PCT

WORLD INTELLECTUAL PROPERTY ORGANIZATION
International Bureau

INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(51) International Patent Classification 6 :		(11) International Publication Number:	**WO 97/30388**
G06F 9/00	**A1**	(43) International Publication Date:	21 August 1997 (21.08.97)

(21) International Application Number: PCT/US97/01375

(22) International Filing Date: 31 January 1997 (31.01.97)

(30) Priority Data:
08/601,313 16 February 1996 (16.02.96) US

(71) Applicant: MOTOROLA INC. [US/US]; 1303 East Algonquin Road, Schaumburg, IL 60196 (US).

(72) Inventors: CORRIGAN, Richard; Apartment A, 2227 Pen-nview Lane, Schaumburg, IL 60194 (US). MANNETTE, Michael, Russell; 314 Stratford Place #302, Bloomingdale, IL 60108 (US). CHAN, Hang; 1061 Waterford Road, Bartlett, IL 60103 (US). GROVER, Naveen; 9139 Holly-berry Lane, Des Plaines, IL 60106 (US).

(74) Agents: POWELL, Jordan, C. et al.; Motorola Inc., Intellectual Property Dept., 1303 East Algonquin Road, Schaumburg, IL 60196 (US).

(81) Designated States: AL, AM, AT, AU, AZ, BB, BG, BR, BY, CA, CH, CN, CU, CZ, DE, DK, EE, ES, FI, GB, GE, HU, IL, IS, JP, KE, KG, KP, KR, KZ, LK, LR, LS, LT, LU, LV, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, TJ, TM, TR, TT, UA, UG, UZ, VN, ARIPO patent (KE, LS, MW, SD, SZ, UG), European patent (AT, BE, CH, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, ML, MR, NE, SN, TD, TG).

Published
With international search report.

(54) Title: METHOD FOR UPDATING A SOFTWARE CODE IN A COMMUNICATION SYSTEM
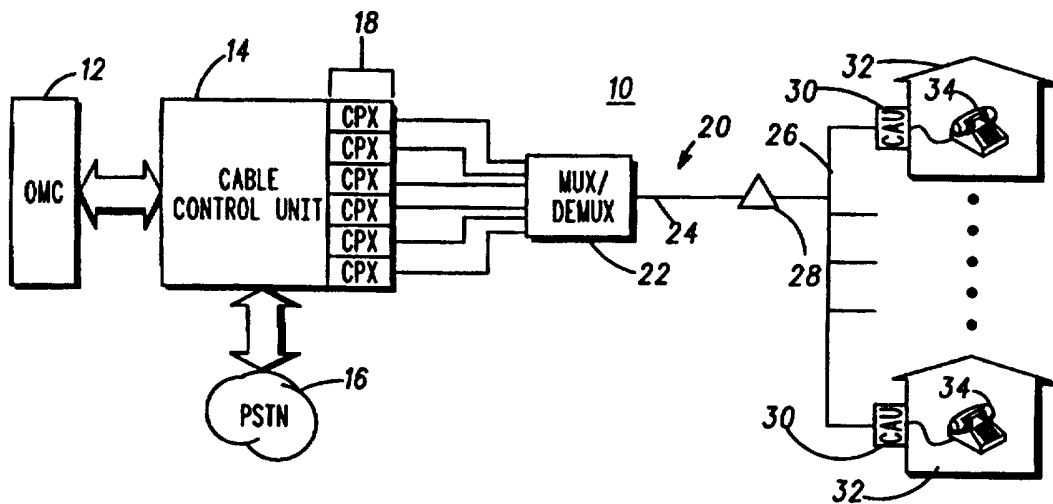
(57) Abstract

A method of updating a software code in a cable communication system (10) involves the steps of receiving a notification (100) by a cable access unit (30). Next, the cable access unit (30) receives (110) the software code. After receiving (110) the software code the CAU (30) resets (114) and reboots using the software code.

METHOD FOR UPDATING A SOFTWARE CODE IN A COMMUNICATION SYSTEM

5   Field of the Invention

The present invention relates generally to the field of communication systems and more particularly to a method of updating software code in a communication system.

10

Background of the Invention

A modern digital communication system has pair of transceivers that transmit messages over a transmission medium.

15   Each of the transceivers has an operating system that is typically stored in a memory. The operating system instructs the transceiver's how to react to overhead messages. For instances in a cable communication system, one of the transceivers involved in a communications session might detect that the communication

20   channel is noisy. Based on this information the transceiver would send a transfer to new communications channel message to the other transceiver. This transfer message is an overhead message and the other transceiver has to know how to respond to continue the communication session. The operating system might instruct

25   the other transceiver to tune to a predetermined alternate channel or it might instruct the other transceiver to tune to a system access channel that will designate an open channel for further communications.

