(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
15 March 2007 (15.03.2007)

PCT

(10) International Publication Number
WO 2007/030301 A2

(51) International Patent Classification:
A63F 9/24 (2006.01)

(21) International Application Number:
PCT/US2006/032479

(22) International Filing Date: 17 August 2006 (17.08.2006)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:

11/220,781	7 September 2005 (07.09.2005)	US
11/319,034	23 December 2005 (23.12.2005)	US
11/380,854	28 April 2006 (28.04.2006)	US

(71) Applicant (*for all designated States except US*): **BALLY GAMING INTERNATIONAL, INC.** [US/US]; 6601 South Bermuda Road, Las Vegas, NV 89119-7990 (US).

(72) Inventors; and
(75) Inventors/Applicants (*for US only*): **SALLS, David** [US/US]; 950 Sandhill Rd., Reno, NV 89521 (US). **MORROW, James, W.** [US/US]; 5032 Pleasant View Dr., Sparks, NV 89434 (US).

(74) Agent: **KOVELMAN, Robert, L.**; Steptoe & Johnson, LLP, 1330 Connecticut Avenue, NW, Washington, DC 20036 (US).

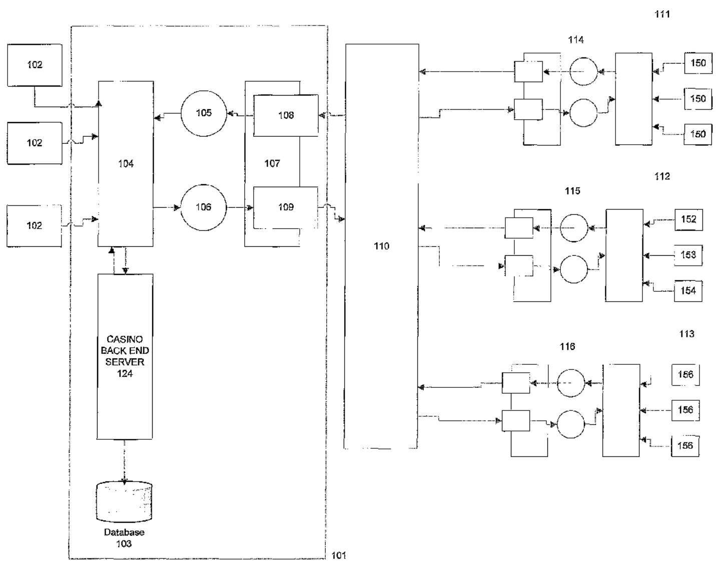
(81) Designated States (*unless otherwise indicated, for every kind of national protection available*): AE, AG, AL, AM, AT, AU, AZ, BA, BB, BG, BR, BW, BY, BZ, CA, CH, CN, CO, CR, CU, CZ, DE, DK, DM, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LV, LY, MA, MD, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PG, PH, PL, PT, RO, RS, RU, SC, SD, SE, SG, SK, SL, SM, SV, SY, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) Designated States (*unless otherwise indicated, for every kind of regional protection available*): ARIPO (BW, GH, GM, KE, LS, MW, MZ, NA, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HU, IE, IS, IT, LT, LU, LV, MC, NL, PL, PT, RO, SE, SI, SK, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, ML, MR, NE, SN, TD, TG).

Published:
— *without international search report and to be republished upon receipt of that report*

[Continued on next page]

(54) Title: HYBRID GAMING NETWORK



(57) Abstract: [0086] The invention provides a method and apparatus for linking a plurality of independent networks to a single central control system. In one embodiment, a casino accounting and reporting processing system provides a common back end to a plurality of networks. In one embodiment, the central server is coupled to at least one network using the same protocol so that no translation or interface is necessary. One or more other networks are coupled to the central server via an interface for translating communications and commands from the independent network to the host network protocol. In one embodiment, the interface is implemented using a standard interface referred to as "S2S". An advantage of the invention is that it can be used to provide a common user interface, common accounting and security, and common player tracking management.

WO 2007/030301 A2

For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.

