem includes a means for changing the packet size of the data which is transmitted. Depending on the response of the receiving modem on whether errors are detected, the size of the data packets is dynamically adjusted. If an error is detected, the data packet size is decreased, and if no error is detected, then the data packet size is increased. If there is no data packet to send, then an idle packet is sent. If the idle packet is received without error, then the packet size of the data packets is increased. If, however, there is an error in the idle packet received, then the size of the data packet will be decreased.

FOR THE PURPOSES OF INFORMATION ONLY

Codes used to identify States party to the PCT on the front pages of pamphlets publishing international applications under the PCT.

AT	Austria	**ES**	Spain	**MG**	Madagascar
AU	Australia	**FI**	Finland	**ML**	Mali
BB	Barbados	**FR**	France	**MR**	Mauritania
BE	Belgium	**GA**	Gabon	**MW**	Malawi
BF	Burkina Faso	**GB**	United Kingdom	**NL**	Netherlands
BG	Bulgaria	**GR**	Greece	**NO**	Norway
BJ	Benin	**HU**	Hungary	**PL**	Poland
BR	Brazil	**IT**	Italy	**RO**	Romania
CA	Canada	**JP**	Japan	**SD**	Sudan
CF	Central African Republic	**KP**	Democratic People's Republic of Korea	**SE**	Sweden
CG	Congo			**SN**	Senegal
CH	Switzerland	**KR**	Republic of Korea	**SU**	Soviet Union
CI	Côte d'Ivoire	**LI**	Liechtenstein	**TD**	Chad
CM	Cameroon	**LK**	Sri Lanka	**TG**	Togo
DE	Germany	**LU**	Luxembourg	**US**	United States of America
DK	Denmark	**MC**	Monaco		

METHOD AND APPARATUS FOR EFFECTING EFFICIENT TRANSMISSION OF DATA

BACKGROUND OF THE INVENTION

This invention relates generally to a data telecommunications system, and more particularly to a system which transmits data in packets, the size of which may be changed based on transmitting conditions. The invention also relates to the efficient establishment of a data transfer link between two microcomputers over telephone lines. The subject matter of the present invention is particularly effective in data transfers in which the line quality may be poor or may rapidly change from good quality to poor quality. An example of this cellular communications, which presents widely varying degrees of line quality, although cellular communications is not the only medium in data communications in which line quality may be poor or may vary. Telephone lines also possess such qualities.

Whereas microcomputers were once only used as dedicated and completely isolated devices, they are now used for a wide range of applications, many of which require microcomputers to communicate with each

other. Problems develop with such communication because of the existence of a wide variety of microcomputers which are incapable of directly communicating with each other. As a result, many data communication systems for microcomputers force the microcomputer to work principally as a dumb terminal for a remote mini- or mainframe computer. Therefore, to facilitate more sophisticated communication, communication protocols which enable data communications between a wide range of computers, from low end microprocessors to large mainframes, have been developed. The proliferation of protocols, however, not to mention operating systems and microprocessors, has for the most part precluded the adoption of a standard file transfer method.

A communication protocol is basically a set of rules that defines how the computers interact. For two computers to transfer a file successfully, they must observe the same protocol. Typically, a protocol specifies when to send a message, how to format the information in the message, and at the other end, how to acknowledge the receipt of the message.

Simple physical connect protocols are concerned only with hardware configurations. Establishing the basic physical connection between two computers requires that a particular series of steps be followed. The originating modem initiates its sending sequence, and the telephone number

representing the electronic address of the receiving modem is formatted as a series of pulses of tones and sent into the telephone network. The receiving modem senses the incoming call as a relatively high voltage (sufficient to cause a phone to ring) and interprets this as a request to establish a connection. If the connection is established, an acknowledgement message sent to the transmitting modem thereby establishes the connection. Such a physical link is capable of being established because both modems use the same physical connect protocol. These basic connect protocols are fairly standard, particularly for the low speed, synchronous modems commonly used with microcomputers, such as Bell 103 and 212A modems. However, as communications become more complex and involve dissimilar machines and file formats, the protocols must, in turn, become increasingly sophisticated.

Connecting two computers is only a small part of the communications work necessary for accurate data transfer. Telephone lines are often noisy, and errors can crop up in the transmitted data. These errors must be detected and corrected. The resources available to store incoming data must also be passed and matched so that the recipient is not flooded with data. Also, information regarding the computer file system must be swapped and file movement and manipulation must be defined. These concerns, therefore, go beyond the physical-level protocols into the realm of complex communications protocols.

One protocol which enables complex communications between microcomputers is the Microcom Networking Protocol (MNP) which has been developed by Microcom, Inc. of Norwood, Massachusetts. MNP provides a sophisticated communications system which includes provisions for both reliable terminal-type communications and reliable file transfer in a manner which can reasonably be implemented on a wide range of computers. MNP accounts not only for hardware and operating system differences, but also provides sophisticated error checking. As a result, file exchanges are thus possible between almost any computers using an MNP based communication system.

The MNP protocol, which has been developed principally for use with microcomputers, includes three layers, and the use of only three layers enables MNP to provide the necessary services with the desired space and performance characteristics for a microcomputer environment. The three layers or modules are combined to perform a series of complex functions in a manner in which changes in one module may not drastically affect another module, as long as certain parts of the module's interface remain the same.

In MNP, each layer is relatively isolated and provides a specific service. If a change is forced in one layer (for example, if MNP is modified for use on a new computer), the change is confined to that layer while the layer's standard interface to the other layers remains unchanged. In addition to

ensuring machine portability, MNP's structure allows services provided by one layer to support those in the layer above. The accumulation of services is then passed upward, from layer to layer to the applications program. MNP defines three unique protocol layers in addition to the physical connection; the link, the session, and the file protocol layers. The protocol layers are triggered sequentially from the bottom (physical) to the top (file transfer).

