OPIC
Office de la propriété intellectuelle du Canada

CIPO
Canadian Intellectual Property Office

(12) (19) (CA) **Demande-Application**

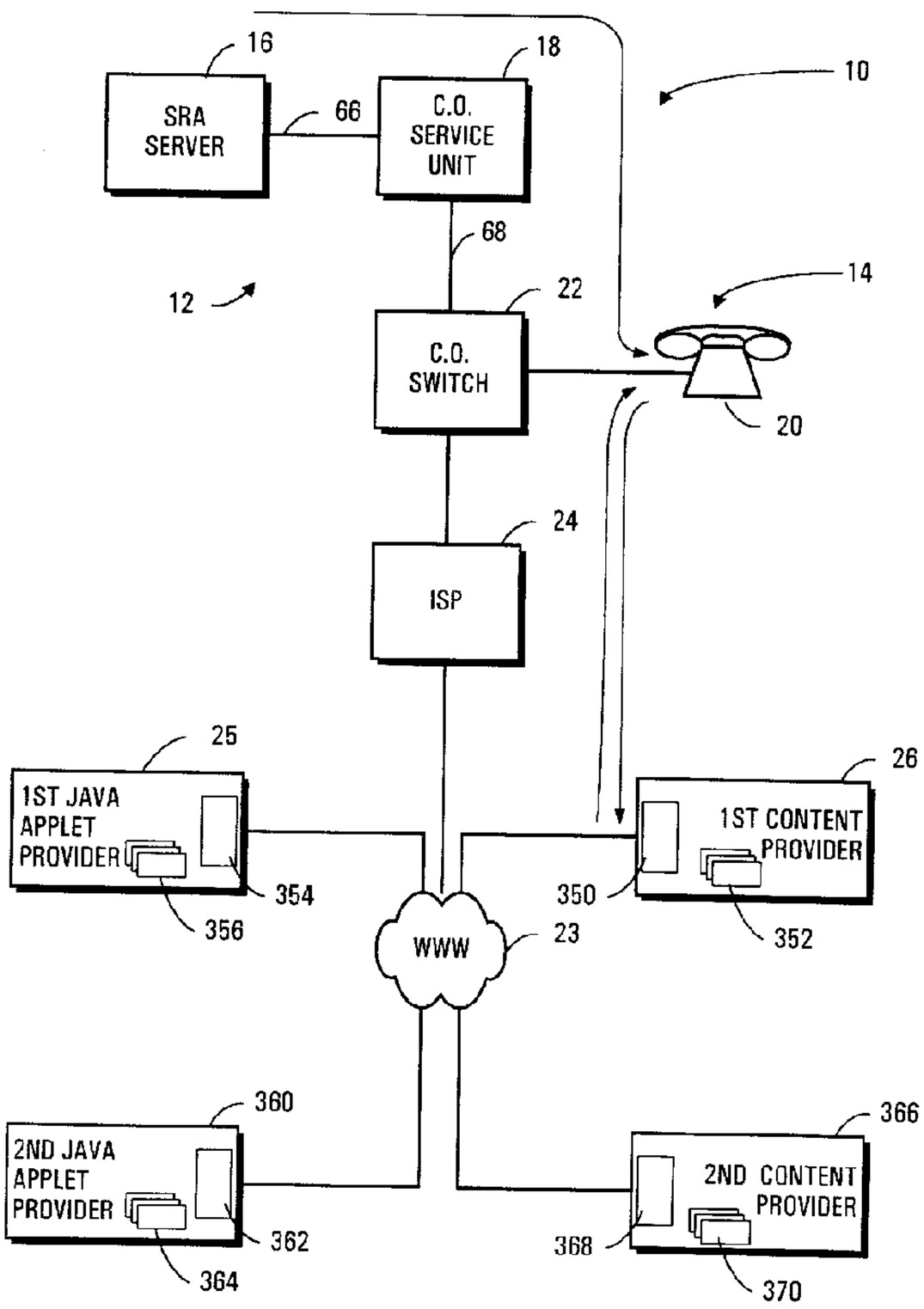
(21) (A1) **2,242,404**
(22) 1998/07/06
(43) 2000/01/06

(72) CRAMER, Seth Whitney, CA
(72) MICHAUD, Bertrand Marc-Andre, CA
(72) ROHS, Charles Gerald, CA
(72) PASLAWSKI, Daniel John, CA
(72) LO, Kevin Chi Hin, CA
(72) LI, Qi, CA
(71) NORTHERN TELECOM LIMITED, CA

(51) Int.Cl.[6] H04L 12/12, H04M 1/26, H04M 3/22, G06F 13/14, G11B 23/00

(54) **METHODE ET APPAREIL DE CONFIGURATION D'UN LOGICIEL TELEPHONIQUE A DISTANCE**

(54) **REMOTE TELEPHONE SOFTWARE CONFIGURATION APPARATUS AND METHOD**



(57) A file download method and apparatus for downloading files from a network into memory. The apparatus includes a processor operable to communicate on the network. The processor runs a browser for downloading from a first

OPIC
Office de la propriété intellectuelle du Canada

CIPO
Canadian Intellectual Property Office

(21) (A1) **2,242,404**
(22) 1998/07/06
(43) 2000/01/06

resource identified by a first network resource locator, an applet including a second network resource locator, an applet viewer for running the applet, a public interface for communicating with the applet to receive the second network resource locator from the applet, a security manager for preventing running applets from accessing the memory and a handler having access to the memory, the handler being in communication with the public interface, for communicating with a second network resource identified by the second network resource locator to receive and store in the memory, a file from the second network resource.

REMOTE TELEPHONE SOFTWARE CONFIGURATION APPARATUS AND METHOD

ABSTRACT

A file download method and apparatus for downloading files from a network into memory. The apparatus includes a processor operable to communicate on the network. The processor runs a browser for downloading from a first resource identified by a first network resource locator, an applet including a second network resource locator, an applet viewer for running the applet, a public interface for communicating with the applet to receive the second network resource locator from the applet, a security manager for preventing running applets from accessing the memory and a handler having access to the memory, the handler being in communication with the public interface, for communicating with a second network resource identified by the second network resource locator to receive and store in the memory, a file from the second network resource.

REMOTE TELEPHONE SOFTWARE CONFIGURATION APPARATUS AND METHOD

FIELD OF THE INVENTION

This invention relates to remote configuration of a telephone and more particularly to downloading files from a network for execution on a telephone.

BACKGROUND OF THE INVENTION

The features and services available in subscriber telephone sets and in available telephone services continue to grow in richness and flexibility. Many microprocessor-based telephones are capable of supporting many different mixtures of features and capabilities, by changing the configuration of processor-readable parameters.

At the same time, the costs and delays associated with visits to the subscriber premises by service provider personnel are becoming less acceptable to the subscriber.

It would be desirable, therefore, for telephone service providers to be able to change the configuration of a subscriber telephone set without visiting the subscriber premises.

Some manufacturers' telephones have been modified to support remote re-configuration by using a variant of the "Caller ID" signal, whereby the telephone service provider can, presumably in response to a request by the subscriber, add, modify or delete services or other parameters defined within the telephone's processor subsystem. In simple situations, this re-configuration can be accomplished by this method in a few seconds. The quantity of data associated with a more complex telephone, or especially one capable of displaying promotional or informative images, is not efficiently transferred using this method, because of inherent limitations in the data transmission speeds

available. A faster method of transferring data to a telephone is required to support these newer, display capable telephones. Operating companies need a way to force a telephone to get information on a periodic basis. Existing suppressed Ring Access (SRA) techniques rely on FSK (frequency shift keying) data which is very slow and is not reliable.

Generally, software in telephones is difficult to field upgrade because it must be downloaded and verified before the old software is deleted. Traditionally, software was upgraded in a single unit requiring that the full software application be duplicated before it can be verified and accepted. This is very costly as it requires a relatively large memory area to hold both the old software and the new software.

The problem is compounded as software, data and service downloads require a large number of files to be downloaded successfully to the telephone. If one file is missed or downloaded improperly then the entire download could be compromised. Individual file downloads are possible but the files must be managed, which greatly complicates the system.

What would be desirable is a way of conveniently field upgrading telephone software on line, without requiring entire software replacement. The present invention addresses this need.

SUMMARY OF THE INVENTION

In accordance with one aspect of the invention, there is provided a file download method and apparatus for downloading files from a network into memory. The apparatus includes a processor operable to communicate on the network. The processor runs a browser for downloading from a first resource identified by a first network

resource locator, an applet including a second network resource locator, an applet viewer for running the applet, a public interface for communicating with the applet to receive the second network resource locator from the applet, a security manager for preventing running applets from accessing the memory and a handler having access to the memory, the handler being in communication with the public interface, for communicating with a second network resource identified by the second network resource locator to receive and store in the memory, a file from the second network resource.

Preferably, the handler is operable to decompress the file from the second network resource to produce a plurality of decompressed files.

Preferably, the handler is operable to verify the file has been received in accordance with pre-defined criteria.

Preferably, the handler is operable to extract a password from the file and compare the password against a pre-defined password and to store the file in the memory when the password matches the pre-defined password.

Preferably, the handler is operable to extract a filename from the file from the second resource and load the file into an area of memory associated with the filename.

Preferably, the apparatus includes an FSK receiver for receiving an FSK message having a first network resource locator identifying the first resource and preferably, the browser is operable to receive the first network resource locator from the FSK receiver.

Preferably, the apparatus includes a first telephone line interface for receiving the FSK message from a first

telephone line and for providing the FSK message to the FSK receiver.

Preferably, the apparatus includes an event parser for extracting the first uniform resource locator from the FSK message for presentation to the browser.

Preferably, the apparatus includes a second telephone line interface, the processor using the second telephone line interface for communicating on the network.

In accordance with another aspect of the invention, there is provided a computer readable medium on which is stored a plurality of computer readable codes representing instructions for directing a processor to download files from a network into memory in communication with the processor, the processor being operable to communicate on the network, the instructions directing the processor to:

a) implement a browser operable to download to the processor from a first resource identified by a first network resource locator, an applet including a second network resource locator;

b) run the applet on the processor;

c) receive the second network resource locator from the applet;

d) prevent applets running on the processor from accessing the memory; and

e) communicate with a second network resource identified by the second network resource locator to receive and store in the memory, a file from the second network resource.

Preferably, the computer readable medium includes a plurality of computer readable codes for directing the processor to decompress the file from the second network resource to produce a plurality of decompressed files.

Preferably, the computer readable medium includes a plurality of computer readable codes for directing the processor to verify the file has been received in accordance with pre-defined criteria.

Preferably, the computer readable medium includes a plurality of computer readable codes for directing the processor to extract a password from the file and compare the password against a pre-defined password and to store the file in the memory when the password matches the pre-defined password.

Preferably, the computer readable medium includes a plurality of computer readable codes for directing the processor to extract a filename from the file from the second resource and load the file into an area of memory associated with the filename.

Preferably, the computer readable medium includes a plurality of computer readable codes for directing the processor to receive an FSK message having a first network resource locator identifying the first resource and receive the first network resource locator from the FSK receiver.

Preferably, the computer readable medium includes a plurality of computer readable codes for directing the processor to receive the FSK message from a first telephone line and provide the FSK message to the FSK receiver.

Preferably, the computer readable medium includes a plurality of computer readable codes for directing the

processor to extract the first uniform resource locator from the FSK message for presentation to the browser.

Preferably, the computer readable medium includes a plurality of computer readable codes for directing the processor to use a second telephone line interface for communicating on the network.

In accordance with another aspect of the invention, there is provided a method for remotely configuring a telephone having a processor operable to communicate on a network and memory accessible by the processor for storing groups of processor readable codes for directing the processor to effect telephone functionality. The method includes the steps of:

a) downloading to the processor from a first network resource on the network, an applet including a second network resource locator;

b) running the applet on the processor;

c) to receiving the second network resource locator from the applet;

d) preventing applets running on the processor from accessing the memory; and

e) establishing communications between the processor and a second network resource identified by the second network resource locator to receive and storing in at least one of the groups, a file from the second network resource.

In accordance with another aspect of the invention, there is provided a remotely configurable telephone having a processor operable to communicate on a network and memory

accessible by the processor for storing groups of processor readable codes for directing the processor to effect telephone functionality, the processor being programmed to run:

a) a browser for downloading to the processor from a first network resource on the network, an applet including a second network resource locator;

b) an applet viewer for running the applet on the processor;

c) a public interface for communicating with the applet to receive the second network resource locator from the applet;

d) a security manager for preventing applets running on the processor from accessing the memory; and

e) a handler providing the processor access to the memory, the handler being in communication with the public interface, for communicating with a second network resource identified by the second network resource locator to receive and store in at least one of the groups, a file from the second network resource.

In accordance with another aspect of the invention, there is provided a method of remotely configuring a telephone having a processor operable to communicate on a network and memory accessible by the processor for storing groups of processor readable codes for directing the processor to effect telephone functionality, the method including the steps of:

a) downloading to the processor from a first network resource on the network, an applet including a second network resource locator;

b) running the applet on the processor;

c) receiving the second network resource locator from the applet;

d) preventing applets running on the processor from accessing the memory; and

e) establishing communications between the processor and a second network resource identified by the second network resource locator to receive and store in at least one of the groups, a file from the second network resource.

In accordance with another aspect of the invention, there is provided a computer readable medium on which is stored a plurality of computer readable codes representing instructions for directing a processor to remotely configure a telephone having a processor operable to communicate on a network and memory accessible by the processor for storing groups of processor readable codes for directing the processor to effect telephone functionality, the instructions directing the processor to:

a) download from a first network resource on the network, an applet including a second network resource locator;

b) run the applet;

c) communicate with the applet to receive the second network resource locator from the applet;

d) prevent running applets from accessing the memory; and

e) communicate with the public interface, to communicate with a second network resource identified by the second network resource locator to receive and store in at least one of the groups, a file from the second network resource.

Effectively an aspect of the invention involves the use of a short FSK blast which instructs the telephone to navigate to the web using a high speed modem. The modem connection is a reliable high speed connection which can be used to download large files of telephone functionality software or display-based advertising.

The use of Java software helps facilitate upgrading of telephone software in small components (packages) which means that only the downloaded component needs to be duplicated in memory. This is much more cost effective and much easier to implement compared to existing systems.

The large file to be downloaded is preferably a compressed file bundle which includes all the necessary replacement or new files to be sent to the telephone in one verifiable download. An applet is started on the telephone which makes a call to a main application running on the phone to download the bundle and install it in a location in the file system. The telephone then starts the download of the file bundle and dynamically decompresses the bundle into individual files. Once the full bundle has been downloaded it is verified and the installation is completed. The bundle is verified, through a checksum algorithm, that it was properly received and in this manner, all files are verified as having been received properly.

This invention provides a subscriber telephone set with the ability to efficiently receive and activate configuration and other files defined by a service provider, without intervention by the subscriber, and without the requirement for attendance at the subscriber premises by service provider personnel.

BRIEF DESCRIPTION OF THE DRAWINGS

In drawings which illustrate embodiments of the invention,

Figure **1** is a schematic diagram of an apparatus for downloading files from a network into memory according to a first embodiment of the invention;

Figure **2** is a block diagram of a suppressed ring access (SRA) server and a central office service unit according to the first embodiment of the invention;

Figure **3** is a schematic representation of an SRA control message and an SRC message transmitted by the SRA server and the central office service unit respectively;

Figure **4** is a schematic representation of a central office and a telephone according to the first embodiment of the invention;

Figure **5** is a block diagram of a central office line interface in the telephone of Figure **4**;

Figure **6** is a tree diagram of a directory structure of file directories within the telephone;

Figure **7** is a schematic representation of storage areas in the RAM shown in Figure **4**;

Figure **8** is a schematic representation of a program architecture in a processor in the telephone of Figure **4**;

Figure **9** is a flowchart of an event parser thread run by a system processor of the telephone of Figure **4**;

Figure **10** is a flowchart of a main event handler thread run by the system processor;

Figure **11** is a flowchart of an event dispatcher run by the system processor;

Figure **12** is a flowchart of the browser thread run by the system processor;

Figure **13** is a flowchart of a network driver run by the system processor;

Figure **14** is a flowchart of an applet thread run by the system processor;

Figure **15** is a flowchart of a public interface method by the system processor; and

Figure **16** is a flowchart of a handler thread run by the system processor.