HYBRID GAMING NETWORK

Background of the Invention

[0001] 1. Field of the Invention The claimed invention relates generally to a network, and more particularly, to a gaming network with an identification and communication system for network devices.

[0002] 2. Background In early gaming environments, gaming machines were stand-alone devices. Security of the gaming machines was accomplished via physical locks, security protocols, security personnel, physical and video monitoring, and the need to be physically present at a machine to attempt to breach the security of the gaming machine. By the same token, management of the gaming machines required a great deal of personal physical interaction with each gaming machine. The ability to change parameters of the gaming machine also required physical interaction.

[0003] In view of the increased processing power and availability of computing devices, gaming machines have become customizable via electronic communications and remotely controllable. Manufacturers of gaming equipment have taken advantage of the increased functionality of gaming machines by adding additional devices and features to gaming machines, thereby maintaining a player's attention to the gaming machines for longer periods of time increasing minimum bet and bet frequency and speed of play. This, in turn, leads to the player wagering at the gaming machine for longer periods of time, with more money at a faster pace, thereby increasing owner profits.

[0004] One technique that has been employed to maintain a player's attention at the gaming machine has been to provide players with access to gambling-related information. In this regard, attaching a small electronic display to the gaming device, gambling-related information, as well as news and advertisements can be sent to the player. The gambling-related information may include, for example, information on sports betting and betting options for those sporting events. Additionally, the gambling-related information may also include information such as horse racing and off-track betting. News and advertisements can also maintain a player's attention by providing the player with access to information ranging from show times, to restaurant and hotel specials, and to world events, thus reducing the need and/or desire of the player to leave the gaming machine.

[0005] Moreover, it has been shown to be desirable to provide the player with interactive access to the above information. This type of interactivity allows players significantly more flexibility to make use of the above-described information. The gambling-related information can also be utilized by the player in a much more efficient manner. In this regard, greater levels of flexibility and access are likely to make the player remain and gamble at the gaming machine for significantly longer periods of time.

[0006] In addition, the player may participate in a "premium" promotion where the player is registered with the gaming establishment as a club member when the player inserts an ID card into the gaming machines during play. The player may be rewarded for certain play patterns (e.g. wager amounts, wager totals, payouts, time of play, or the like) and earn redeemable benefits or upgrade of club member status.

[0007] Attempts to distribute gambling-related information and advertisements to players and to allow the recognition of premium membership players have resulted in additional system components that may be attached to the gaming devices. These components for accessing and displaying information for gaming machines may include a keypad, card reader, and display equipment (including, for example, color touch-screens) It is also desired to have a common user interface for all gaming machines including having a common location for devices such as the card reader, keypad, and display.

[0008] The amount of interactivity and data presentation/collection possible with current processor based gaming machines has led to a desire to connect gaming machines in a gaming network. In addition to the gaming machines themselves, a number of devices associated with a gaming machine or with a group of gaming machines may be part of the network. It has become important for the devices within a gaming machine or cabinet to be aware of each other and to be able to communicate to a control server. Not only is the presence or absence of a network device important, but also the physical location of the device and the ability to associate devices within a particular gaming machine has become a necessary component of a gaming network.

[0009] Current networks for gaming machines have been primarily one-way in communication, have been slow, and have been proprietary (custom designed and incompatible with commercial networking equipment). Prior art networks provided accounting, security, and player related data reporting from the gaming machine to a backend server. Secondary auditing procedures allowed regulators and managers to double check network reporting, providing a method of detecting malfeasance and network attacks. However, such security is remote in time from when a network attack has occurred. Prior art networks lack many security features needed for more rapid detection of cheating from a variety of possible attackers.

[0010] One disadvantage of prior art networks is the inability to integrate one or more proprietary or independent networks to a common central server. The inability of certain networks to communicate with each other makes management of the networks more complex and costly. Accounting, player tracking, and security are problematic when messages originate from multiple networks.

