



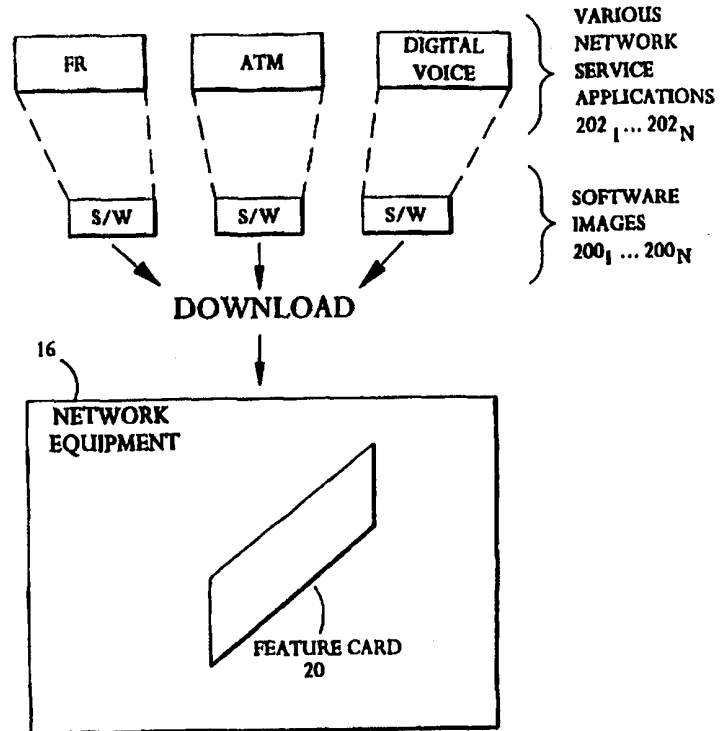
INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

<p>(51) International Patent Classification ⁷ : H04L 12/00</p>	<p>A2</p>	<p>(11) International Publication Number: WO 00/56012 (43) International Publication Date: 21 September 2000 (21.09.00)</p>
<p>(21) International Application Number: PCT/US00/06982 (22) International Filing Date: 15 March 2000 (15.03.00) (30) Priority Data: 09/270,297 15 March 1999 (15.03.99) US (71) Applicant: CISCO TECHNOLOGY, INC. [US/US]; 170 West Tasman Drive, San Jose, CA 95134 (US). (72) Inventors: DEO, Satish, D.; 4221 Norwalk Drive #BB307, San Jose, CA 95129 (US). PATEL, Pankaj, S.; 10915 Miramonte Road, Cupertino, CA 95014 (US). SATHE, Shirish, K.; 13975 Trinity Avenue, Saratoga, CA 95070 (US). (74) Agents: VINCENT, Lester, J. et al.; Blakely, Sokoloff, Taylor & Zafman, 7th floor, 12400 Wilshire Boulevard, Los Angeles, CA 90025-1026 (US).</p>		<p>(81) Designated States: AE, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BY, CA, CH, CN, CR, CU, CZ, DE, DK, DM, DZ, EE, ES, FI, GB, GD, GE, GH, GM, HR, HU, ID, IL, IN, IS, JP, KE, KG, KP, KR, KZ, LC, LK, LR, LS, LT, LU, LV, MA, MD, MG, MK, MN, MW, MX, NO, NZ, PL, PT, RO, RU, SD, SE, SG, SI, SK, SL, TJ, TM, TR, TT, TZ, UA, UG, UZ, VN, YU, ZA, ZW, ARIPO patent (GH, GM, KE, LS, MW, SD, SL, SZ, TZ, UG, ZW), Eurasian patent (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European patent (AT, BE, CH, CY, DE, DK, ES, FI, FR, GB, GR, IE, IT, LU, MC, NL, PT, SE), OAPI patent (BF, BJ, CF, CG, CI, CM, GA, GN, GW, ML, MR, NE, SN, TD, TG).</p> <p>Published <i>Without international search report and to be republished upon receipt of that report.</i></p>

(54) Title: A MULTI-SERVICE ARCHITECTURE WITH ANY PORT ANY SERVICE (APAS) HARDWARE PLATFORM

(57) Abstract

An apparatus for a multi-service network architecture for processing network traffic arriving on a network connection is disclosed. The multi-service network architecture has a plurality of network connection components residing on a single platform and a processor coupled to the network connection components. The processor is configured to execute a predetermined one of a plurality of software images corresponding to the type of network traffic arriving on the network connection and to selectively enable at least one of the network connection components according to the predetermined software image.



FOR THE PURPOSES OF INFORMATION ONLY

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

AL	Albania	ES	Spain	LS	Lesotho	SI	Slovenia
AM	Armenia	FI	Finland	LT	Lithuania	SK	Slovakia
AT	Austria	FR	France	LU	Luxembourg	SN	Senegal
AU	Australia	GA	Gabon	LV	Latvia	SZ	Swaziland
AZ	Azerbaijan	GB	United Kingdom	MC	Monaco	TD	Chad
BA	Bosnia and Herzegovina	GE	Georgia	MD	Republic of Moldova	TG	Togo
BB	Barbados	GH	Ghana	MG	Madagascar	TJ	Tajikistan
BE	Belgium	GN	Guinea	MK	The former Yugoslav Republic of Macedonia	TM	Turkmenistan
BF	Burkina Faso	GR	Greece	ML	Mali	TR	Turkey
BG	Bulgaria	HU	Hungary	MN	Mongolia	TT	Trinidad and Tobago
BJ	Benin	IE	Ireland	MR	Mauritania	UA	Ukraine
BR	Brazil	IL	Israel	MW	Malawi	UG	Uganda
BY	Belarus	IS	Iceland	MX	Mexico	US	United States of America
CA	Canada	IT	Italy	NE	Niger	UZ	Uzbekistan
CF	Central African Republic	JP	Japan	NL	Netherlands	VN	Viet Nam
CG	Congo	KE	Kenya	NO	Norway	YU	Yugoslavia
CH	Switzerland	KG	Kyrgyzstan	NZ	New Zealand	ZW	Zimbabwe
CI	Côte d'Ivoire	KP	Democratic People's Republic of Korea	PL	Poland		
CM	Cameroon	KR	Republic of Korea	PT	Portugal		
CN	China	KZ	Kazakstan	RO	Romania		
CU	Cuba	LC	Saint Lucia	RU	Russian Federation		
CZ	Czech Republic	LI	Liechtenstein	SD	Sudan		
DE	Germany	LK	Sri Lanka	SE	Sweden		
DK	Denmark	LR	Liberia	SG	Singapore		
EE	Estonia						

A MULTI-SERVICE ARCHITECTURE WITH ANY PORT ANY SERVICE
(APAS) HARDWARE PLATFORM

Field of the Invention

The present invention is related to the field of network architecture, more specifically, the method and apparatus of the present invention is a multi-service architecture with "Any Port Any Service" (APAS) hardware platform.