DETAILED DESCRIPTION

Figure 1

Referring to Figure **1**, a system for downloading files from a network into memory in communication with a processor operable to communicate on the network is shown generally at **10**. In this embodiment a first network node is shown generally at **12** while a second network node is shown generally at **14**.

The first network node includes a modified ring access server which, in this embodiment is a suppressed ring access (SRA) server **16** and a central office service unit **18** and the second network node includes a telephone **20**. The suppressed ring access server **16** is in communication with the central office service unit **18** and the central office service unit is in communication with the telephone through a central office switch **22** in the public switched telephone network.

In addition to providing conventional telephone features, the telephone **20** receives FSK messages including a network resource indicator, through suppressed ring access or single ring access initiated by the SRA server **16** and central office service unit **18**. The telephone **20** also has transceiver hardware and software for establishing communications with an internet service provider **24**, through the central office switch **22** to establish communications with first or second Java applet providers **25** and **360** which act as network resources identified by network resource indicators included in the FSK message provided to the telephone by the central office service unit **18**. The SRA server thus directs the telephone **20** to a network resource from which it can download Hypertext Markup Language (HTML) pages. Such HTML pages may include applets which can be run at the telephone **20** to direct the telephone to communicate with first or second content providers **26** and **366** which can provide files to re-configure the telephone to change the functionality thereof to display content such as advertising or to provide enhanced telephone features.

Figure **2**

SRA server

Referring to Figure **2**, the SRA server **16** includes a database shown generally at **28**, a database manager **30**, a communications interface **32** and an administrator terminal

34. The database **28** is used to store a plurality of records including telephone number records **36**, customer premises equipment identification (CPE ID) records **38** and feature records **40**.

The telephone number records **36** have telephone number fields **42**, and subscriber information fields **44** while customer premises equipment records **38** have telephone number fields **46** and CPE ID fields **48**. The contents of the telephone number fields **42** and **46** link telephone number records and CPE ID records together when the telephone number fields **42** and **46** have the same contents.

The feature records **40** have respective URL fields **50** for storing a uniform resource locator identifying a first network resource on the world-wide web at which an HTML page including a download applet is available. The download applet is runable by the telephone **20** to provide the telephone with the functionality required to download a feature applet from the same network resource or a different or second network resource.

The database manager **30** receives commands from the administrator terminal **34** to create and amend telephone number records **36** and CPE ID records **38** in the database **28**. A separate telephone number record is associated with each telephone and therefore in this embodiment, telephone number record **36** is associated with telephone **20** shown in Figure **1**. A CPE ID record is also associated with telephone **20**, and has a CPE ID field **48** identifying an address of an equipment block within the telephone. This facilitates a plurality of different equipment to be connected to the same telephone line from the central office switch (**22**), while allowing each different equipment to be individually addressable.

In one scenario, the SRA server **16** is maintained by a telephone company providing telephone services to subscribers such as the user of telephone **20**. The database records are created such that separate telephone number records are associated with respective corresponding subscriber lines and separate equipment ID records **38** are associated with respective corresponding equipment connected to a given line, at the subscriber's premises.

In the event that a user requests a change in telephone features, an administrator enters the change into the administrator terminal **34** which directs the database manager **30** to search the telephone number records and CPE ID records to locate the user's telephone number and CPE ID for the telephone which is to be changed to include the new telephone features. At the same time, the database manager **30** searches the feature records **40** to locate a record containing a uniform resource locator (URL) defining an address of a resource on the internet at which an HTML page including an applet which provides a URL identifying a resource from which the telephone downloads a file using a file transfer protocol (FTP), the file being executable by the telephone to provide extended functionality thereto.

Figure 3

In response to the database search, the database manager produces an SRA control message as shown at **31** in Figure **3**, including a message identifier **33** identifying the type of loop carrier in the connecting path to the subscriber location, a telephone number of a subscriber line **35**, and a CPE ID **37** identifying equipment on the subscriber line to which the SRA control message is addressed, and a URL **39** copied from the uniform resource locator field **50** of the feature record **40**. Referring back to Figure **2**, this SRA control message is provided to the communication interface **32** which transmits it to the central office service unit **18**.

Central Office Service Unit

Still referring to Figure **2**, in this embodiment, the Central Office service unit **18** includes a telemetry network access computer **60**, an SRA communications interface **62** and a central office interface **64** including an FSK transmitter **65**. The SRA communications interface **62** is connected to the communications interface **32** of the SRA server **16** by a public packet switched network link **66** and provides messages received from the SRA server **16** to the telemetry network access computer **60**. The FSK transmitter **65** in the central office interface **64** is controlled by the telemetry network access computer **60** and is connected to the central office (not shown in Figure **2**) by at least one trunk **68** having a suppressed ring access class of service.

In general, the telemetry network access computer **60** receives SRA messages from the SRA server **16** and produces on the trunk **68** suppressed ringing connection (SRC) messages in an FSK format, as shown generally at **70** in Figure **3**. The suppressed ringing connection messages are thus FSK messages.

Referring back to Figure **3**, SRC messages have a format which includes a message type field **72**, a total length field **74**, a CPE ID field **76**, a parameter type field **78**, a parameter length field **80**, and a parameter data field **82**. The message type field identifies the message as a suppressed ring connection message. Predefined codes are used to identify predefined message types. The total length field **74** is used to hold a value representing the total length of the SRC message. The CPE ID field **76** is loaded with the contents of the CPE ID field **37** of the SRA control message from the SRA server. The parameter type field **78** is used to identify the type of data stored in the parameter data field **82** and, in this embodiment, the parameter type field stores a predefined code representing that the contents of the parameter data field **82** relate to

URL data. The parameter length field **80** is used to identify the length of the parameter data field **82**. The parameter data field **82** is, in this embodiment, used to store the uniform resource locator identifying the first network resource or first provider **25**, from the URL field **39** of the SRA control message from the SRA server. The FSK message thus has a first network resource locator identifying the first network resource.

Figure 4

Central office

Referring to Figure **4**, the central office switch **22** includes a program driven digital switch including a switching network **45**, a program controlled processor **47** a trunk unit **57** and a plurality of subscriber interfaces, only two of which are shown at **59** and **61**.

The trunk unit **57** senses activation by the central office service unit and signals the program controlled processor **47**, to indicate such activation. The program controlled processor then controls the switching network **45** to provide voice path connections for communicating suppressed ring access messages from the central office service unit **18** to subscriber locations such as the telephone **20** served by the central office switch **22**. Communication between each subscriber location and the switching network **45** occurs via the subscriber interfaces, each of which includes a line unit **49**, a digital loop carrier **51** at the switch location, a transmission path **52**, a digital loop carrier **54** at a remote location and a subscriber line **56**, **58**.

Generally, the trunk **68** has a class of service which identifies to the central office switch **22** that the connections being requested by the central office service unit **18** are to be made with ringing suppressed, ie., with no ringing signal transmitted or with a burst of ringing signal transmitted of sufficient length to activate the

digital loop carriers **51** and **54** but of insufficient length to be present for any significant time after connection is completed. Accordingly, the central office service unit **18** can establish connections through the central office switch **22** to the subscriber locations, and such connections are accompanied by no or little (ie., short) ringing. These connections facilitate the transmission of suppressed ring access messages in an FSK format to the telephone. In other words, the central office service unit places a call through the central office on a trunk line having a special class of service. The central office detects that a call is being made on the special trunk line and activates the digital loop carriers for the line to which the call is being made, to enable communication between the central office service unit and the telephone, without activation (suppressed ringing) or with minium activation (single ring) of ring tone generation circuitry (not shown) in the central office. Thus, a communication link is established between the central office service unit **18** and the telephone to permit the FSK suppressed ring access message to be transmitted from the central office service unit to the telephone.

Telephone

Referring to Figure **4**, the telephone is shown generally at **20** and includes a system microprocessor **90**, a telephony microcontroller **92**, and a central office line interface **94** in communication with both the system microprocessor **90** and the telephony microcontroller **92**. The system microprocessor is connected to a keyboard interface **96** and a keyboard **98**, for receiving user input for commanding the system microprocessor **90** to effect certain functionality. The system microprocessor is further connected to a display interface **100** which is further connected to a display **102**, for displaying output to a user.

The telephony microcontroller **92** is in communication with the system microprocessor **90** by a plurality of signal lines **104** and is further in communication with a handset interface **105** for controlling signals to and from a conventional handset **106** and a hands-free interface **107** for providing conventional speakerphone functionality to the telephone.

The telephony microcontroller **92** is further in communication with a dial pad **108** of the conventional type, for enabling a user to dial telephone numbers at the telephone.

Figure 5

Referring to Figure **5**, the central office line interface is shown in greater detail at **94**. The central office line interface has first and second telephone line inputs **110** and **112** which are connected to the first and second telephone lines **56** and **58** respectively. The first line input **110** is connected to a line **1** modem terminal **114**, a two to four wire interface terminal **116** and an on-hook extension in use detector **118**. The modem terminal **114** is connected to a line selector switch shown generally at **120** having a selector terminal **122** and a line **2** modem terminal **124**. The line selector switch **120** is controlled by a line selector control circuit **126** to connect the selector terminal to either the line **2** modem terminal **124** or the line **1** modem terminal **114**. The selector terminal **122** is further connected to a modem hook switch **128** which is controlled by a modem **130** in communication with the system microprocessor **90**, shown in Figure **4**.

Referring back to Figure **5**, the modem hook switch **128** is connected to a data access arrangement **132** which isolates the modem from the telephone line and provides a DC path for seizing the line. The data access arrangement **132** is in communication with the modem **130**, to provide analog

signals to and from the modem **130**, for communication to the system microprocessor.

The central office line interface further includes a line hook switch **134** and a two wire to four wire hybrid circuit **136**. The hook switch **134** is controlled by a line hook switch control circuit **138** which opens and closes the hook switch **134** thereby connecting and disconnecting the first telephone line to and from the two to four wire hybrid circuit **136**.

The two to four wire hybrid circuit **136** has a transmit input **140** and a receive output **142** and an off-hook detect signal output **144**. The transmit input **140** is connected to a DTMF generator **146** and a transmit output of an audio circuit **148**. The DTMF generator is operable to provide tones to the transmit input **140** to provide DTMF tones on the first telephone line **56**.

The receive output **142** of the two to four wire hybrid circuit **136** is connected to a receive input of the audio circuit **148**, and is further connected to an FSK receiver **150** and a caller alerting signal receiver **152**. The two to four wire hybrid circuit thus acts as a receiver or means for receiving the FSK message from a first telephone line and providing the FSK message to the FSK receiver and the FSK receiver acts as a receiver or means for receiving an FSK message having a first network resource locator identifying the first resource.

The off-hook EIU detect output **144** of the two to four wire hybrid circuit **136** is connected to an off-hook EIU detect circuit **154** which provides a signal indicating whether or not an extension is in use on the first telephone line, at the same time that the hook switch **134** is closed.

The on-hook extension in use detector **118** monitors the first telephone line **56** to provide a signal indicating whether or not an extension is in use on the first telephone line, when the hook switch **134** is in the open position.

The line selector control circuit **126**, the line hook switch circuit **138** and the DTMF generator **146** are operable to receive signals from the telephony microcontroller **92** shown in Figure **4**, to control the state of the line selector switch **120**, to selectively connect the modem **130** and data access arrangement **132** to the first or second telephone lines **56** or **58** and to control the state of the line hook switch **134** and to provide DTMF signals to the two to four wire interface for transmission on the first telephone line, respectively. The modem **130** and data access arrangement thus act as a second telephone line interface.

The audio circuit **148** is in bi-directional communication with the telephony microcontroller which effectively provides an audio path, with audio signal processing to the handset interface **105** or hands-free interface **107** shown in Figure **4**.

The FSK receiver **150**, caller alerting signal receiver **152**, off-hook EIU detect circuit **154** and on-hook extension in use detector **118** provide FSK, CAS, off-hook EIU and on-hook EIU signals respectively to the telephony microcontroller **92**, which communicates these signals to the system microprocessor **90**.

Referring back to Figure **4**, the system microprocessor **90** is further in communication with FLASH memory **156** and random access memory (RAM) **158**.

Figure 6

Referring to Figure **6**, the FLASH memory holds a directory structure as shown generally at **161**. The directory structure includes a base directory having a hardware abstraction layer file **163**, an operating system file **165**, a system subdirectory **167**, a data subdirectory **169**, a Java subdirectory **171** and a temporary subdirectory **173**.

The hardware abstraction layer file **163** includes basic boot commands for booting up the system processor and for directing it to load the operating system file **165** into RAM, where it is run by the processor.

The operating system file **165** contains operating system commands for establishing basic operation of the system microprocessor **90**. The operating system commands include a Board Support Package (BSP) and drivers which direct the processor to interact with the central office line interface **94**, the telephony microcontroller **92**, the keyboard interface **96** and the display interface **100** shown in Figure **4**. In this embodiment, the operating system is known as VX Works (tm) provided by Windriver of Alameda, California. The operating system maintains a routing table **155** in the RAM **158** shown in Figure **4** for maintaining details of connections established through the central office line interface **94** on the second subscriber line **58** shown in Figure **4**.

The operating system file further include code implementing a Java (tm) virtual machine and a graphics layer known as RTX X-Windows server (tm) by Visicom of San Diego, California. The Java virtual machine includes core Java packages including Java.lang, Java.io, Java.applet, Java.net and Java.awt which in connection with the operating system files create a Java run time environment at the system microprocessor **90**.

Still referring to Figure **6**, the system subdirectory **167** includes an applications subdirectory **175** and an extensions subdirectory **177**. The applications subdirectory **175** includes a telephone subdirectory **179** containing files for directing the system microprocessor **90** to cooperate with the telephony microcontroller **92** to provide telephony functionality. The applications subdirectory further includes an event parser subdirectory **181** containing event parser files, a main event handler subdirectory **183** containing main event handler files, a browser subdirectory **185** containing browser files, a public interface subdirectory **187** containing public interface method files **189** and a handler subdirectory **191** containing handler files.

The extensions subdirectory **177** includes an email subdirectory **193** containing email files for providing email functionality, a fax interface subdirectory **195** containing fax files for providing fax functionality, a games subdirectory **197** containing game files for providing game functionality, a horoscope subdirectory **199** containing horoscope files for communicating with horoscope content providers, a sports subdirectory **201** containing sports provider files for communicating with sports information providers, a stocks subdirectory **203** containing stock provider communication files for communicating with stock information providers and a marketing subdirectory **205** containing marketing files for providing display advertising on the display **102**.

The data subdirectory **169** includes a configuration data subdirectory **207** containing configuration files including a CPE ID file **210** and an event code lookup table file **208**. The CPE ID file **210** holds a CPE ID of the telephone, to identify the telephone from other telephones which may be connected to the same telephone line. The event code lookup table file **208** contains an event code lookup table

having parameter type and URL event code fields **211** and **213** respectively, for associating parameter types with URL event codes.

The Java subdirectory **171** includes Java class files of the type classes.jar.

The temporary subdirectory **173** is used to store applets, HTML pages and files which may be downloaded over one of the analog subscriber lines **56** and **58** shown in Figure **4**.

In general, the FLASH memory **156** stores groups of processor readable codes for directing the processor to effect telephone functionality and thus acts as a computer-readable storage medium for directing the processor to carry out the method described herein.