The link layer is responsible for providing reliable, controlled data transmission over a medium that is inherently noisy and likely to cause errors. Once a physical connection is established between two machines, the link protocol acts as a negotiator causing both computers to agree on the nature of the link. For example, the link protocol establishes whether the connection will be half- or full-duplex, how many data messages can be sent before confirmation is required, the size of a single data packet, etc. After establishing values for the above requirements, the link protocol initiates data transfer, paces the flow of data and, if necessary, re-transmits data messages that contain errors due to telephone line noise. The link protocol allows blocks or packets of data (as opposed to individual bytes) to be sent synchronously or asynchronously to the receiving computer. Data transfer is faster when packets are transmitted synchronously because start and stop characters are not needed, and as a result,

the ratio of data to control characters regulating the transfer is higher. Control is possible because of a mainframe-like (framing) technique in which a block of data is carried from both ends with specific codes.

The session layer negotiates with the receiving computer with respect to the pertinent system and file information including computer type, how files are formatted, the type of information transmitted (e.g. ASCII, binary) and the user's identity. This layer also provides the automatic negotiation of which level of service can be used between the two communicating devices.

The file transfer layer defines and formats the messages involved in file transfers and manipulations. There are three transfer services available: one allowing the sending of a file, the second allowing the receipt of a file, and the third allowing the appending of a file to an existing file at the other end of the communications link. The file transfer protocol also enables the manipulation of distant computer files. For example, such files can be deleted, renamed or file directories may be displayed. A typical file transfer starts when one computer sends a "file start" message to the other computer. The file start message includes the requested file's name, size and format along with any password needed for the file's return trip to the requesting computer. Both computers exchange "hellos" along with a confirmation that a file will be soon filed by one side and accepted by the other.

In order for any communications protocol to facilitate communications among a wide variety of computers, the protocol must be able to operate in a number of modes. These modes include a matched-protocol mode for use by two communicating devices supporting the same protocol. Such a matched-protocol mode may provide optimized data transmission including any of a number of known optimizing features such as detecting and correcting errors, optimizing transmission speed, etc. A second mode which allows straight forward data transmission (without any optimizing features) between two communicating systems must also be provided. The operating mode is generally negotiated in a lower link layers, and in MNP the mode is selected in the link layer.

A modem operating under the MNP protocol discussed above has four basic modes: reliable mode, normal mode, auto-reliable mode, and direct mode. The reliable mode is the basic matched-protocol mode of MNP which provides error detection and automatic re-transmission of data when an error occurs in order to ensure that communications between two communicating systems are error-free. In order to utilize this reliable mode, however, both of the communicating modems must be able to support this mode. A second mode, MNP's normal mode, allows a modem with MNP to communicate with a modem not supporting this protocol. MNP's auto-reliable mode is an extension of the reliable mode which will

automatically connect two modems in a reliable mode if such a connection is possible. In other words, if both of the communicating modems can support MNP, a reliable, error-correcting connection is established. The auto-reliable mode differs from the reliable mode insofar as the modem in the auto-reliable mode initially looks for incoming MNP protocol characters from the remote modem. If, after a predetermined amount of time has expired, these MNP characters are not detected, a normal link is established. If these characters are detected, a reliable link is established. A fourth mode, the direct mode, is used for special purpose formats or character sets, and the modem buffers and flow control command settings are ignored.

A fifth mode, a subset of auto-reliable mode, called the special or smart auto-reliable mode, contains a special or "fallback" character or set of characters which indicate that a non-matched protocol should be established. The features of the special or smart auto-reliable mode are described and claimed in United States Patent No. 4,680,773 for "Data Communications System and Method Utilizing a Multi-Mode Modem", assigned to the assignee of the present invention, the disclosure of which is hereby incorporated by reference. One aspect of the present invention is a subset of both auto-reliable and special auto-reliable modes, and is especially effective when telephone line quality may have deteriorated. This mode, entitled Robust

Auto-Reliable mode, is explained in greater detail below.

Data sent from one modem to another is usually transmitted as part of a data packet, the size of which is determined by the link layer. Data packets generally include, in addition to the data, header and trailer data which among other purposes identifies the beginning (header) and end (trailer) of each data packet. The header and trailer information may also be used for indicating the type of packet (ex. acknowledgement, control, data packet), the CRC code, and any other identifying information. While the data packets are very useful for maintaining error-free transmission, the header and trailer information increases the amount of data transmitted and as a result data throughput is reduced.

In order to increase the data transmission rate or throughput, many techniques involving the manipulation of the transmitted data are utilized. One such technique is to increase the size of the packets of data which are sent by a modem. Such increased packet size produces an increase in throughput because there is less header and trailer information sent since fewer packets are actually transmitted. United States Patent No. 4,691,314, owned by the assignee of the present invention, discloses and claims a system for transmitting data in adjustable-sized packets, of which the present invention is an improvement upon.

While ideally such increased packet size should speed transmission, the nature of the transmitting environment may lead to the opposite result. Since telephone lines are often noisy (present, for example, in cellular communications in which not only may line quality be poor, but also that it may be erratic and rapidly fluctuate between good and poor line quality) and other hardware problems produce errors in the transmitted data, it is often necessary when using an error-correcting protocol to re-transmit many packets of data which contain data errors. The re-transmission of larger sized packets thereby results in unnecessary transmissions of data because for each altered transmitted character many more correctly transmitted characters must be re-transmitted thereby decreasing throughput. Therefore, if many data packets are re-transmitted, smaller packet sizes are actually beneficial since less data has to be re-transmitted.

Known modems which transmit data in packet are designed so that the packet size is chosen which compromises the advantages of large packet sizes with the advantages of small packets. The nature of the transmitted data, as well as the transmitting conditions, however, frequently change so that any particular packet size will, in reality, not be optimum for all situations.

It is therefore a principal object of the present invention is to provide a data transmission system and method in which the packet size of the

data being sent by a modem over telephone lines is optimized.