Background

In general, a network switch provides a data path, or interface, between networks and is a device that allows a connection to be established as necessary and terminated when there is no longer a session to support. A typical network switch is part of a network equipment such as a modular multi-service access concentrator (MAC) implemented in a multi-slot box chassis with a number of hardware boards with digital signal processors (DSPs). Each hardware board may have a set of port connectors that is wired or connected to a Public Branch Exchange (PBX) or some network level application.

A given network may operate according to a particular networking standard protocol and a typical network switch is designed to support a network application of a single networking standard protocol. Examples of networking standard protocols that may be supported by different network switches include, but are not limited to, Frame Relay, voice, circuit emulation, T1 channelized, E1 channelized, and Asynchronous Transfer Mode (ATM).

For example, a hardware board supporting a 'Voice' application may have a network switch that is connected to T1/E1 Framers and Digital Signal Processor Modules (DSPMs) specifically to support 'Voice'

-2-

applications. A hardware board supporting an 'ATM' application may have a network switch that connects T1/E1 Framers to Serial Communication Controllers (SCCs) on processors specifically to support an 'ATM' application. Thus a hardware board with a typical network switch only supports a specific pre-defined network application.

The disadvantages of the current technology are many. For example, because a hardware board with a network switch designed for a given network application is only capable of supporting that network application, different hardware boards are required to support different network applications (e.g. Voice, ATM, Frame Relay). This multiplies the efforts in development, testing, integration and support of a network product. Additionally, current technology leads to greater inventory for a service provider because a service provider must keep in stock a sufficient number of hardware boards of different support capabilities.

BRIEF SUMMARY OF THE INVENTION

An apparatus for a multi-service network architecture for processing network traffic arriving on a network connection is disclosed. The multi-service network architecture has plurality of network connection components residing on a single platform and a processor coupled to the network connection components. The processor is configured to execute a predetermined one of a plurality of software images corresponding to the type of network traffic arriving on the network connection and to selectively enable at least one of the network connection components according to the predetermined software image.

Other features and advantages of the invention will be apparent from the detailed description and drawings provided herein.

BRIEF DESCRIPTION OF THE DRAWINGS

The present invention is illustrated by way of example and not limitation in the figures of the accompanying drawings in which like references indicated similar elements in which:

Figure 1 shows a network diagram;

Figure 2a is a block diagram of one embodiment of an APAS configuration;

Figure 2b is a software block diagram of exemplary software components interfacing with an APAS hardware platform;

Figure 3 is a block diagram of one embodiment of an APAS hardware platform;

Figure 4 is an embodiment of the TDM highway; and

Figure 5 is a flow chart showing one embodiment of an APAS set up process.

DETAILED DESCRIPTION

A multi-service architecture that supports "Any Port Any Service" (APAS) for use in multi-slot, multi-service networking products is described. As will be described in more detail below, software can be invoked to support the desired service type (HDLC/FR/ATM, etc.) while using the common hardware. The software can configure any port for any of the desired service types. In this way, different service types can be supported simultaneously on different ports. Additionally, making use of the common hardware platform simplifies the configuration of the various network interface cards.

By downloading an appropriate software image, the common hardware can be configured to support various services including but not limited to T1/E1 voice application, channelized FR application, unchannelized FR application, T1/E1 ATM application, serial

-4-

Asynchronous/Bisynchronous mode applications or any combination across the ports, depending on the application and the corresponding software images.

The intended advantages include helping to save in hardware development, testing and qualification time, and helping to reduce costs associated with the training of the "service provider" line of customers, field/support and sales personnel. Further intended advantages include helping to reduce the risks associated with different service/interface types, helping to reduce time-to-market due to savings in efforts at various stages of hardware development, and helping to save in the overall hardware cost due to higher volumes resulting from the use of common hardware across various applications. Additional intended advantages include the benefit to a service provider by requiring only one type of hardware for different network applications and thereby decreasing inventory. This is opposed to requiring multiple types of hardware for different applications. Further, a service provider may be provided with a decreased learning curve on hardware aspects of a platform/product by providing one hardware that can be used for different applications.

Figure 1 is a diagram of one embodiment of a network. The exemplary network connects a head office 10 with at least one remote office 12 through a network carrier 14 such as a wide area network (WAN). A network equipment 16 residing at the head office 10 has several card slots for several feature cards $20_1 \dots 20_n$. Each feature card 20 has a set of port connectors that is wired or connected to a Public Branch Exchange (PBX) or some network level application. The network carrier 14 is supported by all feature cards $20_1 \dots 20_n$ and supports transmission through several different services including but not limited to

an Asynchronous Transfer Mode (ATM) transmission coming from an ATM service connection and a Frame Relay (FR) service connection.

The network equipment 16 may be implemented with the APAS multi-service architecture to configure any port on a feature card for any desired service type and to allow different service types to be supported simultaneously on different ports.

Figure 2a is a block diagram of one embodiment of an APAS configuration. The embodiment is comprised of a feature card 20 of a network equipment 16. The system is also comprised of various software images $200_1 \dots 200_N$ dependent on a set of network service applications $202_1 \dots 202_N$. For one embodiment, the applications $202_1 \dots 202_N$ may be, but are not limited to, an FR application, an ATM application or a digital voice application.

The illustrated system may have one network or multiple networks, and the various components of the network or networks may be deploying an FR service, an ATM service, or a digital voice service. Within the service, there are different types of transmissions including channelized and unchannelized transmissions.

The network equipment 16 may be a modular multi-service access concentrator (MAC) implemented in a multi-slot box chassis with a number of digital signal processors (DSP) and a feature card 20 implemented with an APAS hardware platform architecture.