Referring to Figure **7**, the RAM **158** is used to maintain an FSK receive buffer **200**, a variable buffer **202**, a main event handler buffer **204**, an event queue buffer **206** and a routing table **155**.

The FSK receive buffer **200** is of sufficient length to receive FSK messages, in general. Such messages may include caller ID information and the like and, more particularly, such messages may include SRC messages sent by the central office service unit **18** shown in Figure **1**.

The variable buffer **202** and the main event handler buffer **204** hold variable and main event records respectively. The variable records include a parameter type field **215** and a data field **216** and the main event records include a parameter type field **217** and a data field **209**. The event queue buffer **206** is a variable length buffer for storing various event queue records, each of which includes an event code field **248** and a data field **250**.

Figure 8

Referring briefly to Figure **4**, the system microprocessor **90**, FLASH memory **156**, and RAM **158** together form a computer architecture. Referring to Figure **8**, this architecture is shown generally at **162**.

Figure 9

Referring to Figure **9**, the event parser files in the event parser directory establish an event parser thread as shown generally at **220**. The event parser thread includes a read comm port instance **222** which directs the system microprocessor **90** shown in Figure **4**, to communicate with the telephony microcontroller to read the FSK receiver **150** to determine whether or not an FSK message is being received at the FSK receiver **150**. If a message is not being received, the event parser thread is blocked on the read comm port instance **222**. If the FSK receiver **150** is receiving an FSK message, such message is passed to the system microprocessor and stored in the FSK receive buffer **200**.

The event parser thread then includes a checksum instance **224** which directs the processor to read the contents of the FSK receive buffer **200** to determine whether or not the message was received properly. If the message was not received properly, it is ignored and the processor is directed back to the read comm port instance **222**.

If the message was received properly, the processor is directed to a CPE ID match instance **226** in which the contents of the CPE ID register **210** stored in the FLASH memory **156** is compared against the contents of the CPE ID field of the FSK message stored in the FSK receive buffer **200**. If the CPE IDs do not match, the processor is returned to the read comm port instance **222**.

If the CPE IDs match, a no activity instance **228** directs the processor to determine whether or not a user has initiated any activity at the telephone within the last ten minutes.

If the user has not made use of the telephone within the last ten minutes an EIU instance **230** directs the system microprocessor **90** to communicate with the telephony microcontroller **92**, causing it to determine whether or not the EIU signal provided by either of the EIU detectors **118** and **154** shown in Figure **5**, indicates that an extension is in use. If so, then the processor is directed back to the read comm port instance **222**. If an extension is not in use, then the processor is directed to a store parameter type instance **232** which directs the processor to store the contents of the parameter type field and the contents of the data field in the variable buffer **202**. Thus, the first network resource locator is copied to the variable buffer **202**. The event parser thus acts as a receiver or means for receiving an FSK message having a first network resource locator identifying the first resource. It also acts as means for extracting the first uniform resource locator from the FSK message. The system microprocessor **90** is then directed to a copy instance **234** which directs it to copy the contents of the variable buffer **202** to the main event handler buffer **204**. The processor is then returned back to the read comm port instance **222** where it blocks until a further FSK message is received.

Figure **10**

Main event handler thread

Referring to Figure **10**, the main event handler files in the main event handler subdirectory **183** establish a main event handler thread as shown generally at **240**. The main event handler thread includes a buffer loaded instance **242** which directs the system microprocessor **90** to determine whether or not the main event handler buffer **204** shown in Figure **7**

is full. If this buffer is not loaded then the processor blocks on the buffer loaded instance **242**.

If the main event handler buffer is loaded, the processor is directed to a lookup table instance **244** which directs the processor to find an event code record in the event code lookup table **208** shown in Figure **6**, having parameter type field **211** contents corresponding to the contents of the parameter type field in the main event handler buffer **204** shown in Figure **7**.

On finding such an event code record, the system microprocessor **90** is directed to a copy event code instance **246** which directs the processor to copy the contents of the event code field **213** from the event code record into the event queue buffer **206** shown in Figure **7** and also to copy the contents of the data field **209** of the main event handler buffer **204** shown in Figure **7** to the event queue buffer **206** in association with the event code. Thus, the processor stores, in the event queue buffer **206**, a first event record including an event code from the event code lookup table **208** and data from the data field **250** of the main event handler buffer **204** shown in Figure **7**.

Figure **11**

Referring to Figure **11**, the event dispatcher files in the main event handler subdirectory **183** shown in Figure **6** establish an event dispatcher thread as shown generally at **180**. The event dispatcher thread includes a first instance which directs the system processor to address the next event code record in the event queue buffer **206** shown in Figure **7**. The system processor is then directed to instance **184** which directs it to determine which threads have registered for the event code record addressed at instance **182**. The registration of threads with event code records is assumed to have been previously accomplished according to conventional practices. After having

determined which threads have registered for the current event code record, instance **186** directs the processor to communicate the current event code record from the event queue buffer **206** to the registered threads. In this manner, data is passed from the event queue buffer **206** to appropriate threads.

Figure **12**

Referring to Figure **12**, a browser thread **260** is implemented by the browser files in the browser subdirectory **185** shown in Figure **6**. The browser thread includes a pend instance **262** which directs the processor to pend for receipt of a URL event code record from the event dispatcher thread **180** shown in Figure **11**. The event dispatcher in conjunction with the event parser and the main event handler and the processor **90**, thus cooperate to present the first uniform resource locator from the FSK message to the browser.

On receipt of a URL event code record from the event dispatcher thread, a request instance **264** directs the system microprocessor **90** to request a connection to a world-wide web resource specified by the URL in the data field **250** of the event record. Such connection is requested using classes in the Java.net package of the Java virtual machine.

Figure **13**

Referring to Figure **13**, if a connection is requested by an application, the network driver **270** included within the operating system file **165** shown in Figure **6** is run. The network driver task includes a first instance **272** which directs the system microprocessor **90** to pend awaiting a connection request from the browser thread shown in Figure **12**.

On receiving a connection request, instance **274** directs the system microprocessor **90** to check the routing table **155**

shown in Figure **7** to determine whether or not a network connection has already been established. Referring to Figure **13**, if no such connection has been established, instance **276** directs the system microprocessor **90** to command the modem **130** to connect to the subscriber line **58** to place a telephone call over the public switched telephone network to the Internet services provider **24** shown in Figure **1**.

Referring back to Figure **13**, establishment of a connection with the Internet services provider involves a handshaking instance **278**.

On establishing connection with the Internet services provider, instance **280** directs the processor to negotiate a point to point protocol (PPP) connection with the Internet services provider and to store the details of the PPP connection in the routing table **155**. A network connection is thus created. The network driver, processor **90** and modem **130** thus cooperate to act as means for using the second telephone line interface for communicating on the network.

Instance **282** then directs the system microprocessor **90** to notify the requesting application of the connection to the resource specified by the URL in the data field **250** of the event record.

On completion of transmission of the details to the browser thread, the system microprocessor **90** is directed back to instance **272** where it pends on receipt of a further connection request from the browser thread.

It should be noted that at instance **274**, where the processor is directed to determine whether or not a connection has already been established, if such a connection has already been established, the processor is

directed directly to instance **282** where it notifies the requesting application of such connection.

Referring back to Figure **12**, notification of the connection established by the network driver task shown in Figure **13** is received at instance **290** in Figure **12**.

Still referring to Figure **12**, if no connection has been made and no connection can be established, instance **292** directs the system microprocessor **90** to display an error banner to indicate that the connection could not be established. The processor is then returned back to the pend instance **262**.

If at instance **290**, notification of a connection has been established, instance **294** directs the processor to download and decode an HTML file including an applet including a second network resource locator from the first resource. The HTML file is received and stored in the temporary directory **173** in the FLASH **156**. The browser and system processor thus act as means for downloading from a first resource identified by a first network resource locator, an applet including a second network resource locator.

Instance **295** then directs the processor to determine whether or not the HTML file has been successfully decoded, according to well known browser decoding criteria. In this embodiment, the browser includes a security manager which imposes restrictions on what hosts any applets within the HTML file can communicate with. In particular in this embodiment, applets are restricted from reading or writing to the flash memory shown in Figure **6**. Applets may only open socket connections to the host that served them. They are not permitted to open server sockets themselves, they are not permitted to start other processes on the local host and they are not permitted to have native methods.

The security manager thus acts as means for preventing running applets from accessing the memory.

If the HTML file was not successfully decoded, the processor is directed to the error banner instance **292**. If the HTML file was successfully decoded, instance **296** directs the processor to spawn threads on each applet, if any, contained in the HTML file. After spawning such threads, the processor is returned to the pend instance **262**. The browser and processor **90** thus act as means for running the applet.

Figure **14**

Referring to Figure **14**, an applet thread, such as spawned at instance **296** of the browser thread shown in Figure **12**, is shown generally at **310** in Figure **14**. The applet thread includes a first instance **312** which directs the system microprocessor **90** to pass a pre-stored, second uniform resource locator (URL) contained within the applet to the public interface method shown in Figure **15**. Instance **314** then directs the processor to pend, awaiting a completion notice from the public interface method shown in Figure **14**.

Figure **15**

Referring to Figure **15**, the public interface method is shown generally at **320** and begins with instance **324** which directs the system microprocessor **90** to receive the second URL from the calling applet. The public interface method thus acts as means for communicating with the applet to receive the second network resource locator from the applet. Instance **326** then directs the processor to create a download event code record in the event queue buffer **206** shown in Figure **7**. The download event record includes a download event code, the URL received from the applet and a password. Instance **328** then directs the processor to pend for receipt of a completion notice from the handler thread shown in Figure **16**.

Figure 16

Referring to Figure **16**, the handler thread is shown generally at **400** and includes a first instance **402** at which it pends on receipt of a download event record from the event dispatcher shown in Figure **11**. (In the description thus far, the download event record was entered into the event queue by the public interface method.)

Upon receipt of a download event record, instance **404** compares the password from the download event record with a pre-defined password, and if the password does not match, instance **412** directs the processor to send a message to the public interface to indicate that a download could not occur.

If at instance **404** the password is authenticated, instance **406** directs the processor to receive files using the file transfer protocol, from the resource identified by the URL. At the second network resource specified by the second URL, files operable to be downloaded are named and formatted into directories of the same names as those in the FLASH memory **156** of the telephone and are grouped into Java file packages named after the corresponding directories. The Java file packages are compressed according to the ZIP (tm) algorithm and are stored at the second resource as ZIP files named after the corresponding file packages. The ZIP files are transmitted to the system microprocessor **90** in the telephone, using the file transfer protocol.

When the ZIP files are received at the system processor, they are un-compressed or exploded and are stored in the temporary subdirectory **173**. Instance **404** of the handler thus acts as means for decompressing the file from the second network resource to produce a plurality of decompressed files. More generally, the handler and processor act as means for communicating with a second

network resource identified by the second network resource locator to receive and store in the memory, a file from the second network resource.

Instance **408** then directs the processor to verify and check the files received by determining whether or not checksums match and whether or not the un-compressed files have acceptable package names. Instance **408** thus acts as means for verifying the file has been received in accordance with pre-defined criteria. The package names thus act as passwords which are compared against predefined passwords which, in this embodiment, are the directory names in the FLASH memory. If the files do not have acceptable package names, instance **410** directs the processor to remove the files from the temporary storage directory and to proceed to instance **412** to notify the public interface that the files are not acceptable.

If the files are acceptable, instance **414** directs the processor to move them from the temporary directory to the appropriate directories in the file system, as determined by the names and directory structures associated with the files. The handler thread thus acts as means for extracting a password from the file and for comparing the password against a pre-defined password and to store the file in the memory when the password matches the pre-defined password and means for extracting a filename from the file from the second resource and for loading the file into an area of memory associated with the filename, where the area of memory associated with the filename is the corresponding directory in the FLASH memory **156**.

After the files have been moved, instance **416** directs the processor to notify the public interface method of completion and the processor is returned to the pend instance **402**.

Referring back to Figure **15**, when the notice of completion from the handler thread is received at instance **328**, instance **330** directs the processor to notify the applet thread shown in Figure **14**, of completion and the public interface method shown in Figure **15** is completed.

The applet thread shown in Figure **14** is thus notified of completion at instance **314** and the applet thread is completed.

Operation

Referring to Figure **1**, the content provider **26** includes a web server **350** and a database of compressed files **352** containing Java classes for implementing the Java applications in the applications subdirectory **175** shown in Figure **6**. The compressed files are compressed according to the ZIP(tm) format and may include directories of files. Thus, entire directories and files are zipped into a compressed format and are stored as a single file in the database.

The first Java applet provider **25** includes a web server **354** and a plurality of Java applets **356** containing uniform resource locators identifying the first content provider and compressed files stored in the content provider database files **352**.

Referring to Figure **2**, the feature records **40** have URL fields **50** which identify the individual applets **356** at the Java applet provider **25** shown in Figure **1**. Thus, when a user requests an additional feature such as email capability, fax interface, screen saver, etc., the user pays an administrator of the service, and the administrator enters at the administrator terminal **34** shown in Figure **2**, a request for a change in service, the user's telephone number, the CPE ID of the equipment to be changed and an identification of the desired feature. In response, the

SRA server **16** dispatches an SRA message on public packet switched network link **66**, in the format shown at **31** in Figure **3**.

Referring back to Figure **1**, the SRA message is received at the central office service unit **18** which establishes a connection between it and the central office switch **22** over the trunk **68** enjoying a special class of service to establish a suppressed ring access connection to the user telephone **20**. When such connection has been made, the central office service unit **18** sends a suppressed ring connection message as shown generally at **70** in Figure **3**, to the telephone **20** as shown in Figure **1**.

The telephone receives the SRC message **70** through the event parser thread **220** shown in Figure **9** and the main event handler thread **240** shown in Figure **10** which load the event queue buffer **206** with a URL event code and the contents of the parameter data field **82** which contains the URL of an HTML file **356** containing a Java applet at the Java applet provider **25** as shown in Figure **1**. The HTML file includes an applet with a URL identifying a file on the first content provider **26**.

Referring to Figures **1** and **12**, the browser thread **260** directs the system microprocessor **90** to establish a connection with the Internet services provider **24** and the network driver **270** shown in Figure **13** establishes a network connection between the telephone **20** and the Java applet provider **25** through the central office switch **22**, the Internet services provider **24** and the world-wide web **23**. The Java applet provider **25** transmits the HTML file containing the applet back to the telephone **20** and such HTML file is received in the temporary directory **173** shown in Figure **6**. The browser thread **260** then spawns applet threads on each applet included in the HTML file received.

The applets provided by the Java applet provider **25** include uniform resource locators identifying compressed files **352** at the content provider **26** and thus referring to Figure **14**, the applet calls the public interface method **320** shown in Figure **15** and passes to the public interface method, the URL of the first file in the content provider **26** shown in Figure **1**.

The public interface method **320** then creates a download event record in the event queue buffer **206** shown in Figure **7**, indicating that a URL event has occurred and specifies the URL provided by the applet for use by the handler thread **400** in Figure **16**.

The handler thread **400** receives the URL event code from the event dispatcher shown in Figure **11** and makes a connection to the first content provider **26**. The file identified by the URL is transmitted back to the telephone **20** using the file transfer protocol. Instance **408** verifies the file as being an approved type for inclusion in the extensions subdirectory **177** shown in Figure **6** and the file is copied from the temporary subdirectory **173** into the appropriate sub-directory within the extensions subdirectory **177**. The classes contained with the file may then interact within other classes within the system directory **167** and control system resources to effect the new functionality requested by the user.

It will be appreciated that files for implementing certain functions such as email, fax interface and the like, are relatively large files including many kilobits of program code. Thus, use of the suppressed ring access connection to transmit to the telephone the location of an applet identifying a further location, where such a relatively large file is located, effectively enables a telephone to be informed that a file is to be downloaded. The subsequent downloading of the file using the applet and the

content provider is done using the modem for data transfer rather than by simple FSK messaging which would take an inordinately long time to transfer such large files. Thus, the FSK messaging is merely used to initiate the download process and the actual download process takes place according to conventional rules for high speed data transmission using modems.