Another object of the present invention is to provide a data communication system and method in which the data packet size can be optimized in a real time as a function of the data stream being transmitted as well as the transmitting environment using additional idle packets.

Still another object of the present invention is to provide a data communications system and method in which data may be transmitted in any of a number of packet sizes.

Yet another object of the present invention is to provide a data communications system and method in which an additional mode of operation may be selected to connect two modems over telephone lines which is especially effective for conditions when line quality may not be optimum.

SUMMARY OF THE INVENTION

In accordance with the present invention, a data telecommunications system and method is provided for sending in a data stream of characters in distinct data packets between two units of data terminal equipment (which can be either terminals and/or computers) connected over communication lines (the Aggressive Adaptive Packet Sizing mode). A modem is connected between each unit of data terminal equipment and the communication lines, and one

initiating modem includes a means for changing the packet size of the data which is transmitted. The receiving modem checks the packets for errors, and if errors are found, the packet is re-transmitted by the initiating modem.

The initiating or transmitting modem observes each data packet transmitted with and without error(s) and dynamically changes each packet size. The transmitting modem also sends idle packets to the receiving modem during intervals in which there is no data transmission and likewise observes whether error(s) occur in the transmission of such idle packets. The line quality is determined in real time so that a change in the transmitting environment will result in a change in the packet size.

In addition, the present invention includes an additional mode (the Robust Auto-Reliable mode) under which the modem may operate, particularly when line quality is not optimum. This new mode features the sending of multiple link requests to a receiving modem and, upon detection by the receiving modem of what might be, but is not certain to be, a link request, the sending modem adds a further predetermined amount of time over and above the amount of time present in auto-reliable mode to allow the receiving modem to detect further multiple incoming MNP protocol characters. As a result, when under the prior system of auto-reliable mode an MNP connection may not have been made due to poor line quality, under this aspect of the invention, such connection will be made.

These and other objects and features of the present invention will be better understood from the following detailed description which should be read in light of the accompanying drawings in which corresponding reference numerals refer to corresponding parts throughout the several views.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a generalized schematic view of a data telecommunications system;

Fig. 2 is a schematic view of a modem of the present invention;

Fig. 3 is a flow diagram of the steps utilized by the modem of the present invention in optimizing the size of the transmitted data packets;

Figs. 4 and 4A are flow diagrams of the steps utilized in the modem of the present invention to provide an auto-reliable connection in non-optimum line quality conditions.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

A basic data telecommunications system, shown in Fig. 1 includes an initiating unit of data terminal equipment (DTE) 10, such as, but not limited to, a dumb terminal or a microcomputer, and a receiving unit of DTE 12. An initiating modulator/demodulator (modem) 11 is connected between unit 10 and communication lines (such as, but not limited to

telephone lines), and a receiving modem 13 is connected between unit 12 and communication lines.

A modem of the communication system of the present invention is shown in Fig. 2. The system of the Aggressive Adaptive Packet Sizing mode of the present invention will normally include at least two modems of the type described below. While the system of the Robust Auto-Reliable mode may include only one modem of the type described below, it is preferable that both modems be of this type. For purposes of the following discussion, the modem of the system will be described with reference to modem 13, although the modem 11 for the purposes of this disclosure may be treated as containing identical circuitry and capabilities. The modem 13 includes a DTE-interface 14 which receives data coming from DTE unit 12. Data characters supplied to the DTE-interface 14 pass through a serial communications port 16 of the microprocessor 18 to which characters are fed in a serial fashion. The microprocessor 18 has connected to it status indicators 20, a program and data memory 22 and parameter setting switches 24. Timing synthesizing circuitry 26 is also connected to the microprocessor 18. Data processed by the microprocessor 18 is sent through a serial modem port 28 to modulation circuits 30 which will pass data on through a filter 32 to a telephone interface 34 and then on communication lines over which the data will be transmitted to another microcomputer. The modem 13 also includes

demodulation and data recovery circuits 38 which are used for receiving data from another remotely situated modem, such as modem 11. When the modem 13 acts as a receiving modem, data passes through the same elements described above in a reverse order following the passage of data through the demodulation and data recovery circuits 38.

A modem of the data telecommunications system of the present invention optimizes the size of the data packets taking into account the number of re-transmission of data packets and idle packet errors. The optimizing or adapting of the packet size is preferably performed in real time so that changes in the transmitting environment will result in changes in packet size, thereby maintaining an optimum throughput. Idle packets will be sent when there is no data from associated DTE 10 to send to modem 13, and are used to give modem 11 an idea of the line conditions when there are no data packets to be used to determine line quality. Referring to Fig. 3, in order to enable a modem to optimize or adapt the packet size of transmitted data, a matched protocol link that has negotiated a class that supports the adaptive packet sizing must first be negotiated, as shown in step 120. In step 122, the modem 11 must initialize maximum packet size, as well as initializing and starting an idle transmission timer to determine when the transmitter is idle. If there is data from DTE 10, the modem 11 will check to see if there were errors in any previously transmitted packets in step 126.

This error checking is performed by the receiving modem 13, which will request a re-transmission if it has missed a data packet. If modem 13 has requested a re-transmission, the modem 11 will perform step 132 to decrease the maximum packet size according to formula 2. It should be noted that the maximum does not apply to packets which must be re-transmitted. Then modem 11 will re-transmit any packets sent that were not correctly received by modem 13. After these packets are re-sent in step 134, program control will return to step 124. If modem 13 has not requested any re-transmissions, modem 11 will perform step 128 to increase the maximum packet size according to formula 1. After this increase in packet size is performed, modem 11 will transmit data from DTE 10 in a data packet to modem 13. When this packet is sent, modem 11 will then loop back and perform step 124.