In order to support different applications $202_1 \dots 202_N$ such as digital voice, ATM, FR or serial applications, the same feature card 20 may be used or different feature cards $20_1 \dots 20_N$ may be plugged into different slots and a corresponding software image 200 may be downloaded onto the respective feature cards $20_1 \dots 20_N$ for a particular application 202. For example, if a card in a slot three is designated to

support ATM service, then the software image 200 for ATM service is downloaded onto card three. To process various different applications 202₁...202_N, a corresponding software image 200 is downloaded.

Alternatively, two ports may support a different type of application 202 within the same feature card 20. For this scenario, two different types of software images 200₁ and 200₂ may be run on a processor that supports two different applications 202₁ and 202₂.

The software images 200₁...200_N therefore program the control logic of a feature card 20 differently depending on the network service type to be supported.

Figure 2b is a software block diagram of exemplary software components interfacing with an APAS hardware platform. Application layer software 202 such as Voice over Internet Protocol (IP) or Voice over Asynchronous Transfer Mode (ATM) application software is coupled to network management software 220. The network management software 220 is coupled to various device drivers 230. The device drivers 230 are in turn coupled to firmware 240 which is referred herein as software images 200₁...200_N. The boot and startup software 250 perform initialization of the hardware 260 which is the network equipment 16.

Figure 3 is a block diagram of one embodiment of an APAS hardware platform. One embodiment may be implemented with either a single board housing the processors, T1/E1 framers, TDM switch and all related logic and the proper number of DSPs. In the alternative, the proper number of DSPs may be mounted with the associated logic on pluggable modules, while the remaining logic is housed on the base-board.

For one embodiment, the feature card 20 is composed of a base-board 301 and Digital Signal Processor Modules (DSPMs) 302₁...302_N. The base-board 301 houses a first processor 316 and a second processor 324 with their respective associated logic while each DSPM 302 houses a predetermined number (such as up to six) of DSPs with their associated logic. The base-board 301 may support up to four T1/E1 ports 300₁...300₄ for a total of 96/120 channels.

The base-board 301 is implemented with a dual processor such as the MPC860 from Motorola, Inc. of Schaumburg, Illinois. It may be noted that the MPC860 is an example and other suitable devices may be used to implement the common hardware.

Additionally, the choice of processors, the number of physical ports, the number of DSPs, and other devices on the base-board 301 are dependent on the required or desired networking applications 202₁...202_N and performance of the platform. The illustrated exemplary embodiment is one such case supporting a "common hardware" for specific set of requirements.

The first processor 316 may be configured to run the system's boot code and other firmware/software including one or more software image 200. At power-on or on hard reset, the first processor 316 downloads the boot code from boot flash memory 328 to a first local memory 318 such as an SDRAM and executes the boot code from the first local memory 318. The first processor 316 then initializes various on-board resources, such as the T1/E1 port interface 320, TDM switch 322 and other input/output (I/O) devices. Additionally, the first processor 316 maintains the base-board 301 under reset until its initialization of the various peripherals, including the T1/E1 port interface 320 and the TDM switch 322, is complete. Once initialization is complete, the first

processor 316 removes the second processor 324 out of reset and allows the second processor 324 to run.

The second processor 324 is responsible for the DSP interface. After being removed out of reset by the first processor 316, the second processor 324 copies code from the first local memory 318 to its local memory 326 (a second local memory such as an SDRAM) and begins executing the code out of the second local memory 326. Further, the second processor 324 initializes the DSP's on the DSP modules $302_1 \dots 302_N$. The second processor 324 handles the data transfer to/from the DSP's on the DSP modules $302_1 \dots 302_N$ from/to the second memory 326 and handles various data-centric tasks such as protocol-specific-packetization. For one embodiment, the two processors interact with each other via interrupt-based messaging and/or software mailboxes/semaphores. In an alternate embodiment, the two processors are replaced by a single processor.

The TDM switch 322 under software control (via software images $200_1 \dots 200_N$) routes the individual timeslots (DSOs) from any TDM port to any other TDM port. This allows routing of voice connections (DSOs) from any given port to any other port on board across the T1/E1 ports $300_1 \dots 300_4$ in case of 'Voice' applications. For one embodiment, the TDM switch 322 provides sixteen TDM ports in each direction. The spare ports are wired to Serial Communication Controllers (SCCs) 345 (TDM ports 1 and 2) on both processors. The single platform hardware architecture with the TDM switch 322 allows the single platform for use for Voice/ATM/Frame-Relay and other applications with the corresponding software images $200_1 \dots 200_N$.

T1/E1 port interface 320 is comprised of T1/E1 framers and Line Interface Circuitry (LIUs) and can be integrated (i.e. T1/E1 each framer and LIU integrated as a single unit) or separate. In the ingress direction,

the T1/E1 framers perform data and clock recovery, monitor the trunk for alarms, receive serial bit stream from the physical line, perform the framing and other functions and send out the PCM bit stream consisting of individual timeslots to the TDM switch 322. The T1/E1 framers also extract the received clock from the received data. In the egress direction, the PCM bit stream from the TDM switch 322 is processed by the T1/E1 framer and sent out over the T1/E1 physical line.

The DSP modules $302_1, \dots, 302_N$ comprise DSPs with their respective memories and clocking logic. The DSPs run various software algorithms for the required functions, such as different compression-modes, DTMF Tone detection/generation, Fax tone detection and processing, etc. The DSPs receive the serial data from the TDM switch 322. Each DSP processes the data for certain timeslots and interrupts the second processor 324 at the completion of processing. Upon receiving the interrupt, the second processor 324 copies the processed data from proper DSPs to local memory (318 and 326). In the egress direction, the second processor 324 moves the data from local memory (318 and 326) to the proper DSPs and the DSPs process the data to generate serial bit streams. The serial bit streams from the various DSPMs $302_1, \dots, 302_N$ come to the TDM switch 322 and the TDM switch 322 routes the proper timeslots to the proper T1/E1 framers under software control.

The local memory (318 and 326) stores code and data for the processors including software images $200_1, \dots, 200_N$. More specifically, when a new network connection is being configured (such as a "Voice Call Set-Up" or "an ATM VC being set-up" or a "Frame Relay connection (DLCI) set-up"), a connection manager 340 identifies the "type of connection set-up" being requested. The connection manager 340 invokes the proper low-level software/firmware modules (software images

200₁...200_N) which are then downloaded into the local memory (318 and 326). Either memory is accessible by both processors. The processors execute out of their own local memory.