The use of the applet server, rather than directly accessing the content provider, in conjunction with a level of security which prevents applets from access to system resources within the telephone ensures that system resources are guarded from control by programs and applets not authorized to access system resources.

Second embodiment

Referring to Figure **1**, in accordance with a system according to a second embodiment of the invention, there is provided the entire system as described in the first embodiment with the further inclusion of a second Java applet provider **360** having a web interface **362** and a plurality of HTML files **364** containing Java applets and advertising content or advertisements. In addition, this system includes a second content provider shown generally at **366** including a web interface **368** and a plurality of advertising content or marketing HTML files **370**. In this embodiment, the applets **364** include uniform resource locators pointing to respective marketing HTML files **370** at the second content provider **366** and methods for directing the system processor to display such marketing HTML files on the display **102**.

In this second embodiment, the system works generally as described above with the exception that one of the HTML files **370** is received from the second content provider at instance **406** of the handler thread **400** of Figure **16**. After the HTML file is loaded into the marketing directory, an

applet contained therein is run under the browser, which retrieves the advertisement and displays it on the display **102**, such that the advertisement occupies the entire display, with no navigation tool icons or the like to obstruct it.

Thus, the entire content of the HTML file from the content provider is displayed on the display **102**. In this manner, marketing or advertising content is automatically displayed on the display. The browser has no navigation tools or buttons and, therefore, the advertisement is displayed unobstructed.

The HTML file may include a plurality of links which the user can select using an input device (not shown) and such links may be used by the browser to connect to another resource on the world-wide web or to connect to a resource within the telephone itself. Such resource within the telephone itself would be stored in a sub-directory of the system subdirectory **167** shown in Figure **6**.

Applets may be used to create animated marketing advertisements or presentations which may be more appealing to a user. In addition, high resolution pictures may be included within the HTML file as GIF files or the like.

As HTML pages are for use on display screens of a predefined aspect ratio, advertising for use on telephones of the type described, can be simply and easily developed using conventional HTML page development techniques. This facilitates convenient creation of advertising content for the telephone.

Alternatively, the application sub-directory **175** may include applications which automatically cause HTML files locally stored in the marketing subdirectory **205** to be displayed for marketing purposes by creating, in the event

queue buffer **206**, an event record including a URL event code and a URL specifying a resource within the FLASH memory **156**. In this manner, the display of HTML pages can occur autonomously from within the telephone and may initiate the display of HTML files stored within the directory structure or stored at an external resource location.

Effectively, the invention simplifies the download of a single compressed file, including a bundle of files. This allows multiple files and subdirectories to be downloaded to the telephone in a single, quick file transfer. The software bundle guarantees that all necessary files are downloaded. Security is also enforced by refusing any downloads except from password protected applets.

The invention described provides advantages in enabling field up-gradable software components, extensible applications and the ability to dynamically run Java applets. Any system file components of the telephone can be upgraded without the need to upgrade the entire software load. This allows service providers to change specific pieces of the telephone in short downloads thereby changing, removing or adding functionality to the application. The telephone is also easily extendible using Java plug-ins. Such plug-ins may be designed to integrate into the telephone software and allow third party developers to add functionality to the telephone. In addition, Java applets allow network based providers to use standard interfaces provided on the telephone. This allows Java applets to perform basic telephone functions such as dialling a telephone number. Java applets also have access to the same basic graphical components as the telephone application so that applets can be made to look the same as the main application.

While specific embodiments of the invention have been described and illustrated, such embodiments should be considered illustrative of the invention only and not as limiting the invention as construed in accordance with the accompanying claims.

THE EMBODIMENTS OF THE INVENTION IN WHICH AN EXCLUSIVE PROPERTY OR PRIVILEGE IS CLAIMED ARE DEFINED AS FOLLOWS:

1. A method of downloading files from a network into memory in communication with a processor operable to communicate on the network, the method comprising the steps of:

a) downloading to said processor from a first resource identified by a first network resource locator, an applet including a second network resource locator;

b) running said applet on said processor;

c) receiving said second network resource locator from said applet;

d) preventing applets running on said processor from accessing said memory; and

e) establishing communication between said processor and a second network resource identified by said second network resource locator to receive and store in said memory, a file from said second network resource.

2. A method as claimed in claim **1** further including the step of decompressing said file from said second network resource to produce a plurality of decompressed files.

3. A method as claimed in claim **2** further including the step of verifying said file has been received in accordance with pre-defined criteria.

4. A method as claimed in claim 3 further including the step of extracting a password from said file and comparing said password with a predefined password and storing said file in said memory when said password matches said predefined password.

5. A method as claimed in claim 4 further including the step of extracting a filename from said file from said second resource and loading said file into an area of memory associated with said filename.

6. A method as claimed in claim 1 further including the step of receiving an FSK message having said first network resource locator identifying said first resource.

7. A method as claimed in claim 6 further including the step of receiving said FSK message from a first telephone line.

8. A method as claimed in claim 7 further including the step of extracting said first uniform resource locator from said FSK message for presentation to said processor for use in downloading.

9. A method as claimed in claim 8 further including the step of using a second telephone line interface for communicating on said network.

10. An apparatus for downloading files from a network into memory in communication with a processor operable to communicate on the network, the method including the steps of:

a) means for downloading to said processor from a first resource identified by a first network

resource locator, an applet including a second network resource locator;

b) means for running said applet on said processor;

c) means for receiving said second network resource locator from said applet;

d) means for preventing applets running on said processor from accessing said memory; and

e) establishing communications between said processor and with a second network resource identified by said second network resource locator to receive and storing in said memory, a file from said second network resource.

11. An apparatus as claimed in claim **10** further including means for decompressing said file from said second network resource to produce a plurality of decompressed files.

12. An apparatus as claimed in claim **11** further including means for verifying said file has been received in accordance with pre-defined criteria.

13. An apparatus as claimed in claim **12** further including means for extracting a password from said file and for comparing said password with a pre-defined password and storing said file in said memory when said password matches said pre-defined password.

14. An apparatus as claimed in claim **13** further including means for extracting a filename from said file from said second resource and load said file into an area of memory associated with said filename.

15. An apparatus as claimed in claim 10 further including means for receiving an FSK message having said first network resource locator identifying said first resource.

16. An apparatus as claimed in claim 15 further including means for receiving said FSK message from a first telephone line.

17. An apparatus as claimed in claim 16 further including means for extracting said first uniform resource locator from said FSK message for presentation to said processor for use in downloading.

18. An apparatus as claimed in claim 17 further including means for using a second telephone line interface for communicating on said network.

19. A computer readable medium on which is stored a plurality of computer readable codes representing instructions for directing a processor to download files from a network into memory in communication with said processor, said processor being operable to communicate on the network, said instructions directing said processor to:

a) implement a browser operable to download to said processor from a first resource identified by a first network resource locator, an applet including a second network resource locator;

b) run said applet on said processor;

c) receiving said second network resource locator from said applet;

d) prevent applets running on said processor from accessing said memory; and

e) communicate with a second network resource identified by said second network resource locator to receive and store in said memory, a file from said second network resource.

20. A computer readable medium as claimed in claim **19** further including a plurality of computer readable codes for directing said processor to decompress said file from said second network resource to produce a plurality of decompressed files.

21. A computer readable medium as claimed in claim **20** further including a plurality of computer readable codes for directing said processor to verify said file has been received in accordance with pre-defined criteria.

22. A computer readable medium as claimed in claim **21** further including a plurality of computer readable codes for directing said processor to extract a password from said file and compare said password against a pre-defined password and to store said file in said memory when said password matches said pre-defined password.

23. A computer readable medium as claimed in claim **22** further including a plurality of computer readable codes for directing said processor to extract a filename from said file from said second resource and load said file into an area of memory associated with said filename.

24. A computer readable medium as claimed in claim **19** further including a plurality of computer readable

codes for directing said processor to receive an FSK message having a first network resource locator identifying said first resource and receive said first network resource locator from said FSK receiver.

25. A computer readable medium as claimed in claim **24** further including a plurality of computer readable codes for directing said processor to receive said FSK message from a first telephone line and provide said FSK message to said FSK receiver.

26. A computer readable medium as claimed in claim **25** further including a plurality of computer readable codes for directing said processor to extract said first uniform resource locator from said FSK message for presentation to said browser.

27. A computer readable medium as claimed in claim **26** further including a plurality of computer readable codes for directing said processor to use a second telephone line interface for communicating on said network.

28. A file download apparatus for downloading files from a network into memory, said apparatus including a processor operable to communicate on the network, said processor running:

a) a browser for downloading from a first resource identified by a first network resource locator, an applet including a second network resource locator;

b) an applet viewer for running said applet;

c) a public interface for communicating with said applet to receive said second network resource locator from said applet;

d) a security manager for preventing running applets from accessing said memory; and

e) a handler having access to said memory, said handler being in communication with said public interface, for communicating with a second network resource identified by said second network resource locator to receive and store in said memory, a file from said second network resource.

29. An apparatus as claimed in claim **28** wherein said handler is operable to decompress said file from said second network resource to produce a plurality of decompressed files.

30. An apparatus as claimed in claim **29** wherein said handler is operable to verify said file has been received in accordance with pre-defined criteria.

31. An apparatus as claimed in claim **30** wherein said handler is operable to extract a password from said file and compare said password against a pre-defined password and to store said file in said memory when said password matches said pre-defined password.

32. An apparatus as claimed in claim **31** wherein said handler is operable to extract a filename from said file from said second resource and load said file into an area of memory associated with said filename.

33. An apparatus as claimed in claim **21**, further including an FSK receiver for receiving an FSK message having a

first network resource locator identifying said first resource and wherein said browser is operable to receive said first network resource locator from said FSK receiver.

34. An apparatus as claimed in claim **33** further including a first telephone line interface for receiving said FSK message from a first telephone line and for providing said FSK message to said FSK receiver.

35. An apparatus as claimed in claim **34** further including an event parser for extracting said first uniform resource locator from said FSK message for presentation to said browser.

36. An apparatus as claimed in claim **35** further including a second telephone line interface, said processor using said second telephone line interface for communicating on said network.

37. A method of remotely configuring a telephone having a processor operable to communicate on a network and memory accessible by said processor for storing groups of processor readable codes for directing said processor to effect telephone functionality, the method including the steps of:

a) downloading to said processor from a first network resource on the network, an applet including a second network resource locator;

b) running said applet on said processor;

c) to receiving said second network resource locator from said applet;

d) preventing applets running on said processor from accessing said memory; and

e) establishing communications between said processor and a second network resource identified by said second network resource locator to receive and storing in at least one of said groups, a file from said second network resource.

38. A remotely configurable telephone having a processor operable to communicate on a network and memory accessible by said processor for storing groups of processor readable codes for directing said processor to effect telephone functionality, said processor being programmed to run:

a) a browser for downloading to said processor from a first network resource on the network, an applet including a second network resource locator;

b) an applet viewer for running said applet on said processor;

c) a public interface for communicating with said applet to receive said second network resource locator from said applet;

d) a security manager for preventing applets running on said processor from accessing said memory; and

e) a handler providing said processor access to said memory, said handler being in communication with said public interface, for communicating with a second network resource identified by said second network resource locator to receive and store in

at least one of said groups, a file from said second network resource.

39. An apparatus for remotely configuring a telephone having a processor operable to communicate on a network and memory accessible by said processor for storing groups of processor readable codes for directing said processor to effect telephone functionality, the method including the steps of:

a) downloading to said processor from a first network resource on the network, an applet including a second network resource locator;

b) running said applet on said processor;

c) receiving said second network resource locator from said applet;

d) preventing applets running on said processor from accessing said memory; and

e) establishing communications between said processor and a second network resource identified by said second network resource locator to receive and store in at least one of said groups, a file from said second network resource.

40. A computer readable medium on which is stored a plurality of computer readable codes representing instructions for directing a processor to remotely configure a telephone having a processor operable to communicate on a network and memory accessible by said processor for storing groups of processor readable codes for directing said processor to said processor

to effect telephone functionality, the instructions directing said processor to:

a) download from a first network resource on the network, an applet including a second network resource locator;

b) run said applet;

c) communicate with said applet to receive said second network resource locator from said applet;

d) prevent running applets from accessing said memory; and

e) communicate with said public interface, to communicate with a second network resource identified by said second network resource locator to receive and store in at least one of said groups, a file from said second network resource.

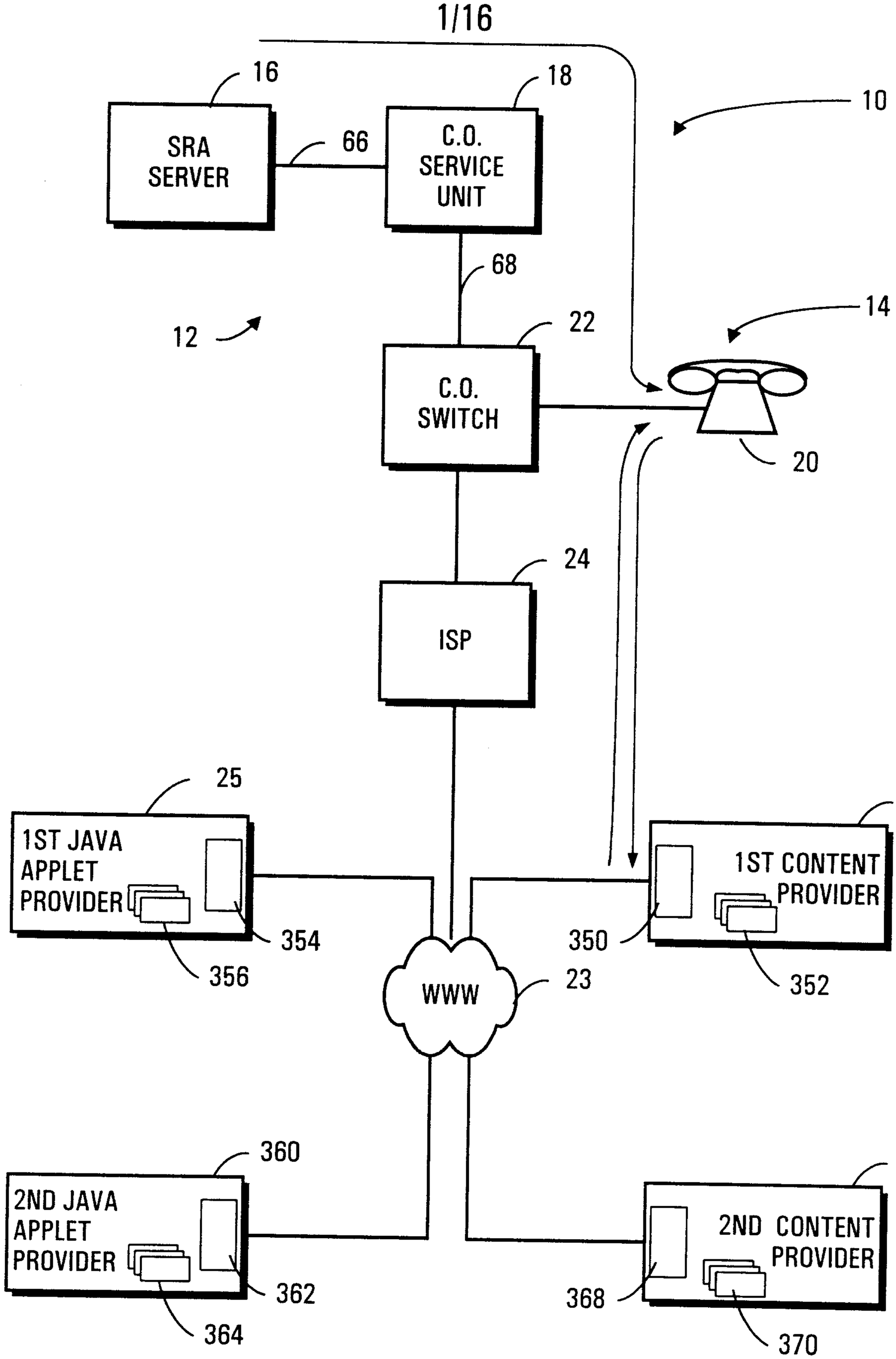

16
18
10
SRA SERVER
66
C.O. SERVICE UNIT
68
12
22
14
C.O. SWITCH
20
24
ISP
25
26
1ST JAVA APPLET PROVIDER
1ST CONTENT PROVIDER
354
350
356
352
WWW
23
360
366
2ND JAVA APPLET PROVIDER
2ND CONTENT PROVIDER
362
368
364
370