If there is no data with which to build packets in step 124, the modem 11 will go to step 136 and determine if the idle timer has timed out. If the idle timer is still running, modem 11 will loop around to step 124 and check for data from DTE 10 again. If idle timer has in fact timed out in step 136, then modem 11 will send an idle packet. The purpose of this packet is to transmit characters to modem 13 so that modem 11 will know if there exist line conditions which would case errors in its transmitted data. Each idle packet will carry a sequence number that modem 13 will use to determine

if it has received all the idle packets in order. If modem 13 has not received a packet in order, it will inform modem 11 about this. After modem 11 has transmitted its idle packet in step 138, it will re-initialize and restart its idle timer. Then it will check if the remote modem 13 has received an idle packet out of order. If there has been an idle packet error, modem 11 will decrease its data packet size in accordance with formula 4 shown in step 146. Otherwise, if step 142 determines that the idle packets were received without error by modem 13, modem 11 will perform step 144, increasing data packets size using formula 3. After steps 144 or 146, the program will return to step 124.

The format of the formulas is data packet size = data packet size multiplied by a constant. The data packet size is then tested to determine if it is less than the minimum data packet size or greater than the maximum size. If it is outside the acceptable range, it is then set to the minimum or maximum size, as appropriate. The constant in the formula is defined separately for each formula as described below. By adjusting the value of the constant, the rate of change of packet size can be adjusted. Some consideration needs to be given to the relative value of the constants, so that the packet size will not react too quickly to errors and take too long to recover from them.

The formula used to increase packet size is as follows:

PS = MAX [(MIN[PS' x X], [PSMAX])], [PSMIN]

The formula used to decrease packet size is as follows:

PS = MAX [(MIN[PS' x Y], [PSMAX])], [PSMIN]

Where:

PS = New Packet Size
PS'= Previous Packet Size
X = Increasing constant
Y = Decreasing constant
PSMAX = Maximum Packet Size
PSMIN = Minimum Packet Size
MIN[X1], [X2] = Use Minimum of X1 and X2
MAX[X1], [X2] = Use Maximum of X1 and X2

Formulas 1 and 3 are special cases of the formula used to increase packet size. It is possible, but not necessary, to set the constant X for formulas 1 and 3 to be identical. The value of constant X must be greater than 1 to insure that packet size increases. The value of constant X depends on many factors, including the physical link used and on the matched protocol used. In the example of FIG. 3, suitable values could be PSMAX = 256, PSMIN = 8, an initial packet size of 16, and a range for constant X from 1.005 to 1.120.

Formulas 2 and 4 are special cases of the formula used to decrease packet size. It is possible, but not necessary, to set the constant Y

for formulas 2 and 4 to the same value. If constant Y is set the same, formulas 2 and 4 would be identical. The value of constant Y must be less than 1 to insure that packet size decreases. The value of constant Y depends on many factors, including the physical link used and on the matched protocol used. In the example of FIG. 3, suitable values could be PSMAX = 256, PSMIN = 8, an initial packet value size of 16, and a range for constant Y from 0.50 to 0.95.

Another aspect of the present invention is the provision of the Robust Auto-Reliable mode referred to above. For purposes of the following discussion, the "auto-reliable" mode refers to a mode of operation of a modem in which a matched-protocol link is automatically established if both modems are capable of operating under such a matched-protocol. If both modems cannot operate under the matched-protocol, a normal mode connection is automatically established.

Referring now to Figs. 4 and 4A, a modem capable of operating under an auto-reliable mode is enabled for operation in the auto-reliable mode 50. A special or "fallback" character or set of characters that indicate that a non-matched-protocol link should be established is set in step 52. Such characters are generally the first characters sent by a computer of the type which will attempt to establish a connection with receiving unit 12. When a call is received by a receiving modem 13, a physical connection is established in step 54. The receiving

modem 13 is initialized for matched protocol receiver operation in step 56. In step 60, the modem 13 initializes and starts a timer to time the auto-reliable operation, in addition to resetting the matched protocol character and loop flags. The receiving modem 13 then enters a loop, in which it constantly checks if the auto-reliable timer set in step 60 has timed out or if a character has been received. If the auto-reliable timer times out in step 62, the modem 13 will then check if the matched protocol character received flag has been set in step 81. If that flag is set, modem 13 checks if the loop flag is set in step 82. If the loop flag is set, then step 83 causes modem 13 to establish a non-matched protocol link. If the loop flag is not set in step 82, step 84 sets the loop flag, in addition to re-initializing and restarting the auto-reliable timer. The modem 13 loops back to step 62. If, in step 81, the matched protocol flag is not set, then step 83 is executed, in which modem 13 sets up a non-matched protocol connection.

If in step 62 the auto-reliable timer has not timed out, modem 13 will check if the character has been received from modem 11 in step 63. If a character has not been received, the modem 13 loops around to step 62. If a character has been received from the initiating modem, step 64 is executed. If the character necessary for initiating a matched protocol link is detected in step 64, the modem 13 will set the matched protocol character received flag

in step 85. If the matched protocol link negotiation parameters are correctly received from modem 11 in step 86, then a matched protocol link is established in step 87. If the matched protocol link negotiation is incorrect in step 86, then the modem 13 returns to step 62.

If, in step 64, the necessary character for signaling a desire to establish a matched protocol link is not detected, the modem 13 then checks, in step 106, whether the modem is enabled to operate in buffered auto-reliable mode. If the modem 13 is not enabled for the buffered auto-reliable mode, then the modem will advance to step 116 to check if the matched protocol character has been received. If, on the other hand, the buffered auto-reliable mode is enabled, the character is placed in a character buffer 23 in the program and data memory 22 of the modem 13. In step 110 the modem determines whether the buffer 23 has been filled, and if buffer 23 is not filled, then control shifts to step 116, where it checks the matched protocol character flag. If the character buffer 23 has been filled, the modem initializes itself to operate under a non-matched protocol mode in step 112, and in step 114 the buffered characters are transmitted through serial port 16 and DTE-interface 14 to the terminal or computer 12 to which modem 13 is connected. At this point, communications will continue with the modem operation in a non-matched protocol mode.