The backplane logic block 325 interfaces the processors with the backplane bus 327. In the ingress direction, the backplane logic block 325 receives data (packets/cells/frames depending on the application and protocols), and transmits them over the backplane to the proper destination. In the egress direction, the backplane logic block 325 receives the data from the backplane (packets/cells/frames) and transmits the data to the processors.

The DSPMs 302₁...302_N DSPs interface to TDM streams on one interface and to the carrier host router on the other. In the ingress direction, the DSPs take PCM traffic and form Internet Protocol (IP) packets, which are then read by the first processor 316 over an HPI bus 350 in the first local memory 318 and processed further according to the configuration of the corresponding software image 200. In the egress direction, the first processor 316 sends the IP packets to the DSPs on the DSPMs 302₁...302₄ over the HPI bus 350. The DSPs then convert the data to PCM bit streams, which are then routed to the proper T1/E1 framers by the TDM switch 322.

Figure 4 is an embodiment of the TDM highway. The APAS hardware platform is implemented with a TDM highway that is connected such that the same hardware can be re-used to support a variety of different network applications (e.g. ATM, FR, voice applications). More specifically, the TDM highway connects the T1/E1 Framers to the TDM for two way communication. Further, the ports on the TDM switch 322 are wired to the DSPMs 302₁...302_N and to the TDM ports on the SCCs 345 (SCC1, 2) on the processors.

When a new network connection is being configured, a connection manager 340 identifies the type of connection set-up being requested. The connection manager 340 invokes the proper low-level software/firmware modules (software images 200₁...200_N) which are then downloaded into the local memory (318 and 326). The processors (316 and 324) execute the code associated with the software image 200 which in turn programs the TDM switch 322 to correctly manage the desired connectivity (i.e. TDM Switch to DSPMs in case of Voice applications, TDM Switch to SCCs ports in case of FR/ATM applications). The SCC 345 would be connected Non-Muxed Serial Interface (NMSI) ports or TDM ports depending on whether the application needs Unchannelized mode or Channelized mode support, etc.

For example, in the ingress direction, the PCM stream from the T1/E1 framers 400₁...400_N comes to the TDM switch 322 and the individual timeslots (or "channels") on the PCM streams are routed by the TDM switch 322 per the software image 200 in the programmed manner to appropriate DSPM's 302₁...302_N.

In the egress direction, the serial bit streams from the various DSPM's 302₁...302_N come to the TDM switch 322 and the TDM switch 322 routes the proper channels to the proper T1/E1 framers 400₁...400₄ as programmed.

In this way, the traffic always passes through the TDM switch 322, and the TDM switch 322 routes the traffic to the appropriate on-board component as programmed by the software image 200 corresponding to the traffic type to be processed. Thus, depending on the application 202, the TDM switch 322 can be properly programmed by a software image 200 to send the data streams for individual (or complete T1/E1 frame) to any port according to the software image 200 being executed (e.g. the ports for the DSPMs 302₁...302_N or the SCCs 345).

With the above described hardware scheme, some timeslots in a single T1/E1 frame may carry 'Voice' while some others may carry 'data' and the timeslots are routed properly through the TDM switch 322 by programming the TDM switch 322 (e.g. voice-channels routed to the DSPM/s and data-channels to the SCCs) simultaneously, without the need for different types of hardware. Thus, the hardware muxing logic (via the control logic on-board and the TDM switch 322) and the TDM highway's connectivity makes the hardware architecture reusable for a variety of applications, thereby providing significant advantages.

Figure 5 is a flow chart showing one embodiment of an APAS set up process. In step 501, whenever a new connection is being set-up, a higher layer software module such as a connection manager 340 detects the type of connection set-up being requested. In step 502, the higher layer software module then invokes the proper lower level software/firmware modules such as a software image 200. For example, in order to support multiple types of applications, different software images $200_1, \dots, 200_N$ are made available in the system such as in a common repository, boot flash memory, PCMCIA disk, or downloadable over the Internet.

In step 503, the desired software image 200 is downloaded to the on-board local memory. In step 504, the code associated with the software image 200 is then executed from the local memory and traffic is processed by the processors according to its corresponding software image 200. For example, the TDM switch 322 routes traffic according traffic type as programmed by the corresponding software image 200 (e.g. TDM Switch to DSPMs in case of Voice applications, TDM Switch to SCCs ports in case of FR/ATM applications). Once a particular type of software image 200 is resident in the local memory (318 and 326), for subsequent connection-setup of similar type (Voice/ATM/FR, etc.) the

code execution simply continues from the local memory (318 and 326) directly.

What has been described is a method and apparatus for a multi-service architecture with "Any Port Any Service" (APAS) hardware platform. A single piece of hardware is capable of supporting various different software or individual software images related to a particular application. The software image simply needs to be downloaded and is processed by the hardware common across various applications.

In the foregoing specification, the invention has been described with reference to specific exemplary embodiments thereof. It will, however, be evident that various modifications and changes may be made thereto without departing from the broader spirit and scope of the invention as set forth in the appended claims. The specification and drawings are, accordingly, to be regarded in an illustrative rather than a restrictive sense.

CLAIMS

What is claimed:

1. A method for a single hardware platform to support multiple network traffic types, comprising:
 - detecting a request to establish a network connection to the hardware platform;
 - determining network traffic type used by the network connection;
 - and
 - executing code to selectively enable connection components to process data over the network connection, according to the network traffic type.
2. The method of claim 1 further comprising invoking an appropriate one of a plurality of software images corresponding to the network traffic type.
3. The method of claim 2 further comprising copying the appropriate one of a plurality of software images into a local memory on the single platform.
4. The method of claim 2 wherein one of the plurality of network traffic type being voice data.
5. The method of claim 2 wherein one of the plurality of network traffic type being Asynchronous Transfer Mode (ATM).
6. The method of claim 2 wherein one of the plurality of network traffic type being Frame Relay.