FIG. 1

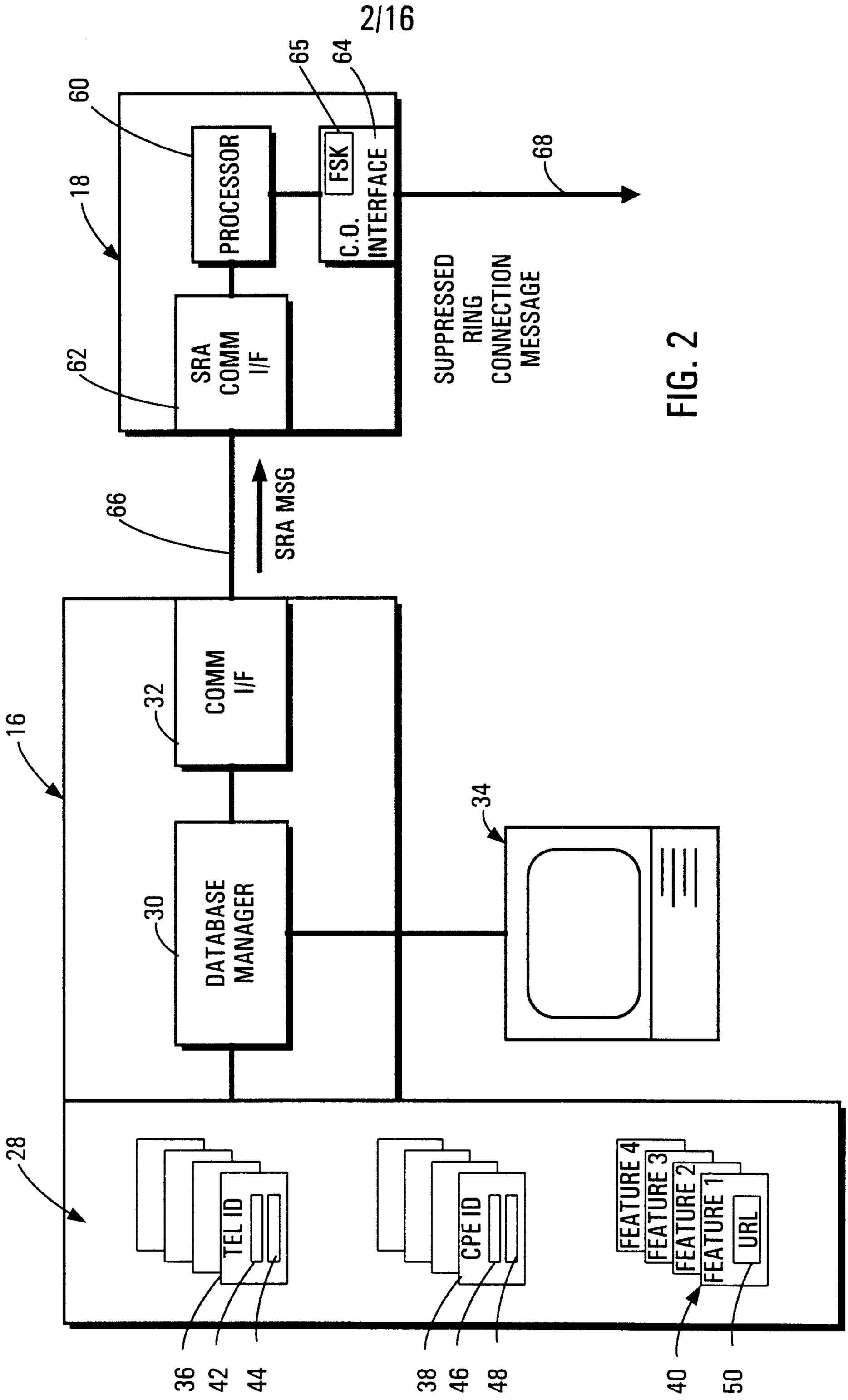

60
65
64
18
PROCESSOR
FSK
C.O.
INTERFACE
68
SUPPRESSED
RING
CONNECTION
MESSAGE
62
SRA
COMM
I/F
66
SRA MSG
16
32
COMM
I/F
30
DATABASE
MANAGER
34
28
TEL ID
CPE ID
FEATURE 4
FEATURE 3
FEATURE 2
FEATURE 1
URL
36
42
44
38
46
48
40
50

FIG. 2

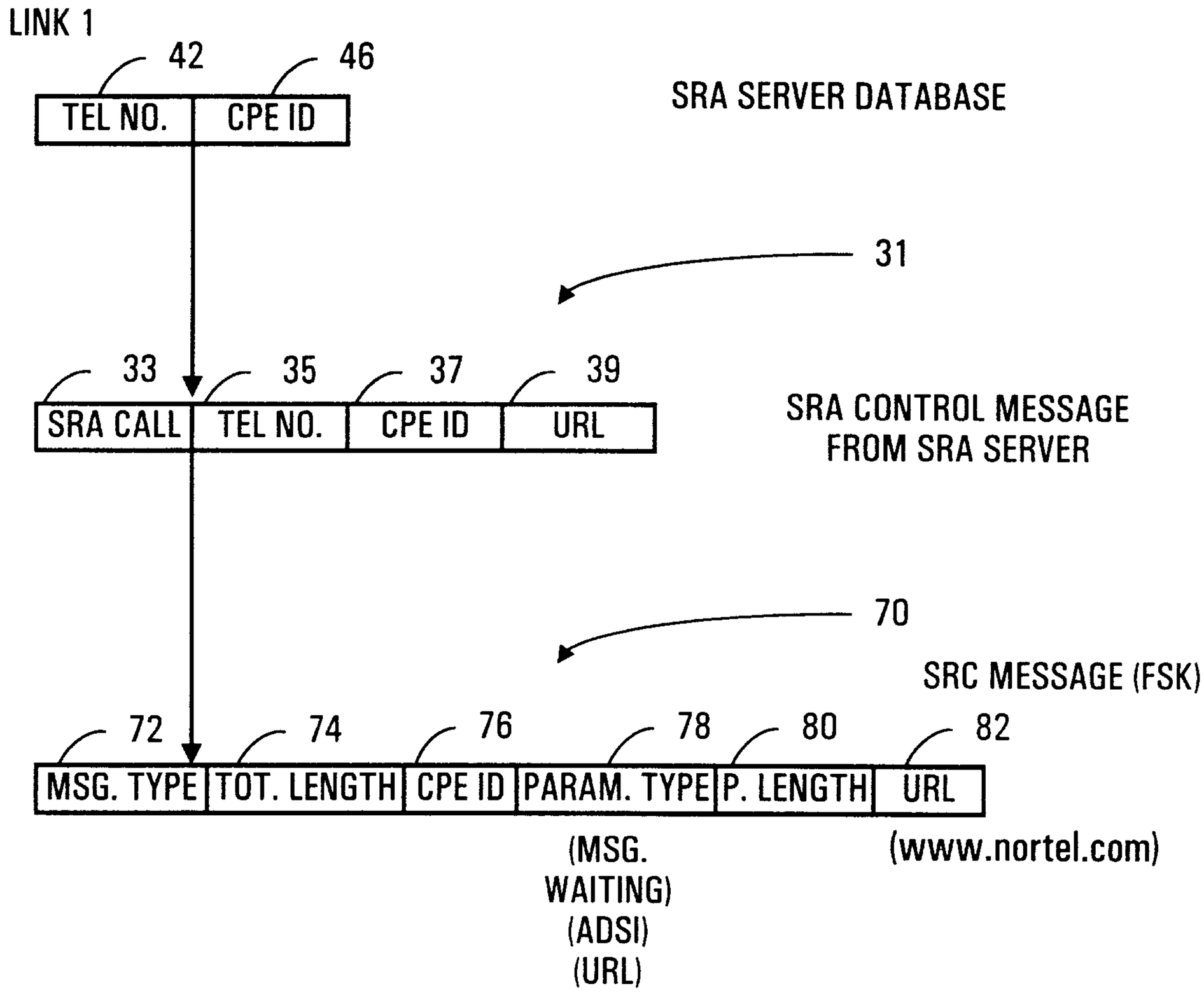

LINK 1
42
46
TEL NO.
CPE ID
SRA SERVER DATABASE
31
33
35
37
39
SRA CALL
TEL NO.
CPE ID
URL
SRA CONTROL MESSAGE FROM SRA SERVER
70
SRC MESSAGE (FSK)
72
74
76
78
80
82
MSG. TYPE
TOT. LENGTH
CPE ID
PARAM. TYPE
P. LENGTH
URL
(MSG. WAITING)
(ADSI)
(URL)
(www.nortel.com)

FIG. 3

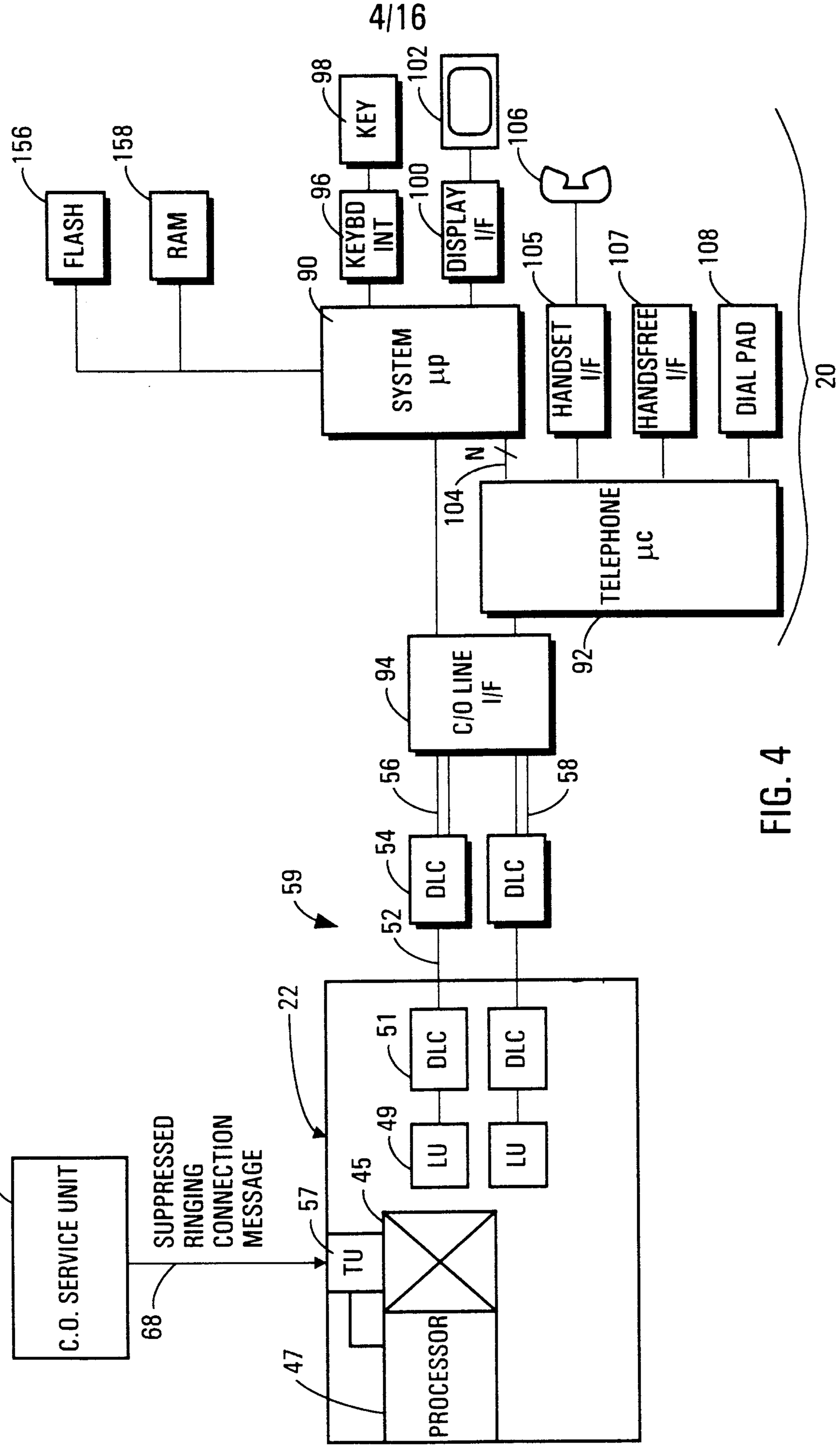

156
FLASH
158
RAM
98
KEY
102
106
96
KEYBD INT
100
DISPLAY I/F
90
105
107
108
SYSTEM µp
HANDSET I/F
HANDSFREE I/F
DIAL PAD
20
N
104
TELEPHONE µc
92
94
C/O LINE I/F
56
58
54
DLC
DLC
59
52
22
51
DLC
DLC
49
LU
LU
45
57
TU
PROCESSOR
47
18
C.O. SERVICE UNIT
SUPPRESSED RINGING CONNECTION MESSAGE
68