Periodically the operating system of the transceivers may

30   have to be updated to correct errors or add new features. In a typical cable communications system this requires the system operator to send out a technician to every premise with a transceiver. The technician then installs the new operating system in the premise transceiver. This is an extremely expensive and

35   time consuming operation. As a result, the system operator is

2

reluctant to add new features or correct errors in the subscribers premise transceiver. This is also true of other communication systems, such as cellular, mobile radio, wireless data and wide area networks.

Thus there exists a need for a method of updating a software code in a transceiver in a communication system, that is inexpensive and can be accomplished promptly.

Brief Description of the Drawings

FIG. 1 is a block diagram of a cable communication system;

FIG. 2 is a block diagram of a cable access unit;

FIG. 3 is a signal diagram of a software code update;

FIG. 4 is an alternative signal diagram of a software code update;

FIG. 5 is a flow diagram of a reset and reboot of a cable access unit using a new software code;

FIGs. 6-10 are each a chart listing the features of a message used in the signal diagrams of FIGs. 2 and 3;

FIG. 11 is schematic diagram of a system broadcast channel;

FIG. 12 is a schematic diagram of the frequency allocation used in the cable communications system; and

FIG. 13 is a schematic diagram of a time division multiplexing scheme used in the cable communication system.

Detailed Description of the Drawings

In summary the invention provides a method for updating software code in a communication system. The method includes a protocol for downloading the software code and rebooting using the software code. The invention is described in connection with a cable communications system, but is in no way limited to this application. The invention applies broadly to any communication system.

3

FIG. 1 is a block diagram of a cable communication system
10. An operations and management center (OMC) 12 controls and
coordinates the overhead functions of the cable communication
system 10. The OMC 12 is coupled to a cable control unit (CCU) 14
5   that acts as the interface between the cable communication system
10 and a public switched telephone network (PSTN) 16. The CCU
14 contains a number of cable port transceivers (CPX) 18. Each
CPX 18 is capable of carrying on a communication session on one of
the communication channels available on a fiber-coax hybrid cable
10   plant 20. A multiplexer-demultiplexer (MUX/DEMUX) 22 combines
the transmissions from the plurality of CPXs 18 onto the cable
plant 20. The MUX/DEMUX 22 also breaks out and distributes the
signals coming into the CPXs 18. The cable plant 20 is composed of
a fiber optic cable 24, a coaxial cable 26 and an optical to electrical
15   converter 28. The coaxial cable 26 fans out and is connect to a
plurality of cable access devices (CAU) 30. The CAUs 30 are
attached to the outside of a subscriber's premises 32. One of the
CAU's 30 functions is to convert the radio frequency digital signals
in the coaxial cable 26 to the analog signals used in a plain old
20   telephone service (POTS) telephone 34.

FIG. 2 is a block diagram of the cable access unit (CAU,
subscriber unit) 30. The CAU 30 has a transceiver 40 coupled to
the coaxial cable 26. The transceiver 40 sends and receives signals
from the cable 26. The transceiver 40 splits the signal from the
25   cable 26 into a POTS signal 42 and a signal 44 intended for other
services. These other services include an ISDN (integrated digital
services network) basic rate interface (BRI) line, an ethernet line
and an RS232 line. The transceiver 40 is coupled to a processor 46
that executes a software code (operating code, operating software)
30   that controls the overhead functions of carrying on a
communications session with one of the CPXs 18. The software
code is stored in a memory 48. The memory 48 is divided into a
first memory portion (active code bank, code bank, first portion of
memory) and a second memory portion (designated code bank). A
35   non-volatile memory 50 is coupled to the processor 46 and

4

contains a pointer to the active code bank, a code version (version) and other information. The processor 46 is also coupled to a subscriber loop interface (SLI) 52. The SLI 52 converts the digital data into an analog signal 42 used in POTS.

5   In an alternative embodiment, the memory is divided into a plurality of code banks. The plurality of code banks are divided into active code bank and a back up code bank for each of the services: POTS, RS232, ethernet and BRI line. In another embodiment, the plurality of code banks are divided into an active

10   operating system, containing an active driver for all the services, and a back up driver for each of the services in its own code bank.

FIG. 3 is a signal diagram of a software code update. The OMC 12 has the job of ensuring that all the CAUs 30 have the latest version of the operating software (new operating software). When

15   the OMC 12 determines that the CAUs' 30 software code needs to be updated, it downloads the software code to the CCU 14. In one embodiment the CCU 14 stores the software code for later downloading (caching) In another embodiment the CCU 14 immediately proceeds to download the software code to the CAUs

20   30. The CCU 14 sends out an alert (notification) 100 to the appropriate CAU 30 or group of CAUs 30. Every CAU 30 has an alert ID for the CAU box and one for each telephony line. When the CAU 30 receives an alert ID in the alert message that is associated with the CAU box it knows that the communication

25   session that follows is concerned with the CAU's 30 internal operation and not a telephone call. The CAU 30 and the CCU 14 then perform a system access procedure 102. The CAU 30 then sends an alert acknowledgment (acknowledgment) 104. This response informs the CCU 14 that the CAU 30 received the alert