-15-

7. An apparatus for a multi-service network architecture for processing network traffic arriving on a network connection comprising:
a plurality of network connection components residing on a single platform; and
a processor coupled to the plurality of network connection components and configured to execute a predetermined one of a plurality of software images corresponding to the type of network traffic arriving on the network connection and to selectively enable at least one of the plurality of network connection components according to the predetermined one of a plurality of software images.
8. The apparatus of claim 7 further comprising a local memory coupled to the processor and configured to hold the predetermined one of a plurality of software images.
9. The apparatus of claim 8 wherein at least one the plurality of network connection components is a Time Division Multiplexed (TDM) switch configured to provide full-duplex serial paths.
10. The apparatus of claim 9 wherein the plurality of network connection components comprises a plurality of T1/E1 framers coupled to a first set of plurality of ports on the TDM switch.
11. The apparatus of claim 10 further comprising a plurality of digital signal processing modules coupled to a second set of a plurality of ports on the TDM switch.

-16-

12. The apparatus of claim 10 further comprising a plurality of serial communication controllers coupled to a third set of a plurality of ports on the TDM switch.

13. The apparatus of claim 11 further comprising a connection management software coupled to the local memory and configured to identify the type of connection set-up being requested and to invoke a corresponding one of a plurality of software images which programs the TDM switch to correctly manage desired connectivity.

14. A system for a multi-service network architecture for processing network traffic arriving on a network connection comprising:
a plurality of network connection components residing on a single platform; and
a processor coupled to the plurality of network connection components and configured to execute a predetermined one of a plurality of software images corresponding to the type of network traffic arriving on the network connection and to selectively enable at least one of the plurality of network connection components according to the predetermined one of a plurality of software images.

15. The system of claim 14 further comprising a local memory coupled to the processor and configured to hold the predetermined one of a plurality of software images.

16. The system of claim 15 wherein at least one the plurality of network connection components is a Time Division Multiplexed (TDM) switch configured to provide full-duplex serial paths.

-17-

17. The system of claim 16 wherein the plurality of network connection components comprises a plurality of T1/E1 framers coupled a first set of plurality of ports on the TDM switch.

18. The system of claim 17 further comprising a plurality of digital signal processing modules coupled to a second set of a plurality of ports on the TDM switch.

19. The system of claim 18 further comprising a plurality of serial communication controllers coupled to a third set of a plurality of ports on the TDM switch.

20. The system of claim 19 further comprising a connection management software coupled to the local memory and configured to identify the type of connection set-up being requested and to invoke a corresponding one of a plurality of software images which programs the TDM switch to correctly manage desired connectivity.

21. An apparatus for a multi-service network architecture for processing network traffic arriving on a network connection comprising:
a plurality of means for processing data for a predetermined network traffic type residing on a single platform; and
means for executing code for a predetermined one of a plurality of software images corresponding to the type of network traffic arriving on the network connection and to selectively enable at least one of the plurality of means for processing data according to the predetermined one of a plurality of software images, the means for executing coupled to the plurality of means for processing.

-18-

22. The apparatus of claim 20 further comprising means for storing the predetermined one of a plurality of software images, the means for storing coupled to the means for executing.

23. The apparatus of claim 22 wherein at least one the plurality of means for processing is a Time Division Multiplexed (TDM) switch configured to provide full-duplex serial paths.

24. The apparatus of claim 23 wherein the plurality of means for processing comprises a plurality of T1/E1 framers coupled to a first set of plurality of ports on the TDM switch.

25. The apparatus of claim 24 further comprising a plurality of digital signal processing modules coupled to a second set of a plurality of ports on the TDM switch.

26. The apparatus of claim 25 further comprising a plurality of serial communication controllers coupled to a third set of a plurality of ports on the TDM switch.

27. The apparatus of claim 26 further comprising means for identifying the type of connection set-up being requested at the network connection and to invoke a corresponding one of a plurality of software images which programs the TDM switch to correctly manage desired connectivity, the means for identifying coupled to the means for storing.