FIG. 4

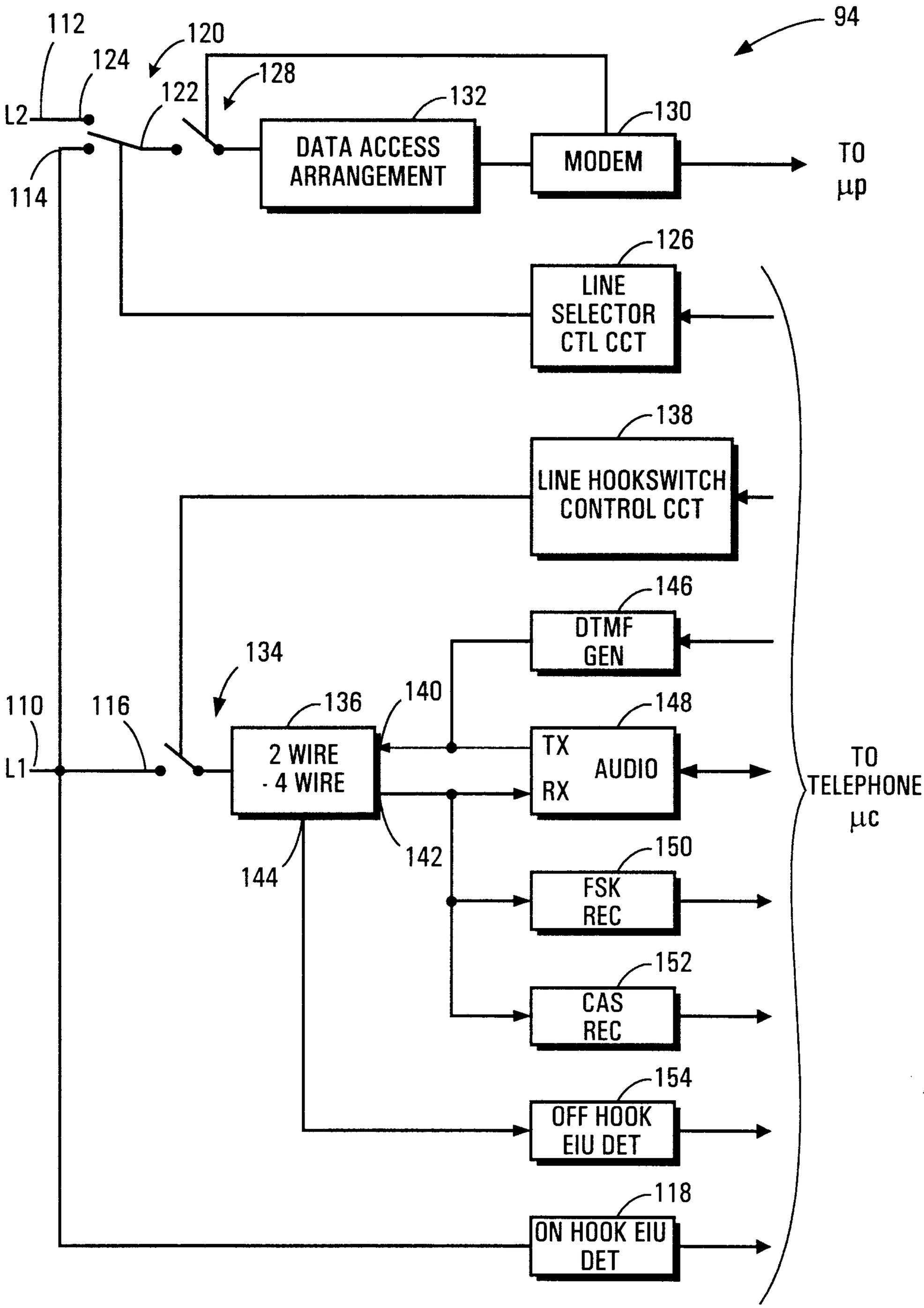

112
120
94
124
128
122
132
130
L2
DATA ACCESS ARRANGEMENT
MODEM
TO µp
114
126
LINE SELECTOR CTL CCT
138
LINE HOOKSWITCH CONTROL CCT
146
DTMF GEN
134
110
116
140
136
148
2 WIRE - 4 WIRE
TX
AUDIO
RX
TO TELEPHONE µc
L1
142
144
150
FSK REC
152
CAS REC
154
OFF HOOK EIU DET
118
ON HOOK EIU DET

FIG. 5

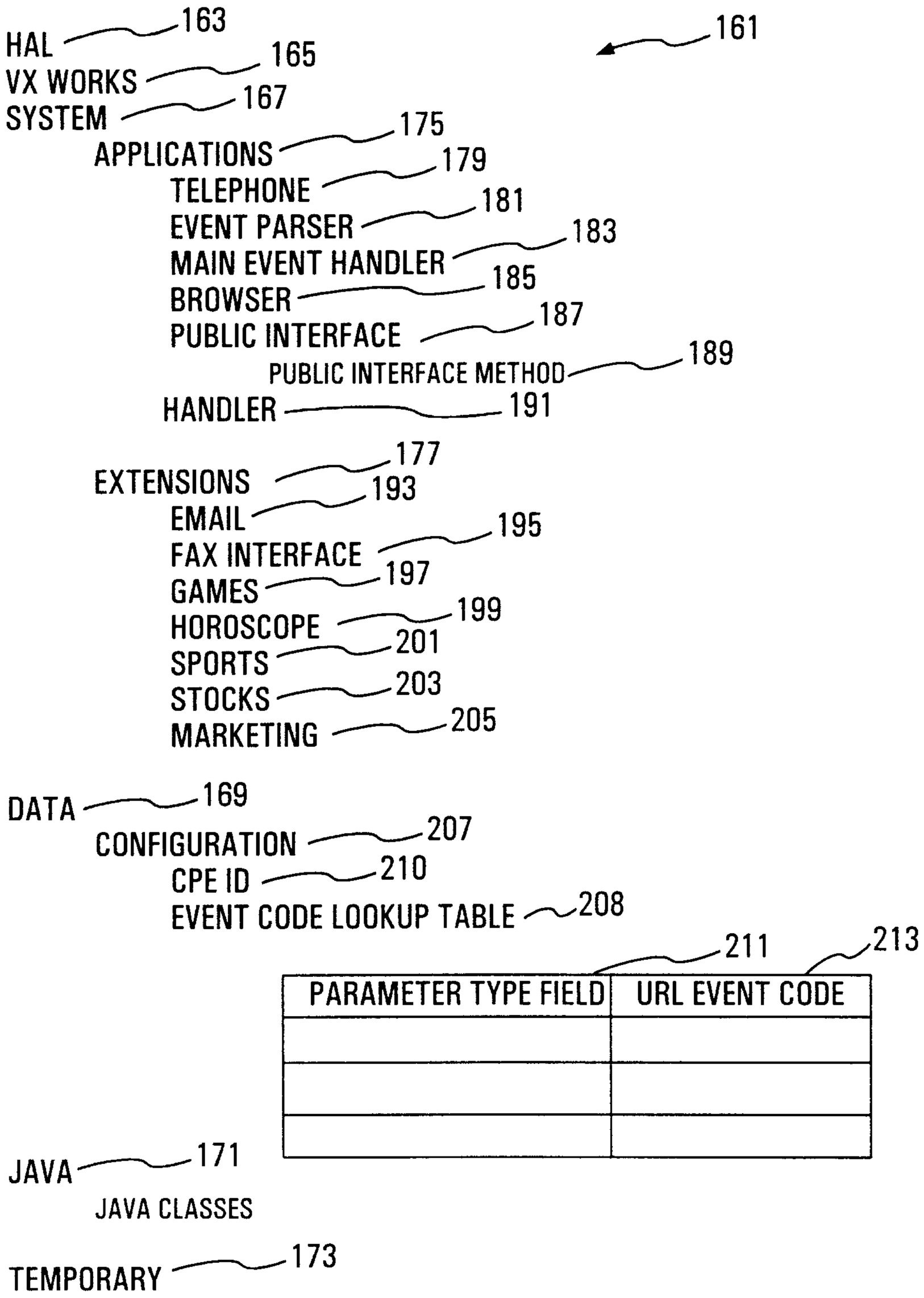

161
HAL 163
VX WORKS 165
SYSTEM 167
APPLICATIONS 175
TELEPHONE 179
EVENT PARSER 181
MAIN EVENT HANDLER 183
BROWSER 185
PUBLIC INTERFACE 187
PUBLIC INTERFACE METHOD 189
HANDLER 191
EXTENSIONS 177
EMAIL 193
FAX INTERFACE 195
GAMES 197
HOROSCOPE 199
SPORTS 201
STOCKS 203
MARKETING 205
DATA 169
CONFIGURATION 207
CPE ID 210
EVENT CODE LOOKUP TABLE 208
211
213
PARAMETER TYPE FIELD
URL EVENT CODE
JAVA 171
JAVA CLASSES
TEMPORARY 173

FIG. 6

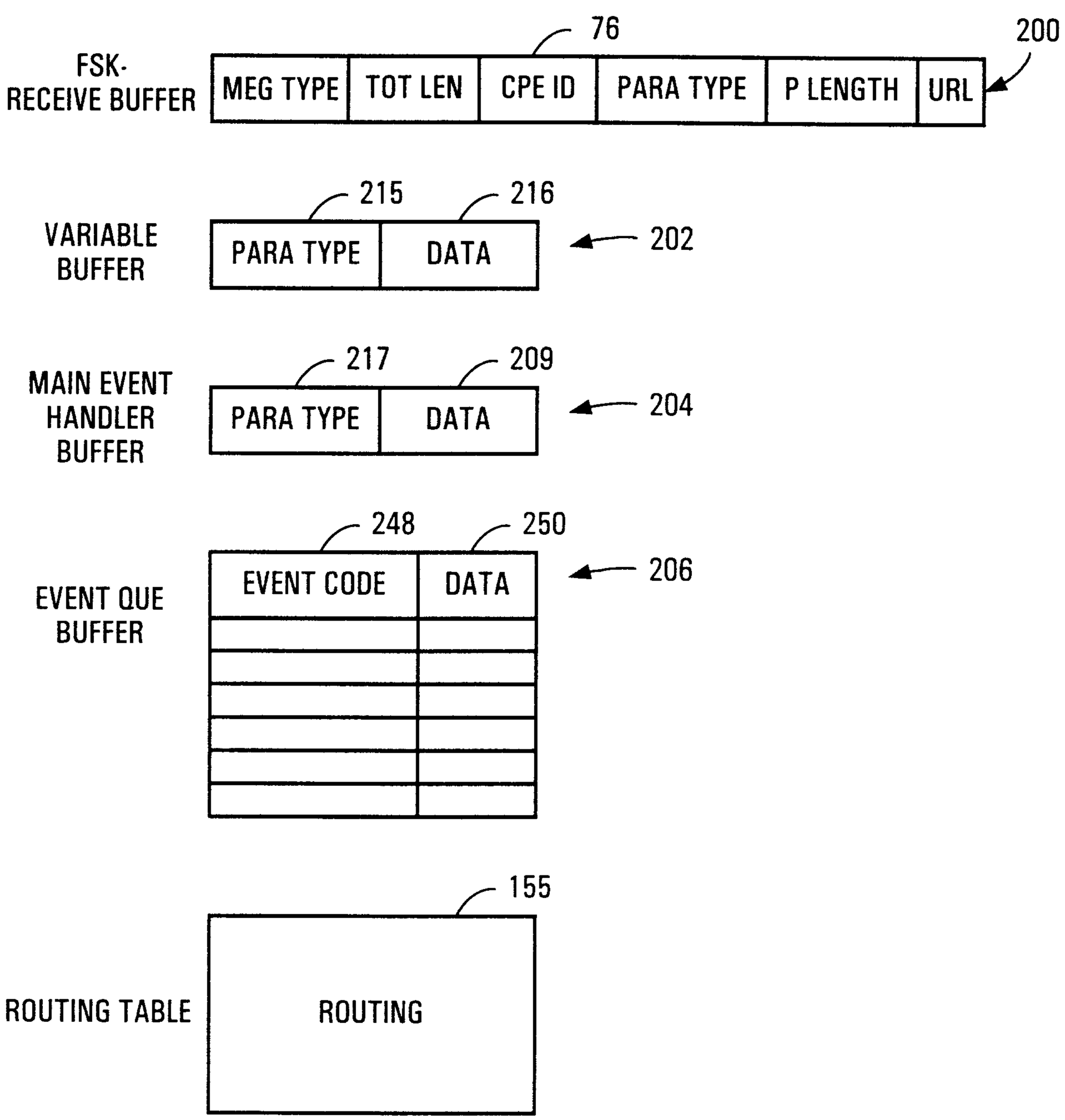
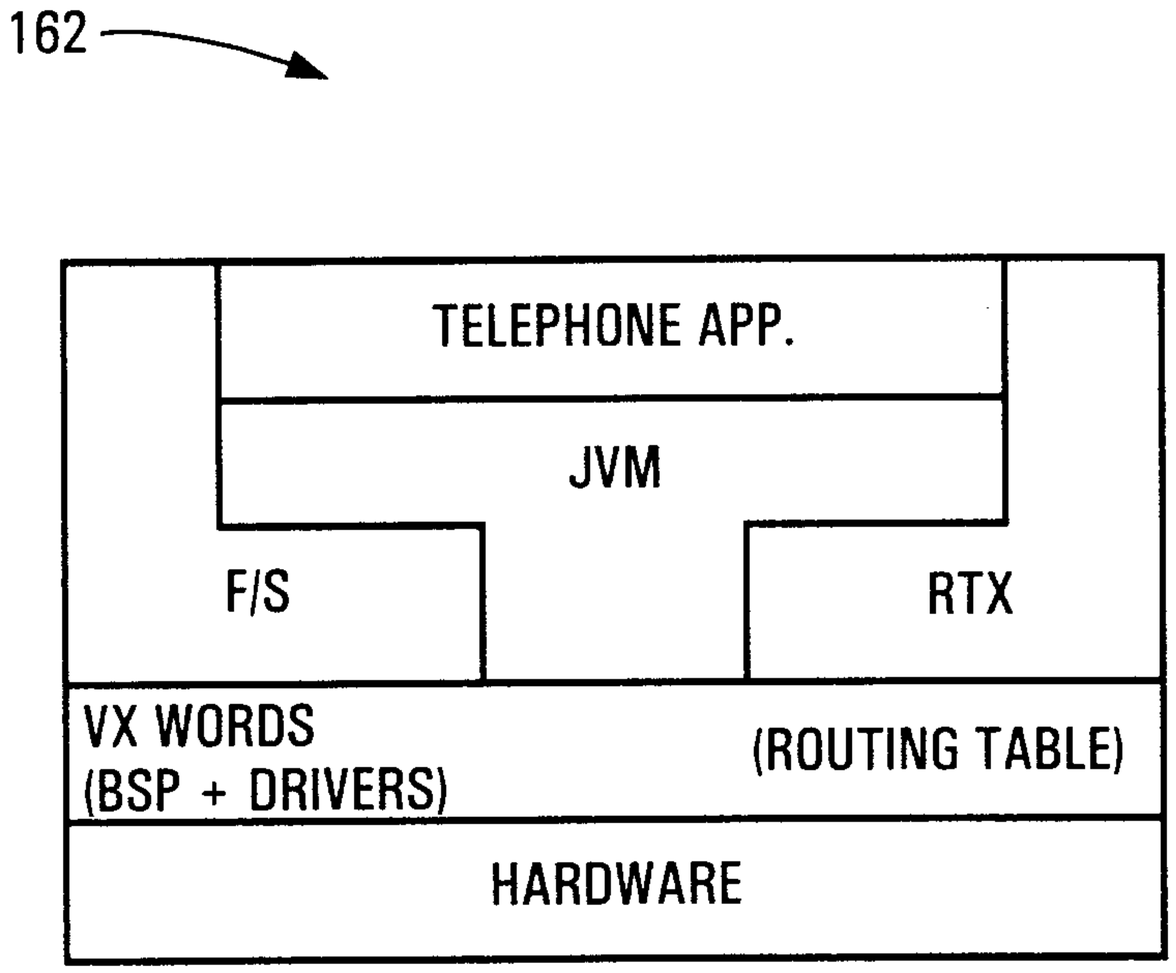

RAM
FSK-
RECEIVE BUFFER
76
200
MEG TYPE
TOT LEN
CPE ID
PARA TYPE
P LENGTH
URL
VARIABLE
BUFFER
215
216
PARA TYPE
DATA
202
MAIN EVENT
HANDLER
BUFFER
217
209
PARA TYPE
DATA
204
EVENT QUE
BUFFER
248
250
EVENT CODE
DATA
206
ROUTING TABLE
155
ROUTING