In step 116, the matched protocol received flag is checked to determine if the beginning of the

matched protocol negotiation had been attempted. If this character had been received, then control returns to step 62 to once again check the auto-reliable timer. Otherwise, if no matched protocol character has been received, step 68 will be executed.

In step 68, the modem 13 checks whether the modem is enabled to operate in smart or special reliable mode and thereby recognize the fallback character or set of characters indicating the desire of the initiating modem 11 to establish a non-matched protocol link. If the modem 13 is not enabled to recognize the fallback character(s), then the program control will be passed to step 62.

If the modem 13 is enabled to detect a specific character for establishing a non-matched-character mode connection, then in step 70, the most significant bit is masked off the data character under examination. The reason for this masking off of the most significant bit is that the most significant bit serves as a parity bit and is not needed for determination of the operating mode.

After masking off the most significant bit in step 70, the modem 13 will determine in step 72 whether the character currently under examination is actually the fallback character. If the examined character is not the fallback character, then control returns to step 62, and the auto-reliable timer is checked to determine if there is time remaining on it. If, on the other hand, the fallback character is

detected, the modem 13 initializes itself for operation under a non-matched protocol mode in step 74, and the fallback character is restored by replacing the most significant bit in step 76. The restored character is then sent to the computer or terminal through the serial communications port 16 in step 78. From this point onward, the modem 13 will operate in a non-matched-protocol mode without any error detection or other special features normally associated with the matched-protocol mode.

While the foregoing invention has been described with reference to its preferred embodiments, various modifications and alterations will occur to those skilled in the art. All such modifications and alterations are intended to fall within the scope of the appended claims.

What is claimed is:

1. A method of transmitting data in packets in a data telecommunications system, said system including an initiating unit of data terminal equipment and a receiving unit of data terminal equipment, a first modem connected between the initiating unit and communication lines and a second modem connected between said receiving unit and said communication lines, said method comprising the steps of:

transmitting a stream of data characters from said initiating unit to said first modem for transmission over communications lines to said second modem;

dividing said data stream received by said first modem into packets of data with each packet including identifying data;

transmitting one data packet from said first modem to said second modem over communication lines;

checking data in said transmitted data packets for errors, and when an error is detected re-transmitting the data packet;

decreasing and increasing the size of data packets dependent upon the detection of errors;

sending idle packets in the event there are no data packets to send from said first modem to said second modem;

checking the idle packets transmitted for errors; and

decreasing and increasing the size of data packets responsive to detection of errors in the idle packets.

2. A method of initiating data telecommunications between two units of data terminal equipment over communications lines, each unit of said data terminal equipment having a modem connected between the unit and the communications lines, at least one of said modems being capable of operating under a special mode and a normal mode, said special mode providing optimized data transmission that paces the flow of data between said two modems and provides error detection in a transmitted data stream by a receiving modem and automatic re-transmission of said data stream by an initiating modem when an error occurs and said normal mode providing direct, non-optimized data transmission that provides transmission of the data stream without any analysis of the data stream, said method comprising the steps of:

(a) establishing a physical connection between said initiating modem and said receiving modem;

(b) transmitting a stream of data characters from said initiating modem to said receiving modem over communication lines;

(c) examining in said receiving modem a next character in said data stream transmitted by said initiating modem to said receiving modem;

(d) attempting to establish a special mode link between said initiating and receiving modems if said next character examined by said receiving modem is a special mode indicating character and attempting to establish a normal mode link if said character is a normal mode indicating character;

(e) repeating steps (c) through (d) after fetching a new data character from the data stream until either a sequence of special mode indicating characters or a normal mode indicating character is detected or until a preselected period of time has elapsed;

(f) incrementing the preselected period of time by additional preselected periods of time upon detection of the first of a sequence of special mode indicating characters but not others of the sequence in order to detect either said special mode indicating character or said normal mode indicating character;

(g) continuing the transmission by said initiating modem of said stream of data if a special or normal mode link is established until said stream is transmitted.

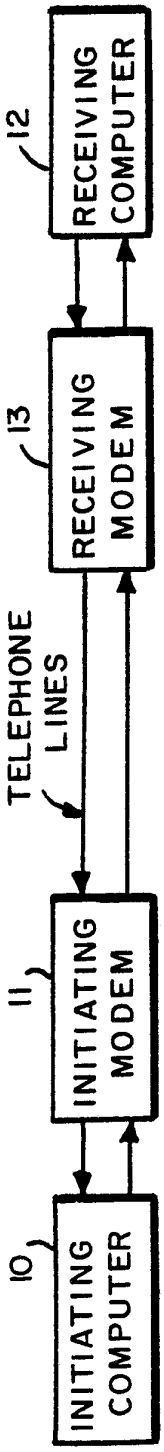
INITIATING COMPUTER
10
INITIATING MODEM
11
TELEPHONE LINES
RECEIVING MODEM
13
RECEIVING COMPUTER
12