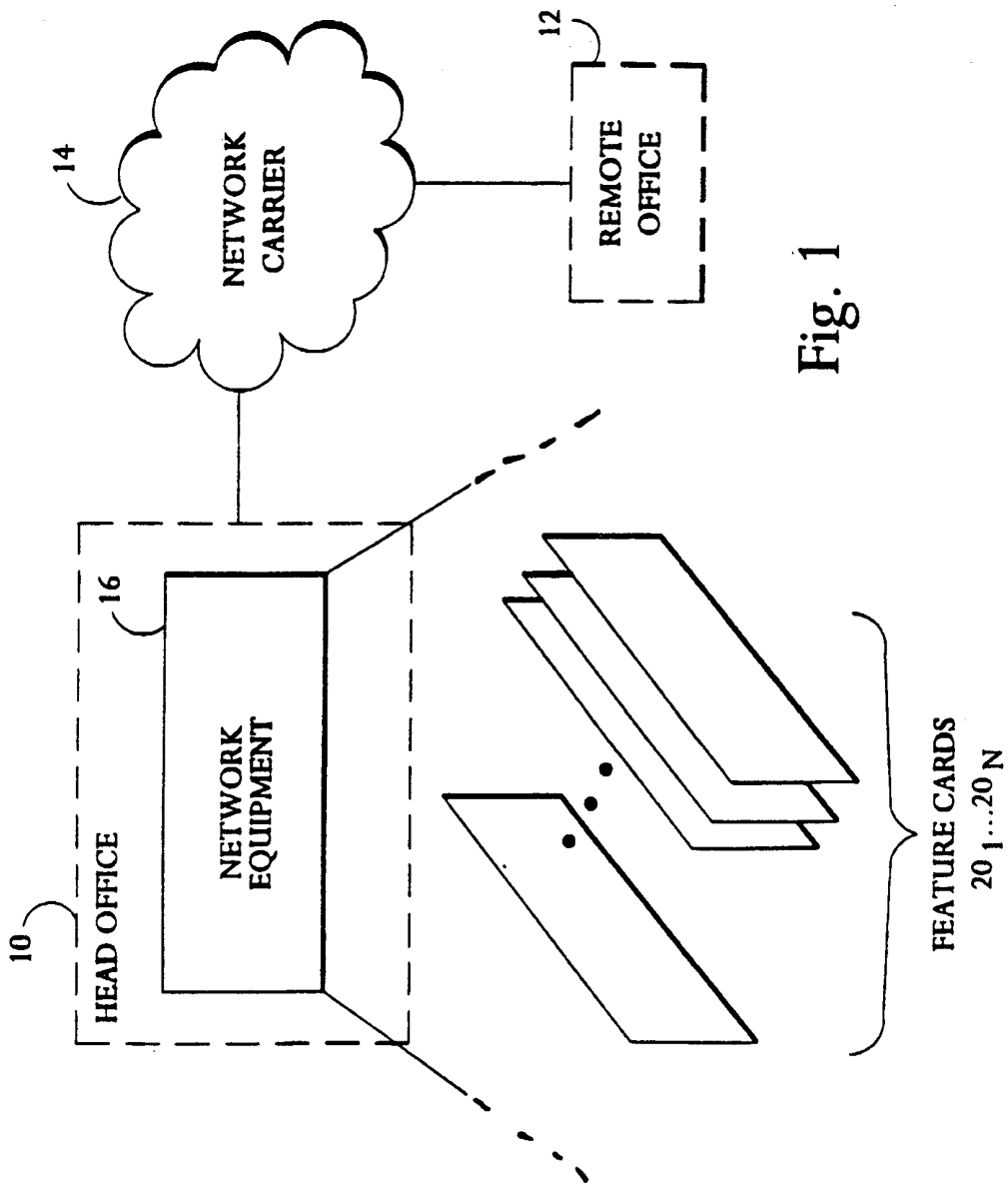


Fig. 1

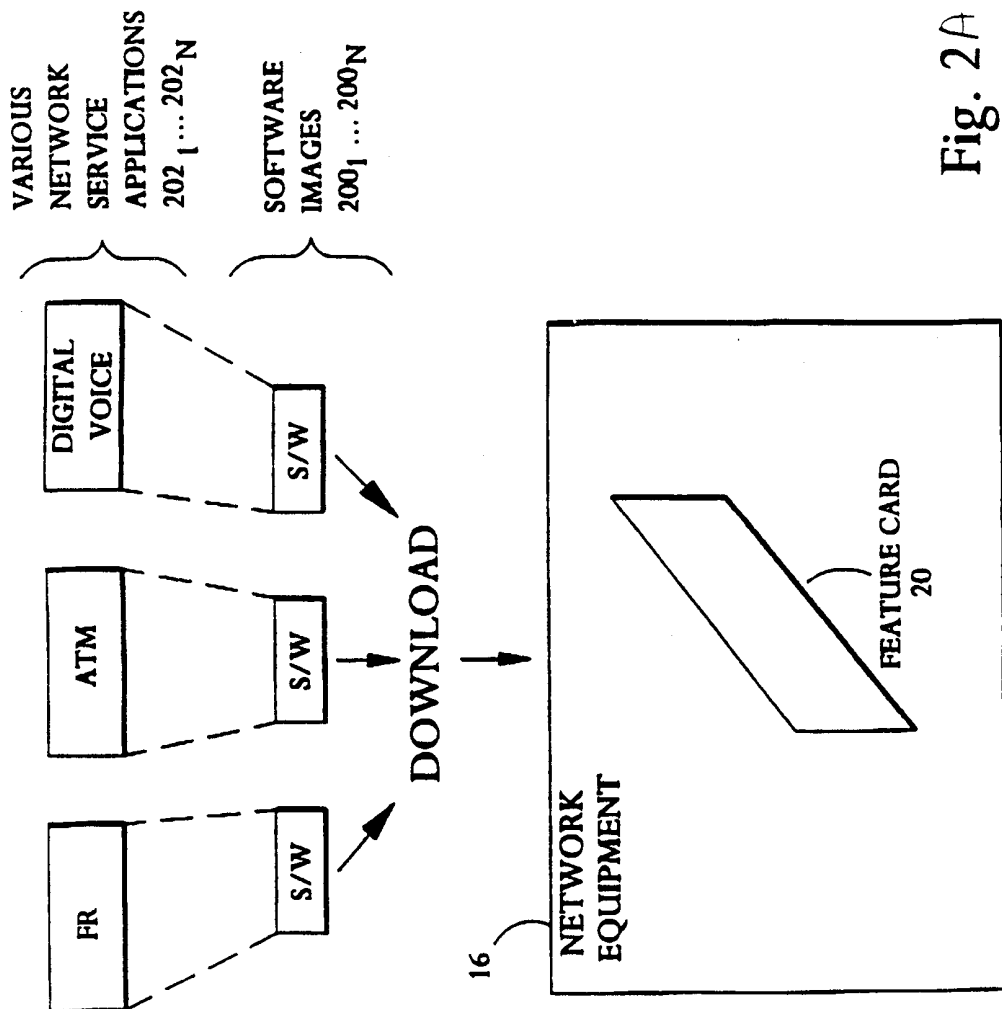


Fig. 2A