FIG. 7



F/S = FLASH FILE
JVM = JAVA VIRTUAL MACHINE
RTX = REAL TIME XSERVER

FIG. 8

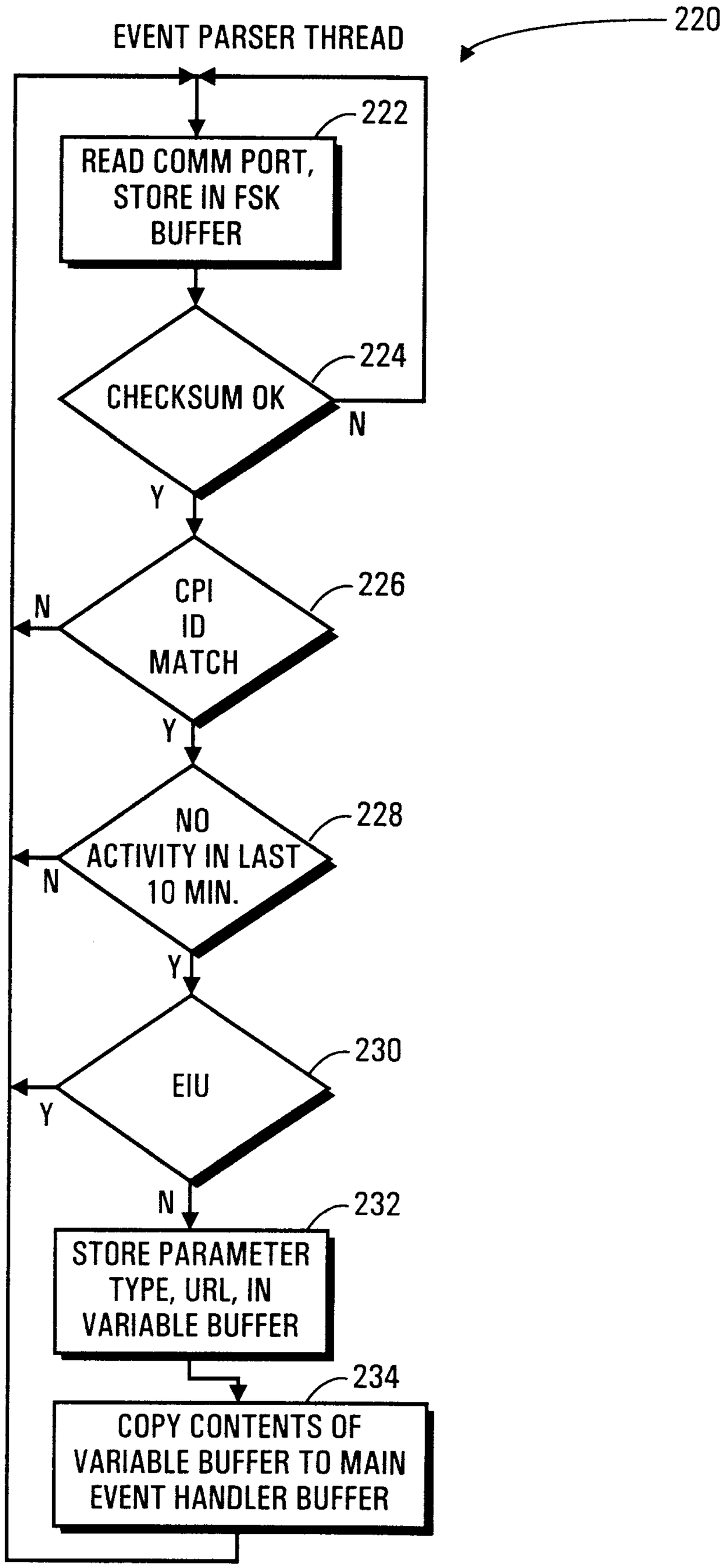

EVENT PARSER THREAD
220
222
READ COMM PORT, STORE IN FSK BUFFER
224
CHECKSUM OK
N
Y
226
CPI ID MATCH
N
Y
228
NO ACTIVITY IN LAST 10 MIN.
N
Y
230
EIU
Y
N
232
STORE PARAMETER TYPE, URL, IN VARIABLE BUFFER
234
COPY CONTENTS OF VARIABLE BUFFER TO MAIN EVENT HANDLER BUFFER

FIG. 9

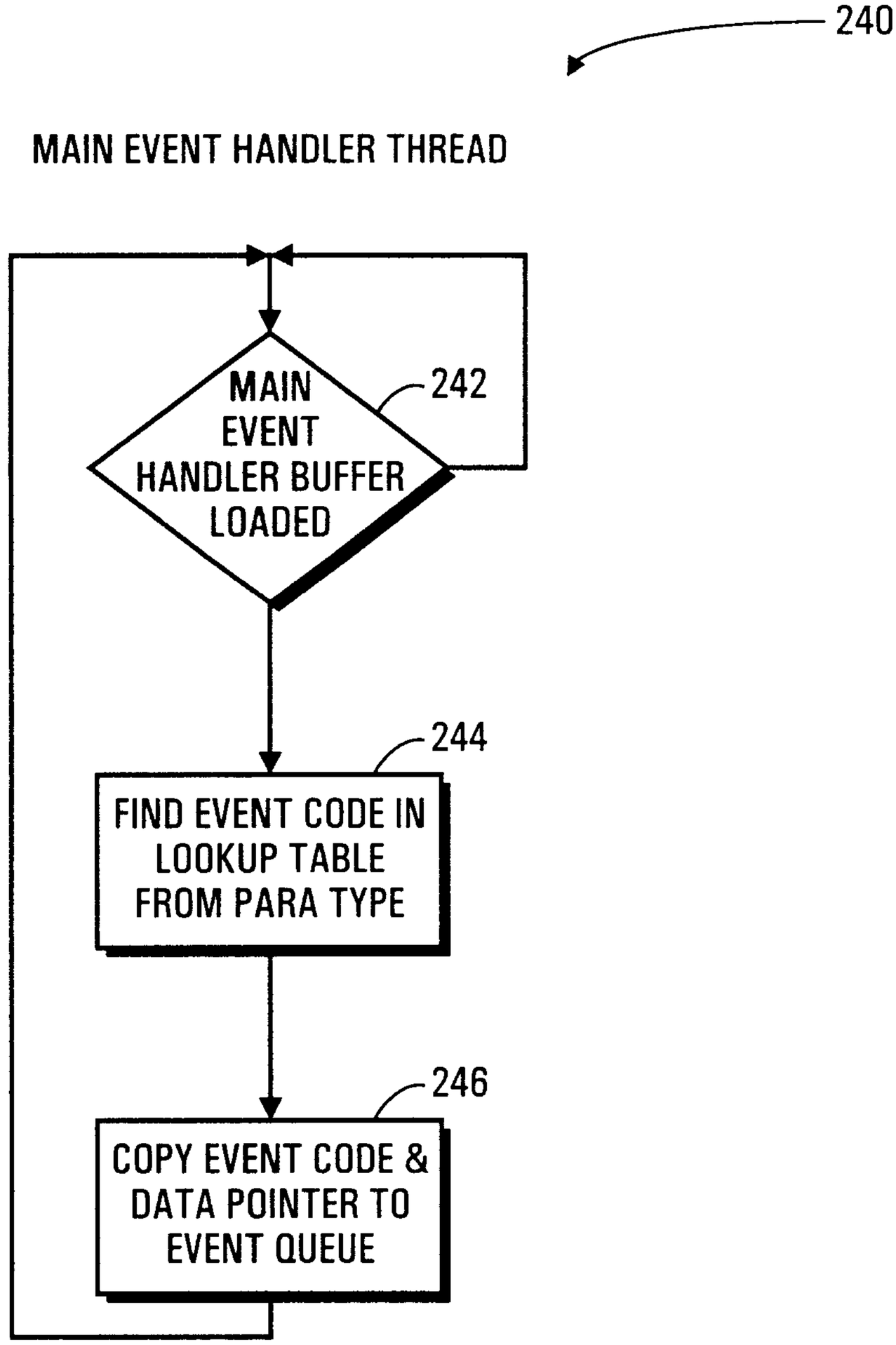

240
MAIN EVENT HANDLER THREAD
MAIN EVENT HANDLER BUFFER LOADED
242
FIND EVENT CODE IN LOOKUP TABLE FROM PARA TYPE
244
COPY EVENT CODE & DATA POINTER TO EVENT QUEUE
246

FIG. 10

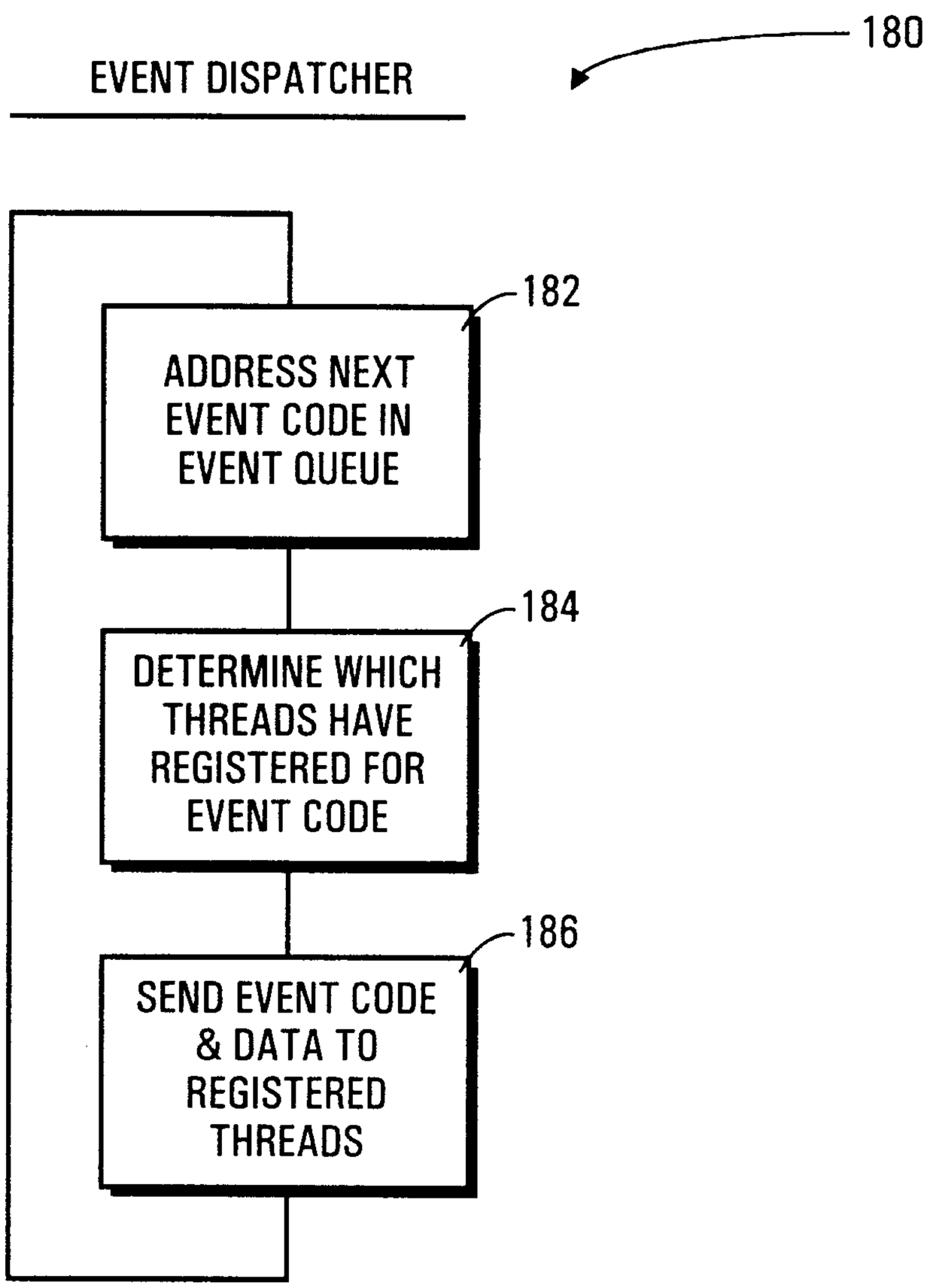

EVENT DISPATCHER
180
ADDRESS NEXT EVENT CODE IN EVENT QUEUE
182
DETERMINE WHICH THREADS HAVE REGISTERED FOR EVENT CODE
184
SEND EVENT CODE & DATA TO REGISTERED THREADS
186

FIG. 11

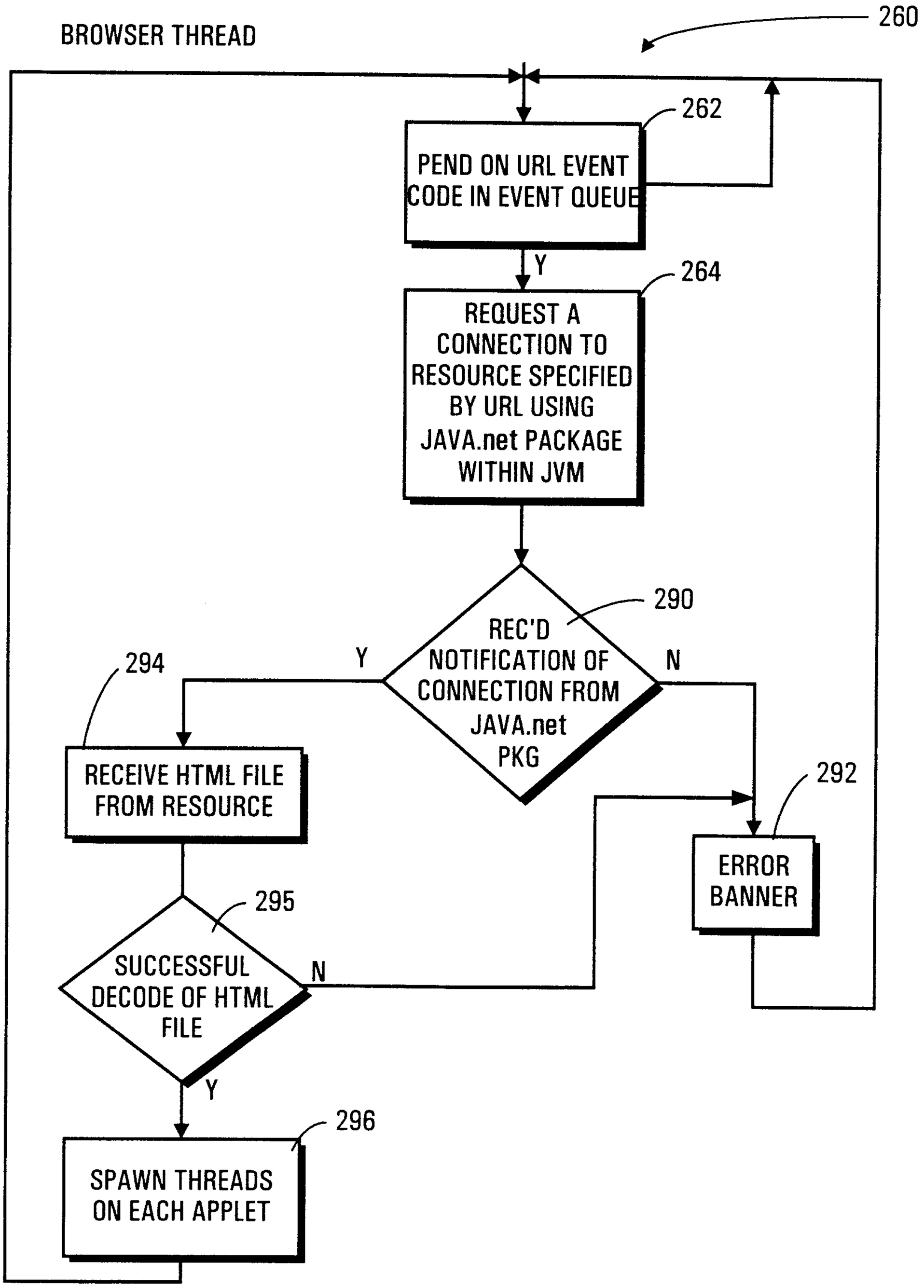

BROWSER THREAD
260
262
PEND ON URL EVENT CODE IN EVENT QUEUE
Y
264
REQUEST A CONNECTION TO RESOURCE SPECIFIED BY URL USING JAVA.net PACKAGE WITHIN JVM
290
REC'D NOTIFICATION OF CONNECTION FROM JAVA.net PKG
Y
N
294
RECEIVE HTML FILE FROM RESOURCE
292
ERROR BANNER
295
SUCCESSFUL DECODE OF HTML FILE
N
Y
296
SPAWN THREADS ON EACH APPLET