FIG. 1

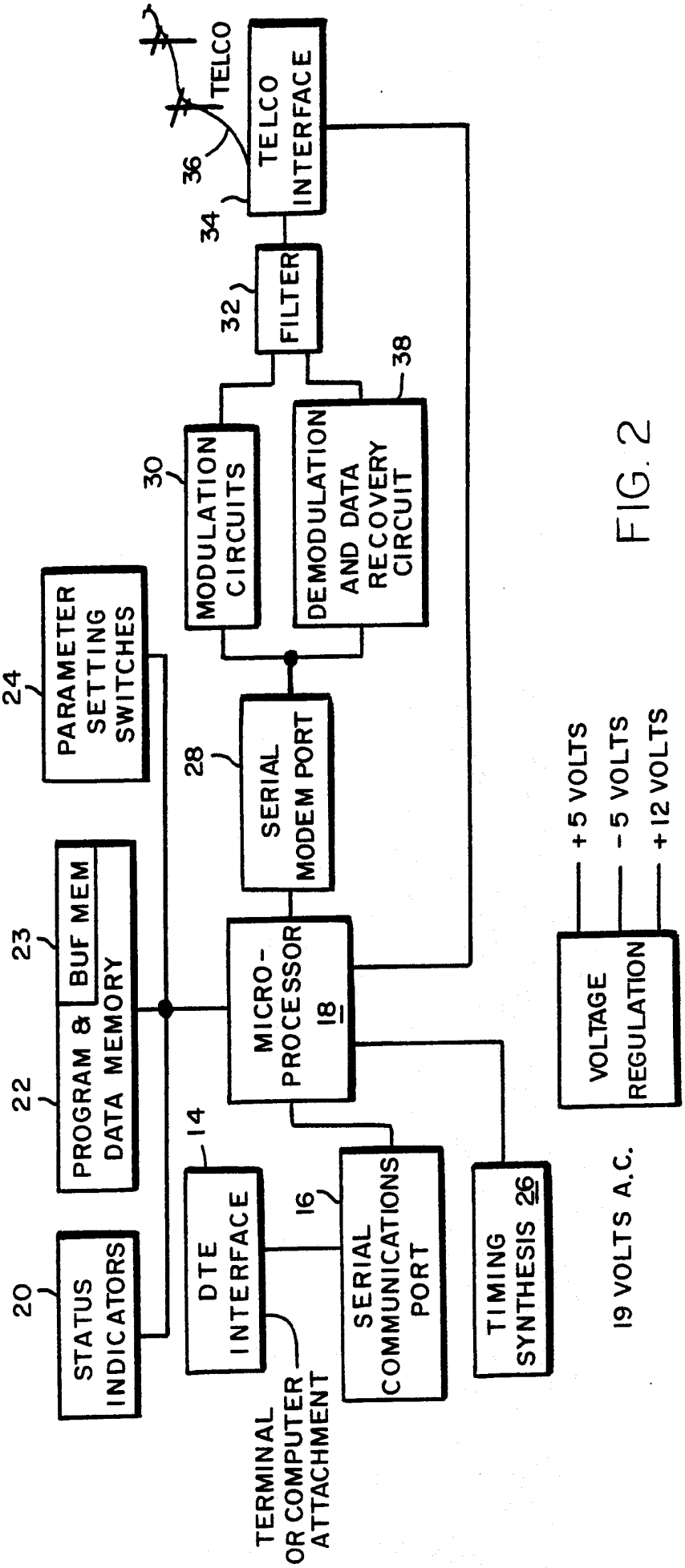
TELCO
36
TELCO INTERFACE
34
FILTER
32
MODULATION CIRCUITS
30
DEMODULATION AND DATA RECOVERY CIRCUIT
38
PARAMETER SETTING SWITCHES
24
SERIAL MODEM PORT
28
BUF MEM
23
PROGRAM & DATA MEMORY
22
MICRO-PROCESSOR 18
STATUS INDICATORS
20
DTE INTERFACE
14
16
SERIAL COMMUNICATIONS PORT
TIMING SYNTHESIS 26
TERMINAL OR COMPUTER ATTACHMENT
+5 VOLTS
-5 VOLTS
+12 VOLTS
VOLTAGE REGULATION
19 VOLTS A.C.

FIG. 2

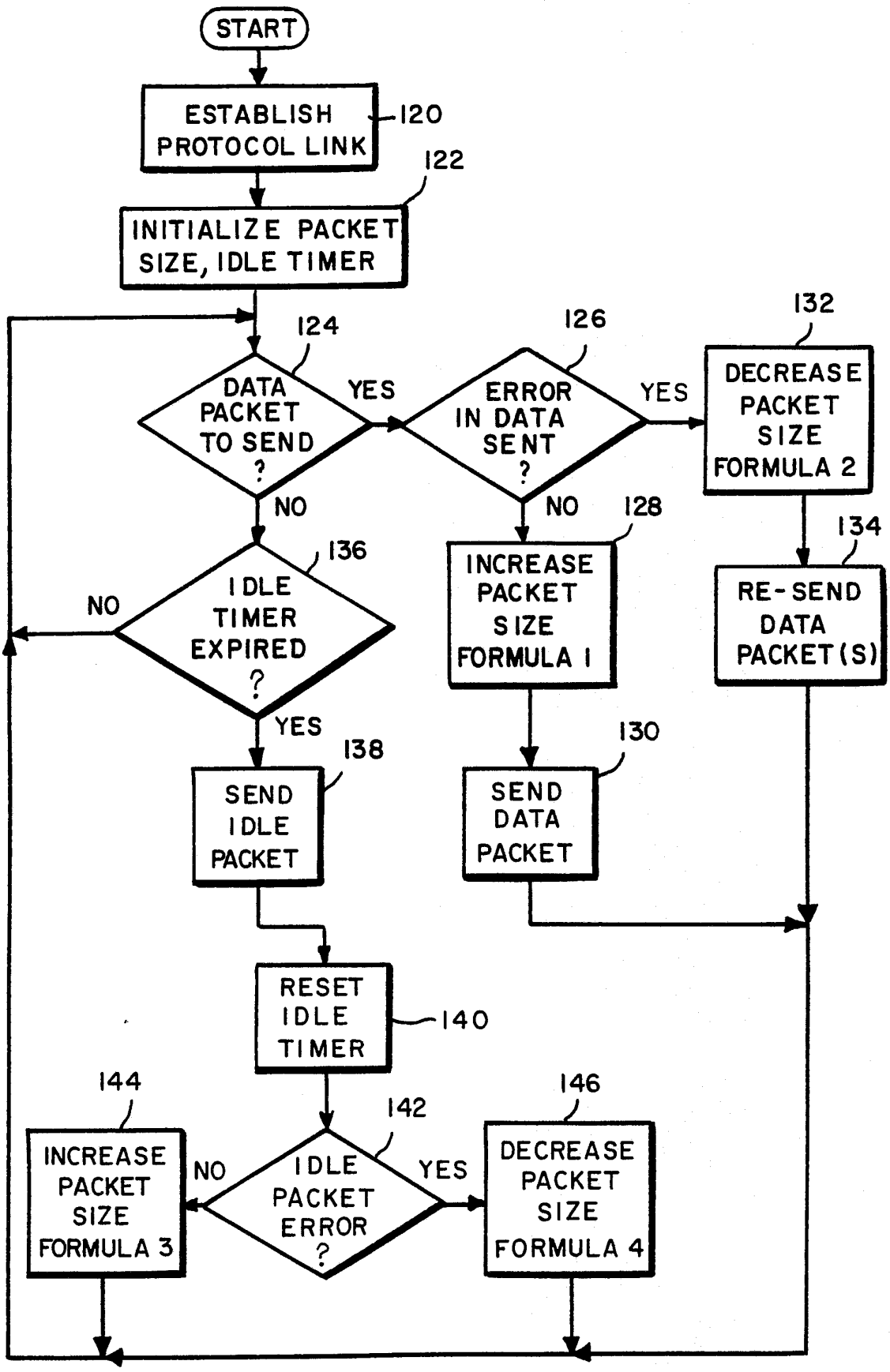
START
ESTABLISH PROTOCOL LINK
120
INITIALIZE PACKET SIZE, IDLE TIMER
122
DATA PACKET TO SEND ?
124
YES
NO
ERROR IN DATA SENT ?
126
YES
NO
DECREASE PACKET SIZE FORMULA 2
132
INCREASE PACKET SIZE FORMULA 1
128
RE-SEND DATA PACKET(S)
134
IDLE TIMER EXPIRED ?
136
NO
YES
SEND IDLE PACKET
138
SEND DATA PACKET
130
RESET IDLE TIMER
140
IDLE PACKET ERROR ?
142
NO
YES
INCREASE PACKET SIZE FORMULA 3
144
DECREASE PACKET SIZE FORMULA 4
146