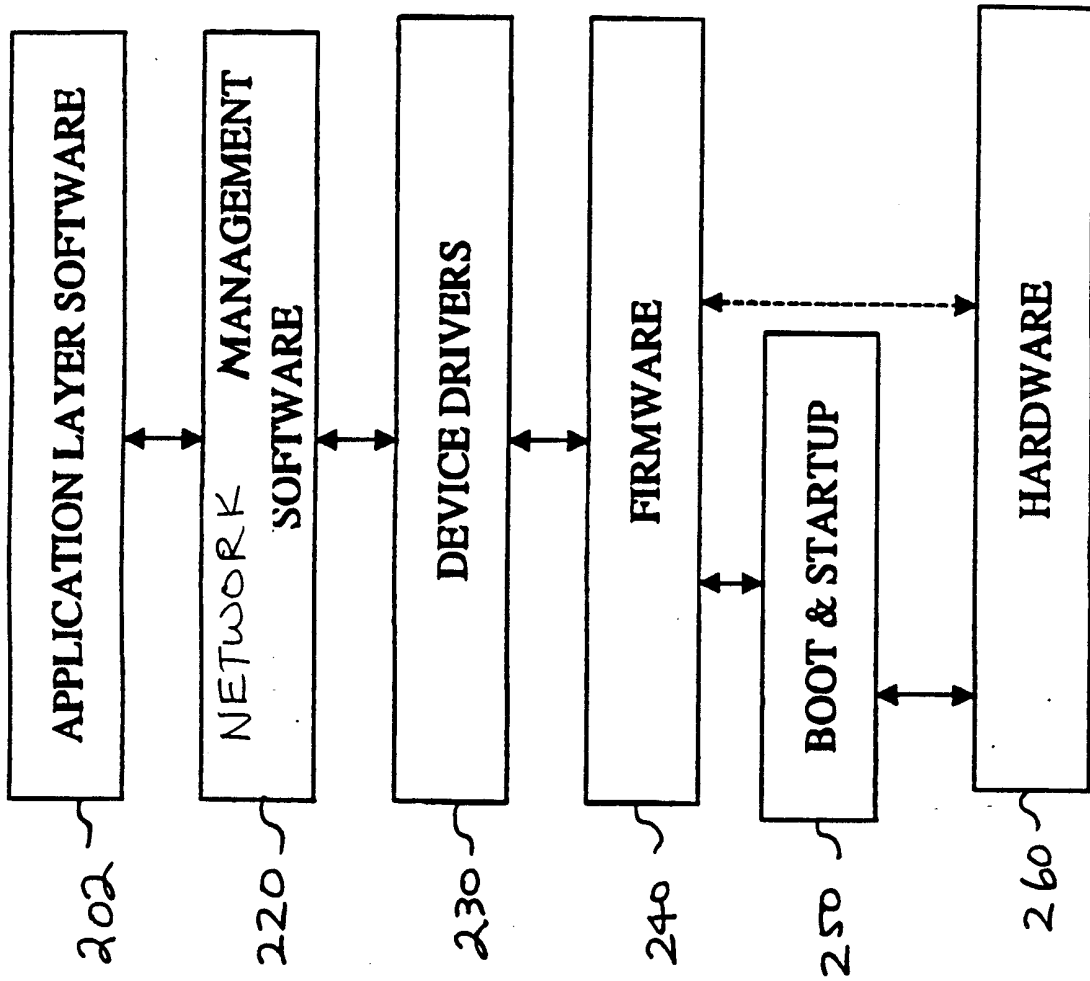


FIG. 2B

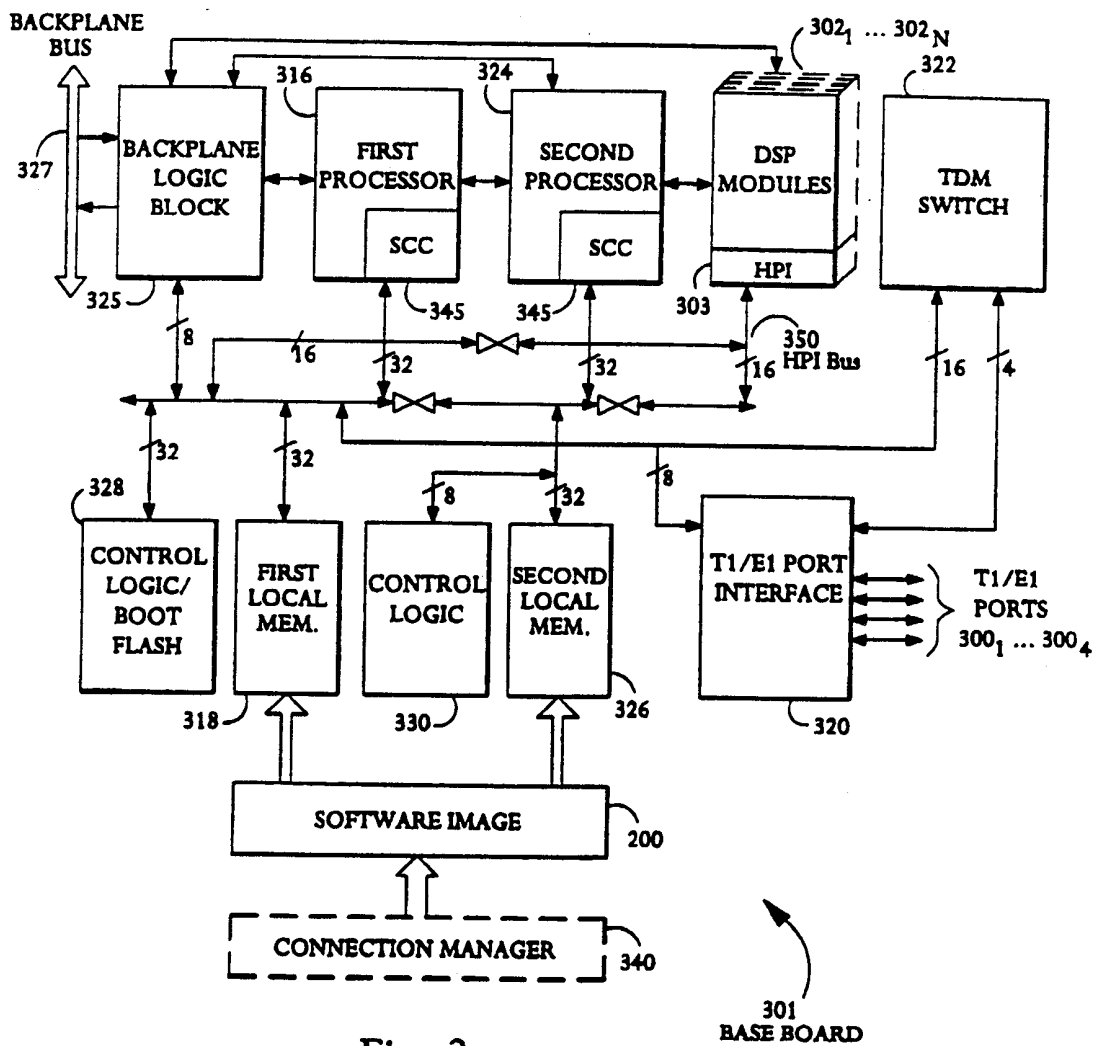


Fig. 3