FIG. 12

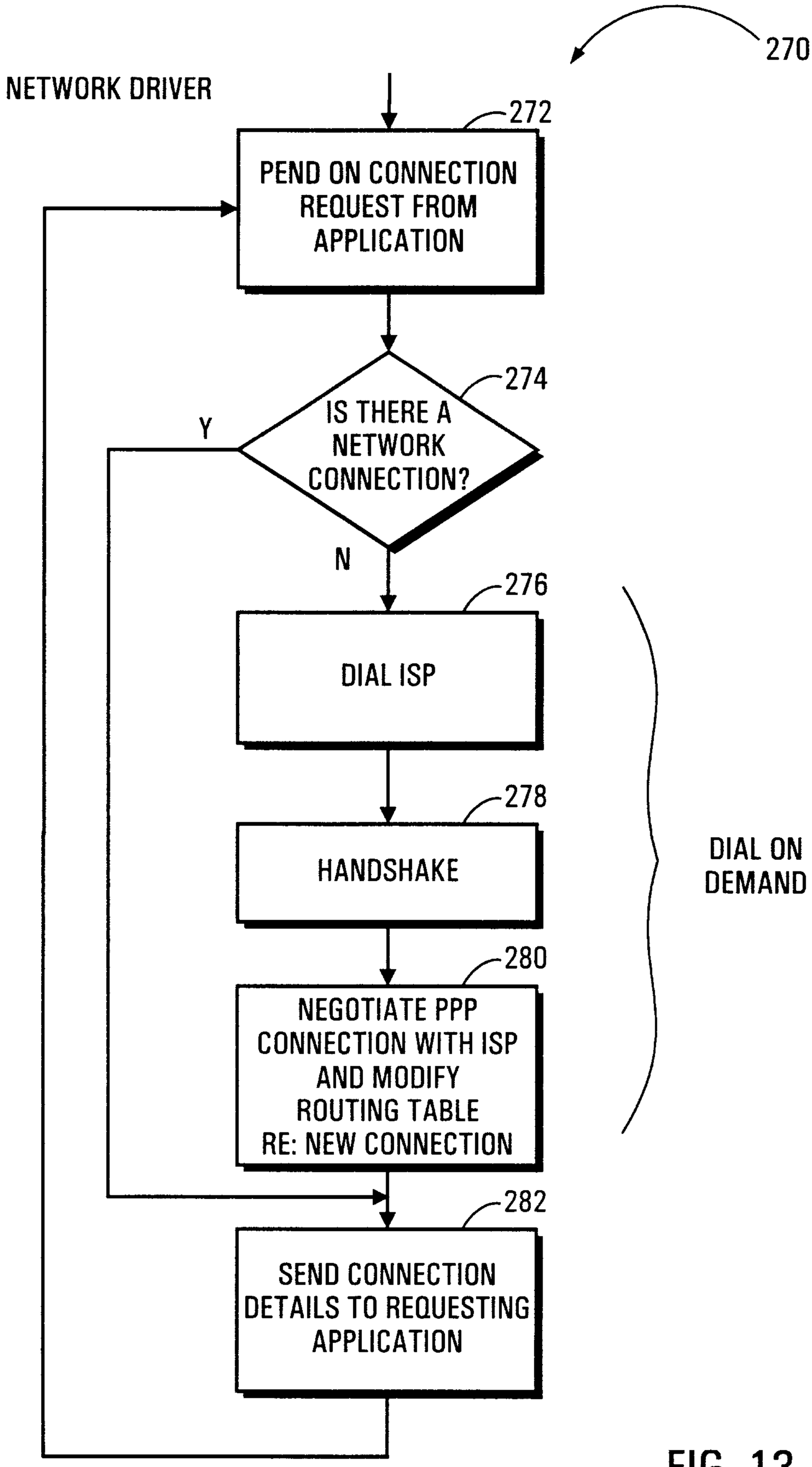

270
NETWORK DRIVER
272
PEND ON CONNECTION REQUEST FROM APPLICATION
274
IS THERE A NETWORK CONNECTION?
Y
N
276
DIAL ISP
278
HANDSHAKE
280
NEGOTIATE PPP CONNECTION WITH ISP AND MODIFY ROUTING TABLE RE: NEW CONNECTION
282
SEND CONNECTION DETAILS TO REQUESTING APPLICATION
DIAL ON DEMAND

FIG. 13

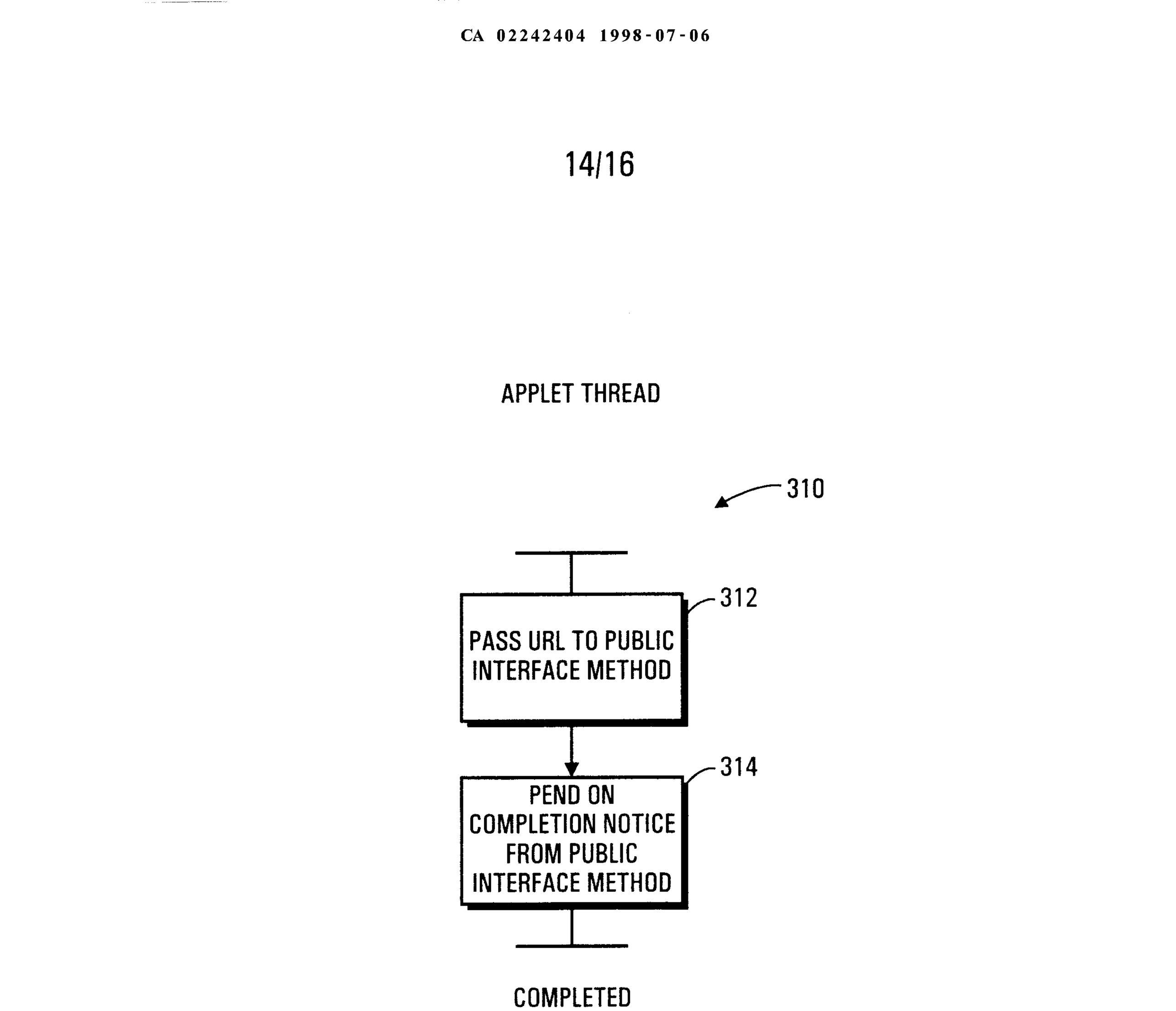

APPLET THREAD
310
312
PASS URL TO PUBLIC INTERFACE METHOD
314
PEND ON COMPLETION NOTICE FROM PUBLIC INTERFACE METHOD
COMPLETED

FIG. 14

PUBLIC INTERFACE METHOD

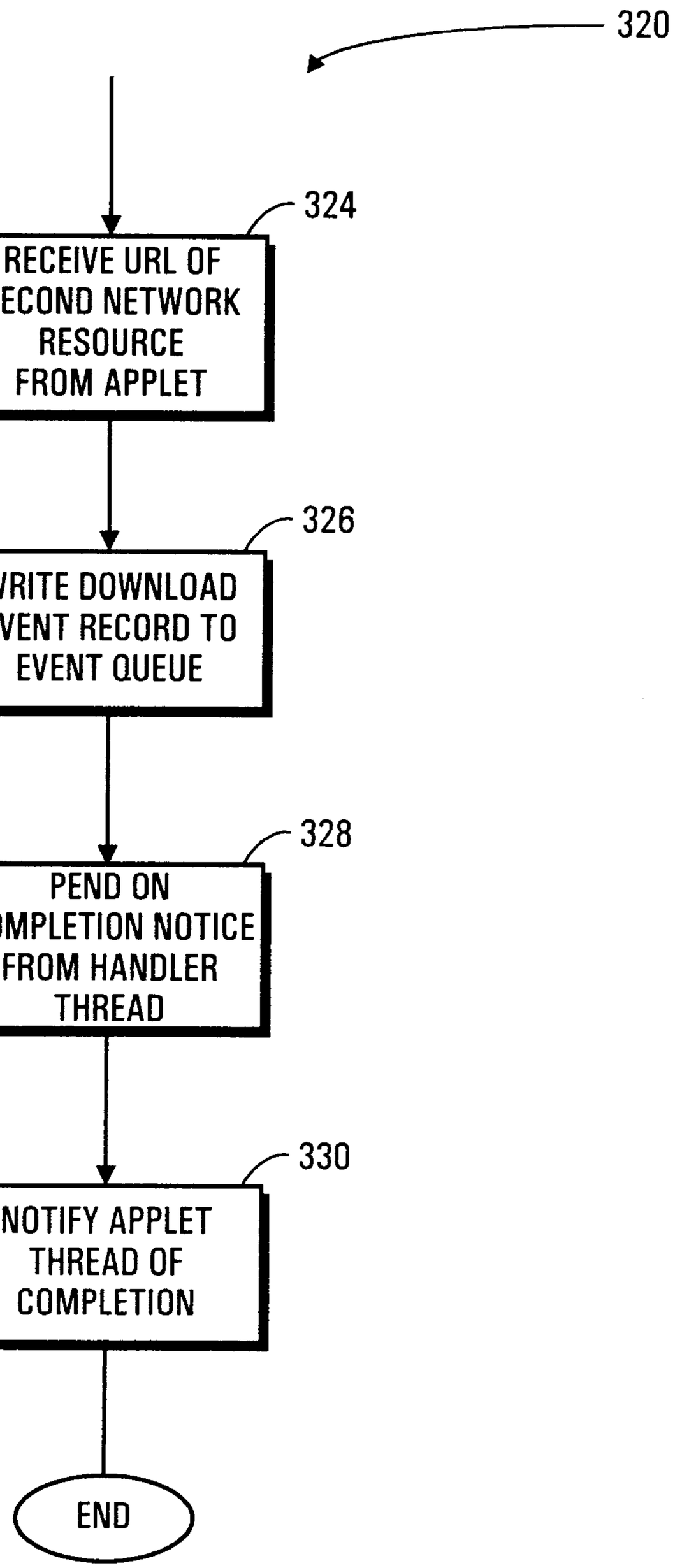


FIG. 15

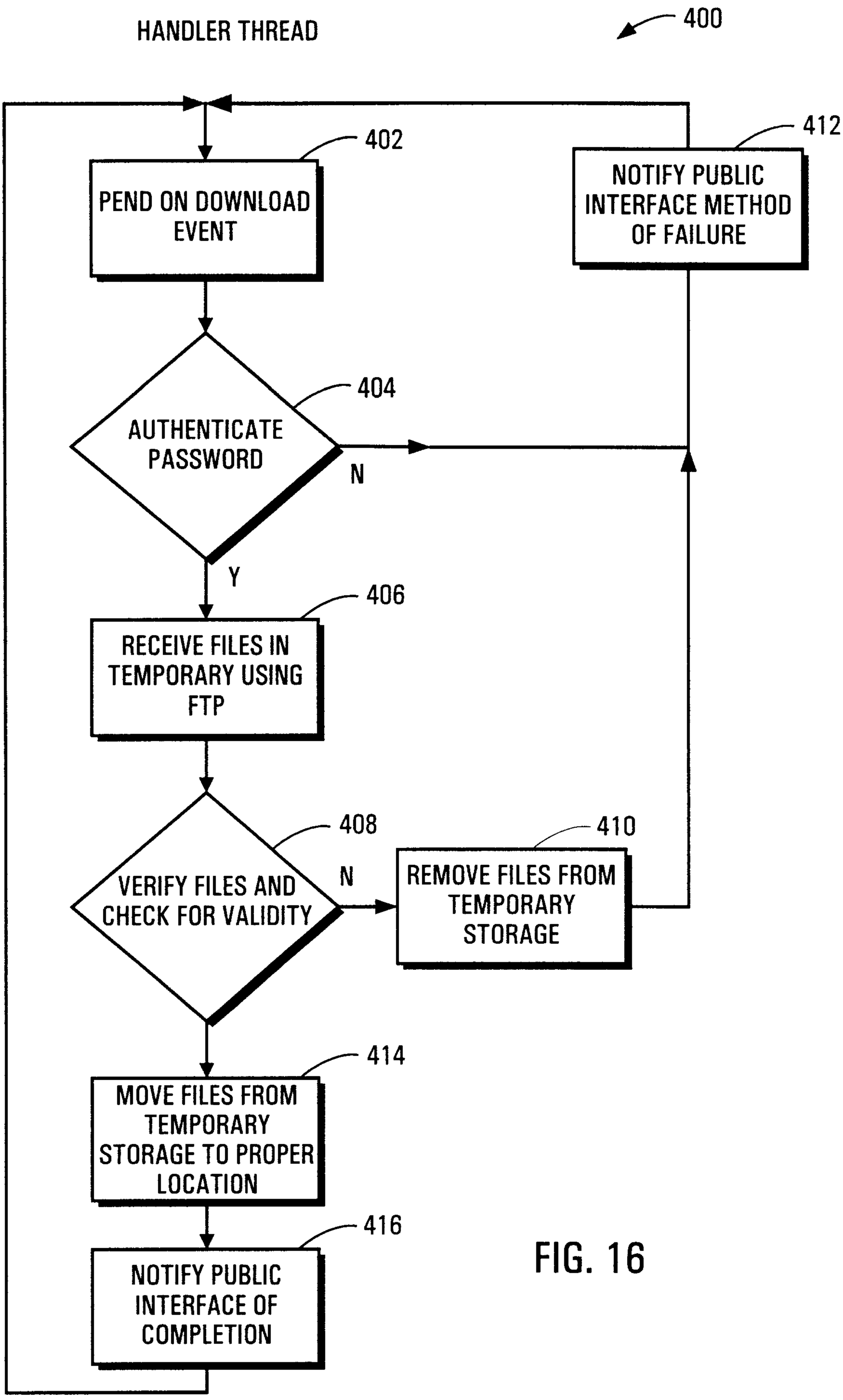

HANDLER THREAD
400
402
PEND ON DOWNLOAD EVENT
404
AUTHENTICATE PASSWORD
N
Y
406
RECEIVE FILES IN TEMPORARY USING FTP
408
VERIFY FILES AND CHECK FOR VALIDITY
N
410
REMOVE FILES FROM TEMPORARY STORAGE
412
NOTIFY PUBLIC INTERFACE METHOD OF FAILURE
414
MOVE FILES FROM TEMPORARY STORAGE TO PROPER LOCATION
416
NOTIFY PUBLIC INTERFACE OF COMPLETION

FIG. 16