FIG.3

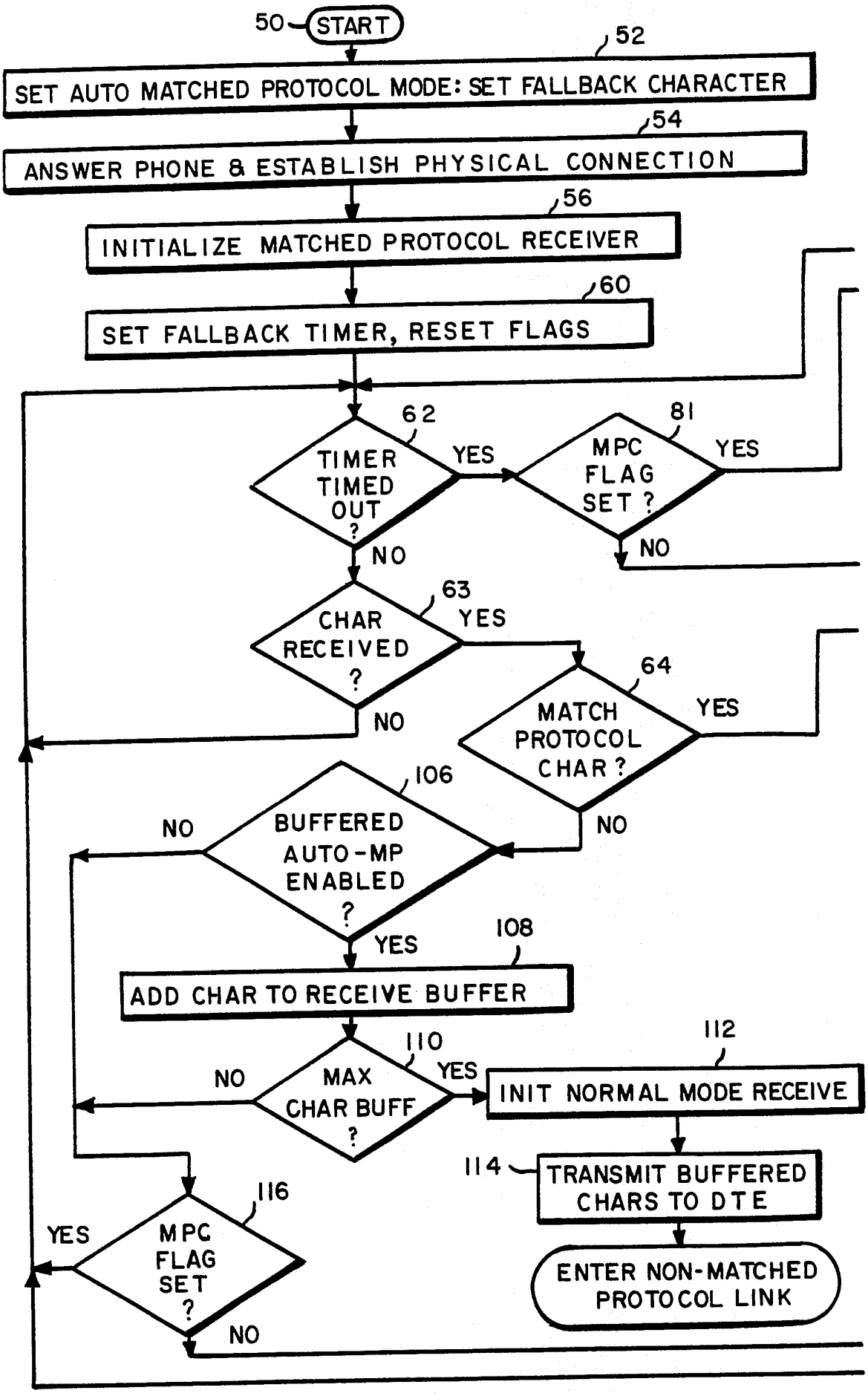
50
START
52
SET AUTO MATCHED PROTOCOL MODE: SET FALLBACK CHARACTER
54
ANSWER PHONE & ESTABLISH PHYSICAL CONNECTION
56
INITIALIZE MATCHED PROTOCOL RECEIVER
60
SET FALLBACK TIMER, RESET FLAGS
62
TIMER TIMED OUT ?
YES
81
MPC FLAG SET ?
YES
NO
NO
63
CHAR RECEIVED ?
YES
64
MATCH PROTOCOL CHAR ?
YES
NO
NO
106
BUFFERED AUTO-MP ENABLED ?
NO
YES
108
ADD CHAR TO RECEIVE BUFFER
110
MAX CHAR BUFF ?
NO
YES
112
INIT NORMAL MODE RECEIVE
114
TRANSMIT BUFFERED CHARS TO DTE
ENTER NON-MATCHED PROTOCOL LINK
116
MPC FLAG SET ?
YES
NO