30   message 100 and is ready to proceed. The CCU 14 responds with a write request (cable file transfer protocol {CFTP} write request) 106. This informs the CAU 30 that the CCU 14 is going to transfer a file. The CAU 30 responds with a CFTP acknowledgment 108. The CCU 14 then starts transferring data blocks or packets (CFTP

35   data) 110. When each data packet is received error free the CAU

30 transmits a CFTP acknowledgment 112. This is repeated until the all the software code has been transferred to the CAU or CAUs 30. The CCU 14 will then send either a release 114 or a reset message 116. If the CCU 14 sends a release it will have to send a reset message 116 later to the CAU 30 and this is shown in FIG. 4. The CCU 14 then reports to the OMC 12 that the software code has been downloaded.

FIG. 4 shows a delayed reset and reboot of the CAU 30 using the new software code. The OMC 12 directs 120 the CCU 14 to have the CAU 30 reset and reboot using the new software code. The CCU 14 first transmits an alert 100 containing the alert ID associated with the CAU box. The CCU 14 and CAU 30 then perform a system access procedure 102. The CAU 30 responds with the alert acknowledgment 104. The CCU 14 sends the reset message 116 to the CAU 30. In an alternative embodiment the CAU 30 responds with a reset acknowledgment. The CCU 14 reports to the OMC 12 that the CAU 30 received the reset message. The CCU 14 and CAU 30 perform a layer 2 link release 122. The CAU 30 next performs a reboot and reinitializes service 124. When the CAU 30 reinitializes service it informs the CCU 14 of the version of the software code it is operating under.

The method of updating software code described herein is designed to avoid several potential pitfalls. The CAU 30 always has two different versions of the software operating code. One version is stored in high memory (first memory portion) and the other is stored in low memory (second memory portion). When the CCU 14 sends a write request 106 it includes information (designated memory bank) on whether to store the software code in the first memory portion or the second memory portion. A pointer in the non-volatile memory 50 indicates whether the first or the second memory portion is the active operating software code (active operating software). The CAU 30 will not over write the active software code. Any attempts by the CCU 14 to over write 106 the active software code with the new software code

6

(new code version) results in an error message. This prevents the
active software code (active code version) from being overwritten.

FIG. 5 is a flow diagram of the method of updating the
software code. This method avoids the situation of the CAU 30
5   getting locked in a loop of repeatedly attempting to reboot from a
damage software code. The process starts at step 150 with the
CAU 30 receiving a reset command at step 152. The CAU 30 then
updates its non-volatile memory with a version and a codebank of
the new software code at step 154. At step 156 the CAU reboots
10   using the new software code. When the reboot is unsuccessful at
step 158, the CAU uses the other codebank (second code bank),
containing the old software code (second software code), and
updates the non-volatile memory with the second version, at step
160. The CAU reboots from the old software code at step 162.
15   Since the CAU has been running under this old software code the
CAU will reboot without any problem. The CAU reinitializes
service and transmits the version of the software it is operating
under at step 166. When the reboot at step 158 is successful the
process skips to step 166. At step 168 the CCU compares the
20   active version of the software code being used by the CAU with the
version of the new software code it transmitted to the CAU. When
the comparison at step 168 shows the two versions are not the
same, an error message (error condition) is sent to the OMC, at step
170. The process then ends at step 172. When the comparison at
25   step 168 shows the two versions of software code are the same,
the process ends at step 172.

The process described in FIG. 5 ensures that if the new
software code downloaded to the CAU 30 has a problem it does not
disable the CAU 30. The CAU 30 always has a tested version of
30   operating software code. If there is a problem in the new
software, the CAU 30 uses the tested version of software code
(second software code).

FIG. 6 is a table of the information provided in the CFTP
write request message 106. The message type 200 identifies the
35   message as a write request and requires 1 byte of data. The

transfer mode 202 tells the CAU if the file type is binary, enciphered, compressed or in some other format. The version 204 contains a variety of information including an identifier, the length of the this version of software code and the designated code bank that the software code is to be stored in the CAU 30.

FIG. 7 is a table of the information provided in the CFTP data block message 110. The message type 200 identifies the message as a data block and requires 1 byte of data. The block number 206 identifies which block of data is being transferred. The data block 208 is the raw data being transferred and can be up to 252 bytes of data per data block message.

FIG. 8 is a table of the information provided in the CFTP acknowledgment message 108. The message type 200 identifies the message as an acknowledgment and requires 1 byte of data. The block number 206 identifies which block of data was last received.

FIG. 9 is a table of the information provided in the CFTP error message. The message type 200 identifies the message as an error message and requires 1 byte of data. The cause 210 explains why the error occurred.

FIG. 10 is a table of the information provided in the reset message 116. The message type 200 identifies the message as a reset message and requires 1 byte of data. The version (software version code, new software version code) 204 tells the CAU which version of software code to use when rebooting.