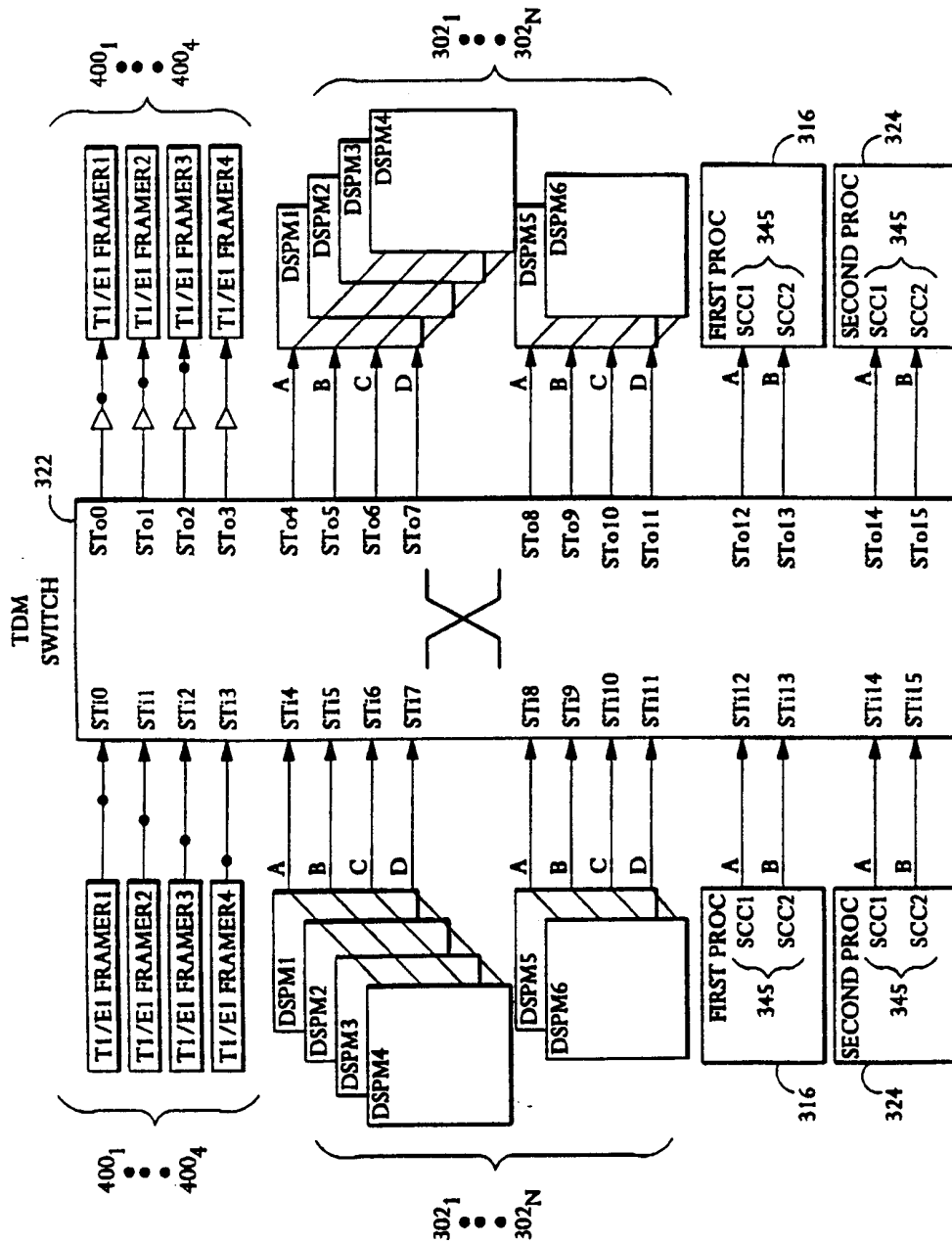


Fig. 4

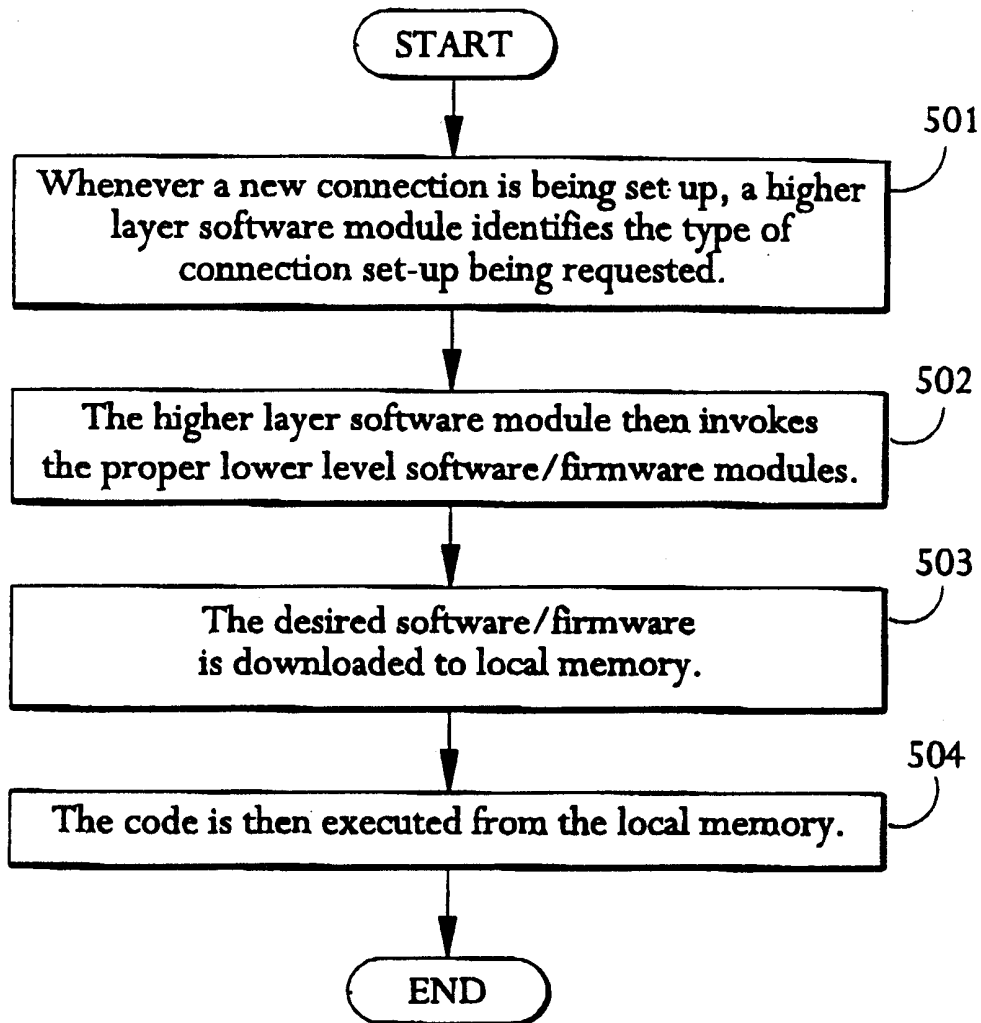


Fig. 5