FIG. 4

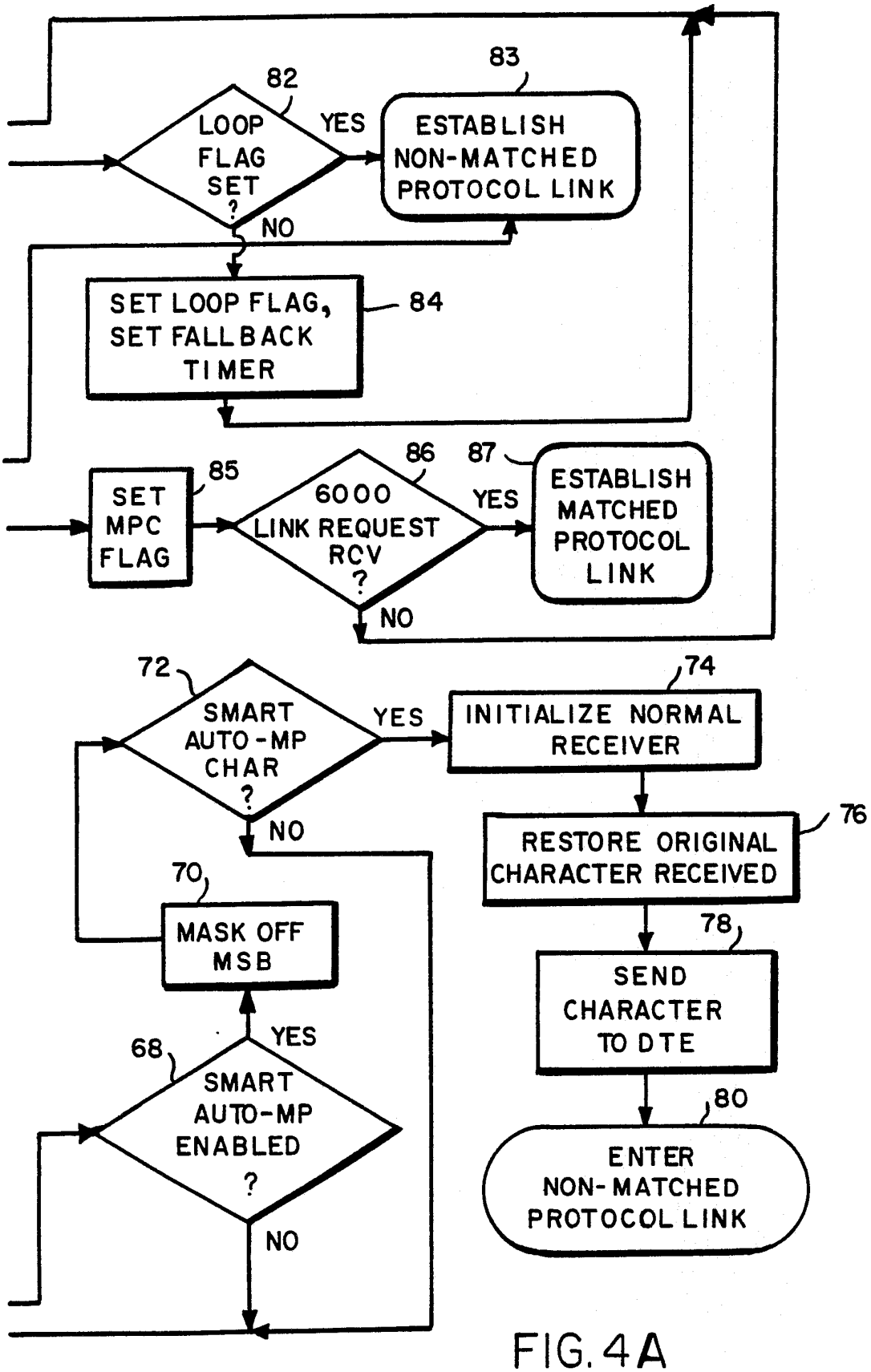
82
LOOP FLAG SET ?
YES
83
ESTABLISH NON-MATCHED PROTOCOL LINK
NO
SET LOOP FLAG, SET FALLBACK TIMER
84
SET MPC FLAG
85
6000 LINK REQUEST RCV ?
86
YES
87
ESTABLISH MATCHED PROTOCOL LINK
NO
72
SMART AUTO-MP CHAR ?
YES
74
INITIALIZE NORMAL RECEIVER
NO
RESTORE ORIGINAL CHARACTER RECEIVED
76
70
MASK OFF MSB
78
SEND CHARACTER TO DTE
YES
68
SMART AUTO-MP ENABLED ?
80
ENTER NON-MATCHED PROTOCOL LINK
NO

FIG. 4A