The invention is designed so that the software code can be downloaded while a telephone connection exists on the same CAU. This is made possible by two things. The first is that the alert IDs for the CAU boxes is different than the alert IDs for the telephone lines. The alert IDs are transmitted over a system broadcast channel. The structure of the system broadcast channel 220 is shown in FIG. 11. The system broadcast channel 220 includes in its message payload 222, an alert channel 224 and a system information channel (SIC) 226. The alert channel 224 carries the alert IDs (plurality of alert identifiers). The CCU can send an alert

8

for either the CAU box or a telephony line associated with the CAU. When an alert is sent for either the associated telephone line or the CAU box, the CAU performs a system access and is instructed to tune to a traffic channel.

The traffic channel has a different carrier frequency than the system broadcast channel. FIG. 12 shows the frequency allocation used by the cable communication system. The upstream traffic (CAU to CCU) is confined between 5-42 MHz. The downstream traffic (CCU to CAU) is confined between 50-750 MHz. A 6 MHz block 240 of spectrum is divided into ten 600 KHz wide carriers 242. Each 600 KHz wide carrier 242 is divided into 8 time slots 250, see FIG. 13. Each time slot 250 is a traffic channel 250. A traffic channel 250 includes a synchronization channel 252, a control channel 254, a slow channel 256, a fast channel 258 and an error channel 260.

If a CAU is involved in a telephony connection (telephony communication) when the CCU is attempting to download the software code it sends the CAU box alert ID over the slow channel 256. The software code is then downloaded over one of the empty time slots 250 having the same carrier frequency as the traffic channel being used for the telephone connection. If a CAU is involved in a download when an incoming call is being placed to the associated telephone line, the CAU receives an alert over the slow channel 256 of the traffic channel the CCU is using to download the software. If the CAU attempts to place a call while a download of software code is occurring, the CAU will request access to a traffic channel over a system access channel. In all cases the CAU will have to use traffic channels having the same carrier frequency for the download and the telephone connection.

Thus there has been described a method for updating a software code in a communication system, that is inexpensive and is performed promptly. The method reduces the need for technicians to individually update every subscriber unit. In addition, the method does not interfere with on-going communication sessions and provides a fail safe mechanism in case

of error in the new software code. While the invention has been described in conjunction with specific embodiments thereof, it is evident that many alterations, modifications, and alterations will be apparent to those skilled in the art in light of the foregoing

5   description. Accordingly, it is intended to embrace all such alterations, modifications, and variations in the appended claims.

1 0

Claims

1.   A method of updating a software code in a cable communication system, comprising the steps of:

5   (a) receiving a notification;

(b) receiving the software code; and

(c) resetting and rebooting using the software code.

2.   The method of claim 1, wherein step (c) further

10   includes the step of:

(c1) receiving a reset command.

3.   The method of claim 1, wherein step (a) further includes the steps of:

15   (a1) transmitting an acknowledgment; and

(a2) receiving a write request.

4.   The method of claim 3, wherein step (a2) further includes the steps of:

20   (i) determining a designated code bank to hold the software code;

(ii) checking whether the designated code bank is equal to an active code bank; and

(iii) when the designated code bank is equal to the active

25   code bank, rejecting the software code.

5.   The method of claim 4, further including the steps of:

(iv) when the designated code bank is not equal to the

30   active code bank, storing the software code in the designated code bank.

11

6.   A method of updating a software code in a cable communication system, comprising the steps of:

(a) receiving the software code from an operations and management center;

(b) transmitting an alert to a cable access unit;

(c) transmitting the software code; and

(d) transmitting a reset command.

7.   A method of updating a software code in a cable communication system, comprising the steps of:

(a) updating a memory with a version and a code bank; and

(b) rebooting a cable access unit with the software code.

8.   The method of claim 7, further including the steps of:

(c) when the step of rebooting is unsuccessful, switching to a second software code;

(d) updating the memory with a second version and a second code bank; and

(f) rebooting the cable access unit with the second software code.

9.   A method of updating software in a subscriber unit of a communication system, comprising the steps of:

(a) providing the subscriber unit comprising a first memory portion and a second memory portion;

(b) providing an active code version in the first memory portion;

(c) loading a new code version into the second memory portion;

(d) restarting the subscriber unit using the new code version; and

(e) in response to the step of restarting failing, restarting the subscriber unit using the active code version.

12

10.   The method of claim 9, further including the step
of:

(f) in response to the step of restarting the subscriber
unit using the new code version being successful, designating
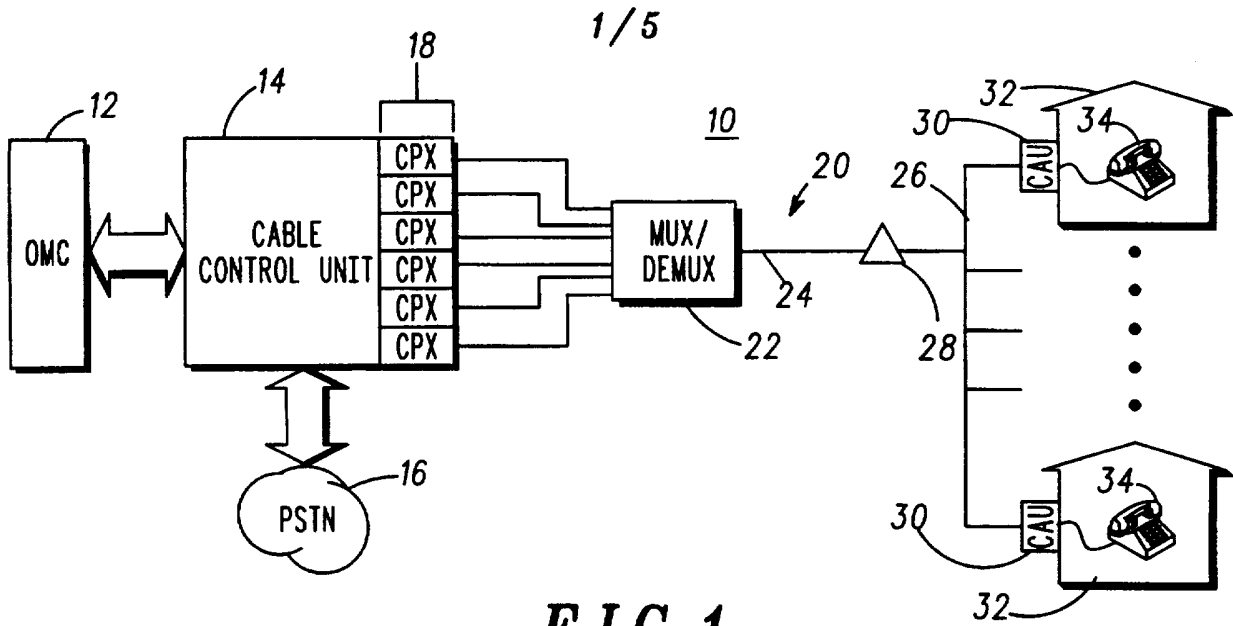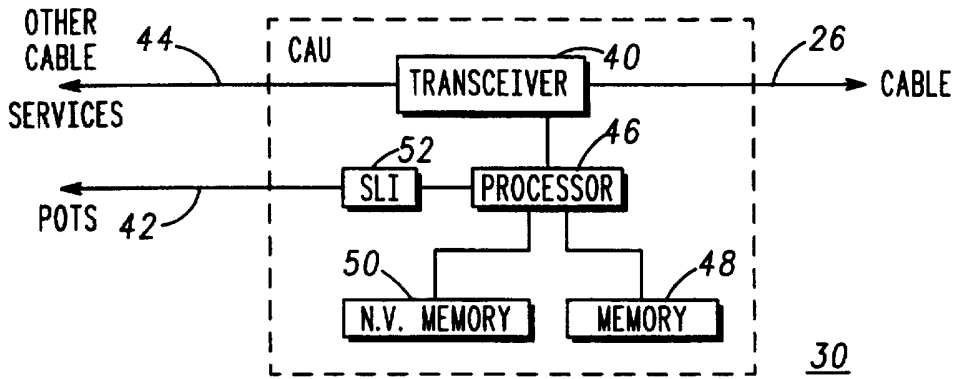the new code version as the active code version.

1/5

FIG.1

FIG.2

_220_

BITS								FC
8	7	6	5	4	3	2	1	OCTET
RESERVED				s_2	s_1	CHANNEL TYPE		1
MESSAGE PAYLOAD 222 [AC, SIC] 226 224								2
								(CONT.)
								18
CHECKSUM(IF s_2= 1), OR MESSAGE PAYLOAD (IF s_2= 0)								19
								20

FIG.11

2/5
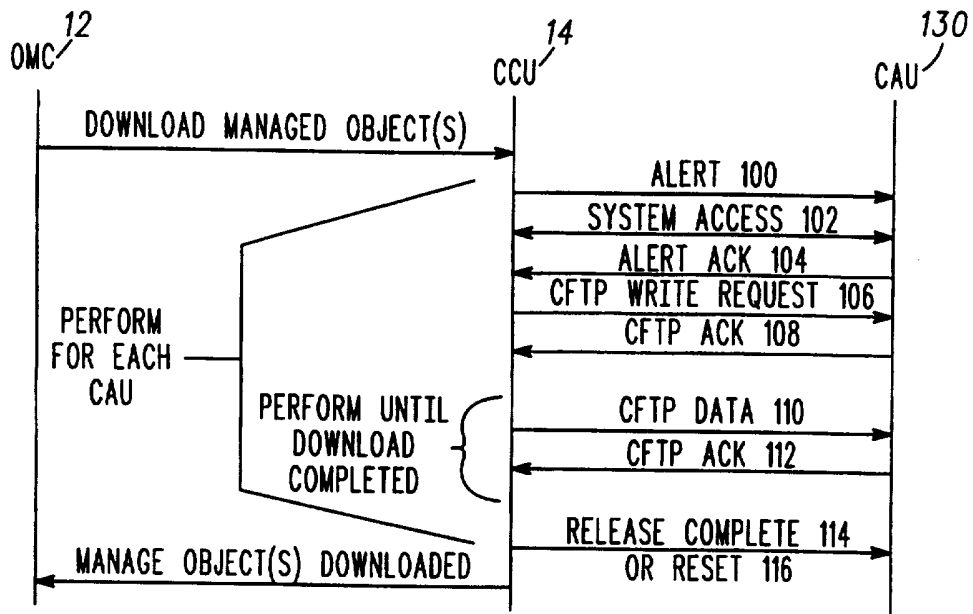
FIG.3
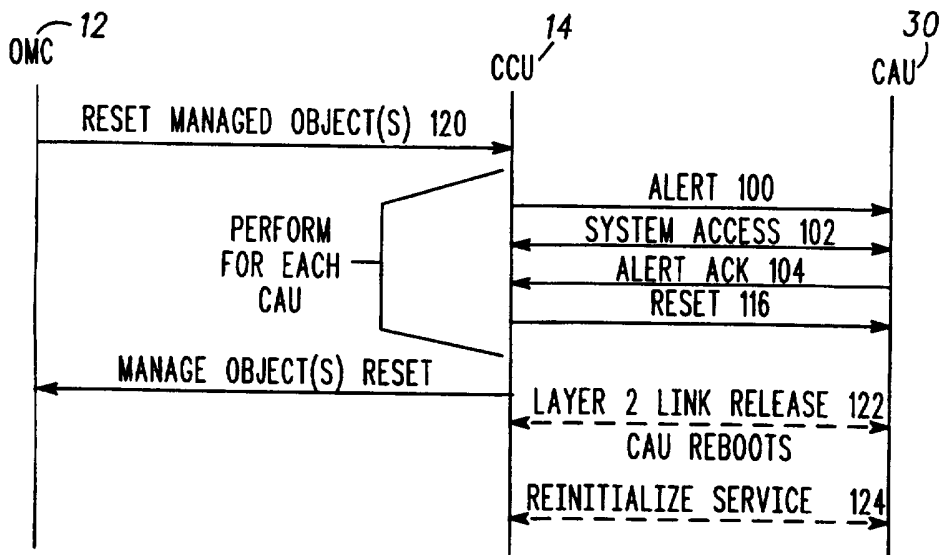
FIG.4
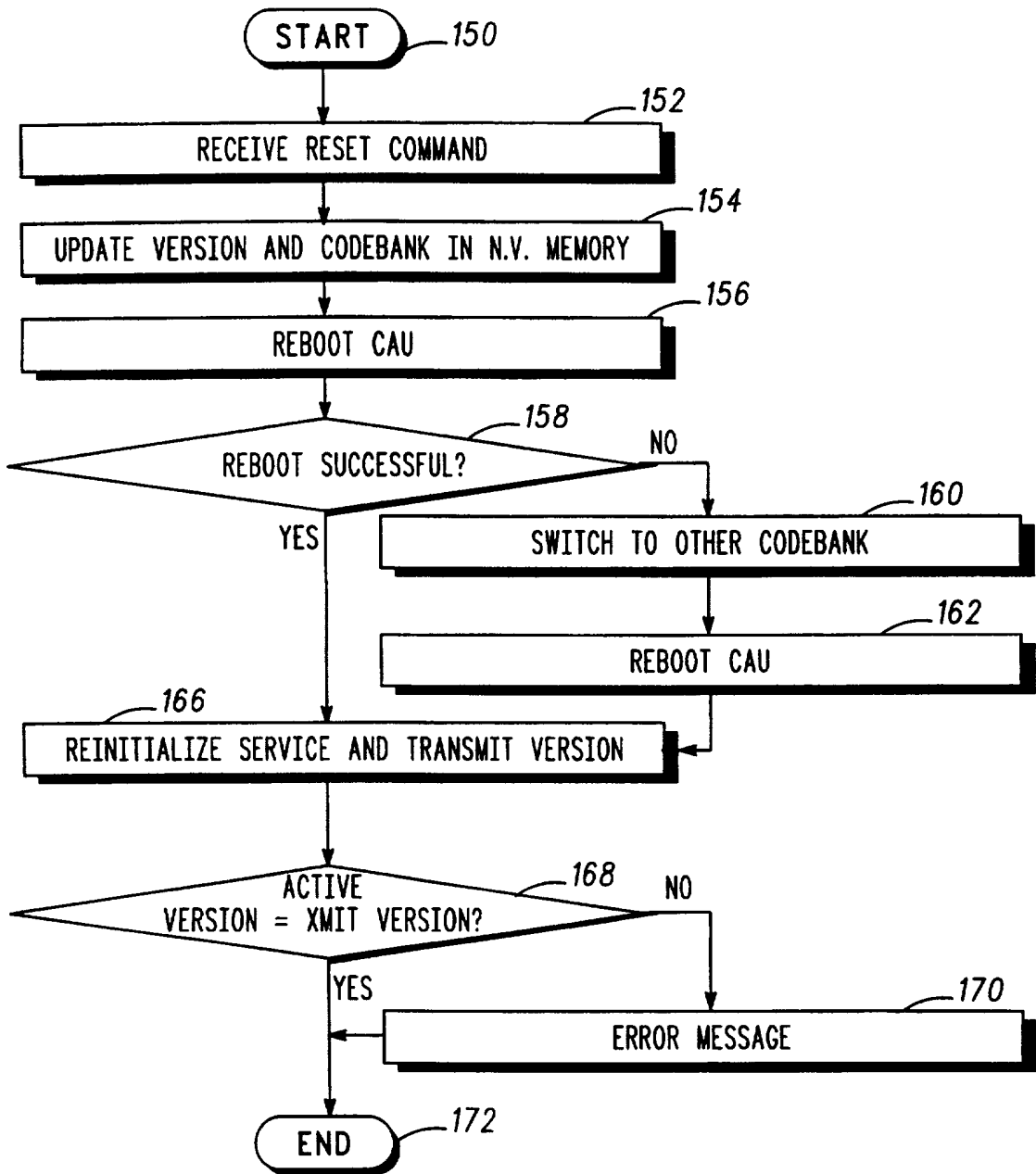
FIG.5

4/5

CFTP WRITE REQUESST MESSAGE

INFORMATION ELEMENTS	LENGTH	DESCRIPTION
MESSAGE TYPE	1	LAYER 3 MESSAGE IDENTIFIER
TRANSFER MODE	3	SPECIFIES FILE TYPE
VERSION	14	SOFTWARE VERSION NUMBER

200 — MESSAGE TYPE
202 — TRANSFER MODE
204 — VERSION

FIG.6

CFTP DATA BLOCK MESSAGE

INFORMATION ELEMENTS	LENGTH	DESCRIPTION
MESSAGE TYPE	1	LAYER 3 MESSAGE IDENTIFIER
BLOCK NUMBER	4	SEQUENTIAL NUMBER IDENTIFYING DATA BLOCK
DATA BLOCK	2 TO 252	FIXED BLOCKS OF 252 BYTES WITH THE LAST DATA BLOCK BEING A LENGTH FROM 0 TO 251

200 — MESSAGE TYPE
206 — BLOCK NUMBER
208 — DATA BLOCK

FIG.7

CFTP ACK MESSAGE

INFORMATION ELEMENTS	LENGTH	DESCRIPTION
MESSAGE TYPE	1	LAYER 3 MESSAGE IDENTIFIER
BLOCK NUMBER	4	SEQUENTIAL NUMBER IDENTIFYING DATA BLOCK

200 — MESSAGE TYPE
206 — BLOCK NUMBER

FIG.8

CFTP ERROR MESSAGE

INFORMATION ELEMENTS	LENGTH	DESCRIPTION
MESSAGE TYPE	1	LAYER 3 MESSAGE IDENTIFIER
CAUSE	5	REASON FOR REJECTING DATA BLOCK

200 — MESSAGE TYPE
210 — CAUSE

FIG.9

RESET MESSAGE

INFORMATION ELEMENTS	LENGTH	DESCRIPTION
MESSAGE TYPE	1	LAYER 3 MESSAGE IDENTIFIER
VERSION	14	SOFTWARE VERSION NUMBER

200 — MESSAGE TYPE
204 — VERSION

FIG.10
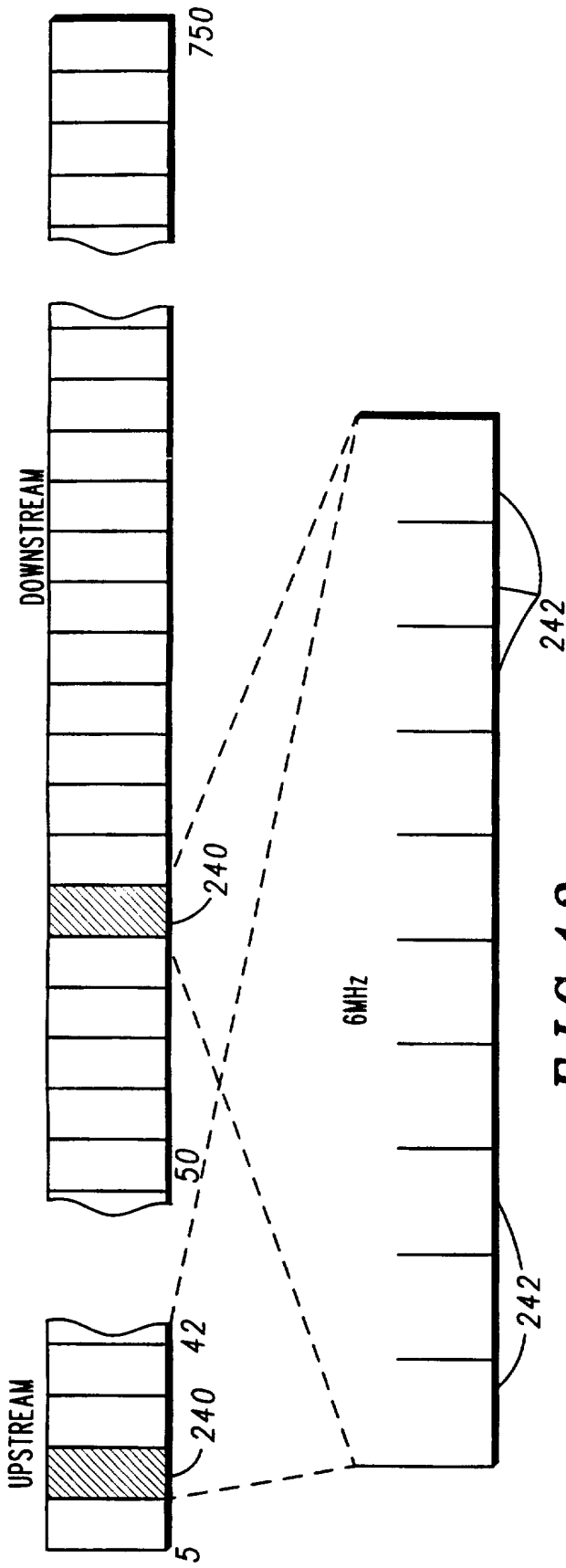
FIG.12
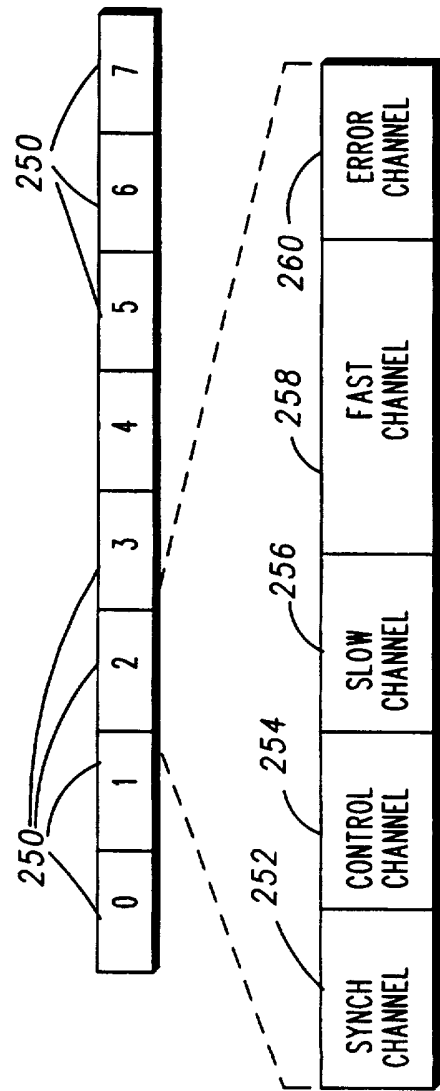
FIG.13

INTERNATIONAL SEARCH REPORT

A. CLASSIFICATION OF SUBJECT MATTER

IPC(6) :G06F 9/00

US CL :395/712, 653

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

U.S. : 395/712, 653, 651, 652

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

APS

search terms: reboot, update, reset, telephony, cable

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 5,003,591 A (KAUFFMAN et al.) 26 March 1991, col. 5, line 6 - col. 13, line 50.	1-10
Y	US 5,388,211 A (HORNBUCKLE) 07 February 1995, col. 4, line 45 - col. 14, line 14.	1-10
Y, P	US 5,568,641 A (NELSON et al.) 22 October 1996, col. 3, line 6 - col. 6, line 26.	1-10
Y	US 5,471,674 A (STEWART et al.) 28 November 1995, col. 2, line 60 - col. 12, line 29.	1-10

☐ Further documents are listed in the continuation of Box C. ☐ See patent family annex.

*	Special categories of cited documents:	"T"	later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"A"	document defining the general state of the art which is not considered to be of particular relevance		
"E"	earlier document published on or after the international filing date	"X"	document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"L"	document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)		
		"Y"	document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"O"	document referring to an oral disclosure, use, exhibition or other means		
"P"	document published prior to the international filing date but later than the priority date claimed	"&"	document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
20 MARCH 1997	30 APR 1997

Name and mailing address of the ISA/US Commissioner of Patents and Trademarks Box PCT Washington, D.C. 20231	Authorized officer KEVIN A. KRIESS
Facsimile No. (703) 305-3230	Telephone No. (703)305-9668

Form PCT/ISA/210 (second sheet)(July 1992)*