[0011] Although prior art networks of gaming machines provide advantages to gaming establishment operators, they also engender new risks to security of the gaming establishment and to the gaming machines. Not only is traditional data associated with gaming machines now potentially at risk on the gaming network, but personal player information is now at risk, as well.

[0012] In addition, the proprietary nature of prior art gaming machine networks limits the ability to use commercially available technology. This adds to the cost of gaming networks and limits their scalability and the ability to upgrade as technology improves. Further, as gaming machines are grouped in networks, the value of the pooled financial data traversing the network creates a great temptation to attack the network. The potential reward from attacking a network of gaming machines is greater than the reward from attacking a single machine.

[0013] A gaming network may have a large number of dynamically changing and reconfigurable components. Because of the desire to keep down-time to a minimum, it is important that the population of devices on the network be determinable and verifiable. In the past, this has meant pre-programming knowledge of all other devices into each device, so that communication between devices could take place. Such a requirement of pre-programming or pre-knowledge is too time consuming to be practical in a gaming network environment.

[0014] In addition, operators desire to be able to access individual devices inside of a gaming machine from a central server or from other machines. In addition, it may be desirable for multiple machines to be able to communicate with a peripheral so that peripherals may be shared. In other cases, it may be desired to temporarily use another machine's peripheral upon failure of one of its own peripherals.

[0015] Another disadvantage of current gaming systems is the hybrid nature of networks that include a number of different protocols. Many legacy devices are unable to effectively communicate with other devices in the system. In addition, proprietary protocols add to the expense of device manufacture and replacement.

Summary of the Invention

[0016] The invention provides a method and apparatus for linking a plurality of independent networks to a single central control system. In one embodiment, a casino accounting and reporting processing system provides a common back end to a plurality of networks. In one embodiment, the central server is coupled to at least one network using the same protocol so that no translation or interface is necessary. One or more other networks are coupled to the central server via an interface for translating communications and commands from the independent network to the host network protocol. In one embodiment, the interface is implemented using a standard interface referred to as "S2S". An advantage of the invention is that it can be used to provide a common user interface, common accounting and security, and common player tracking management.

[0017] These and other features and advantages of the claimed invention will become apparent from the following detailed description when taken in conjunction with the accompanying drawings, which illustrate, by way of example, the features of the claimed invention.

Brief Description of the Drawings

[0018] Figure 1 is a block diagram of an embodiment of the invention.

[0019] Figure 2 is a block diagram illustrating an embodiment of a communication model using the interface of the invention.

[0020] Figure 3 is a flow diagram illustrating communication in an embodiment of the invention.

[0021] Figures 4A, 4B, and 4C are block diagrams of a gaming machine configuration in embodiments of the invention.

[0022] Figure 5 is a flow diagram illustrating one embodiment of game machine device management using the invention.

[0023] Figure 6 is a flow diagram illustrating the transmission step of Figure 5.

[0024] Figure 7 is a flow diagram illustrating the identification step of Figure 5.

[0025] Figure 8 is a flow diagram illustrating the communication step of Figure 5.

DETAILED DESCRIPTION

[0026] The claimed invention is directed to a hybrid network where one or more independent networks may be managed by a single central server. The preferred embodiments of the system and method are illustrated and described herein, by way of example only, and not by way of limitation.

[0027] Referring now to the drawings, wherein like reference numerals denote like or corresponding parts throughout the drawings, there are shown embodiments of the hybrid gaming network constructed in accordance with the claimed invention. Figure 1 is a block diagram illustrating an embodiment of the invention. A central server or back end system 101 is configured as the controller of the system. Although this back end system may be comprised of many parts, it may be referred to herein as a central server 101. The protocol upon which the central server 101 operates is referred to as the native protocol. In the embodiment shown, a native network 102 is directly coupled to the central server 101. The native network 102 is a group of game machines that operate using the same protocol as the central server 101. It should be understood that the invention may be practiced with or without one or more native networks.

[0028] The central server 101 includes a database 103 for storing operational data, performance data, financial data, security data, and the like. An application layer 104 interfaces between the native network 102 and the command queues 105 (inbound) and 106 (outbound). In one embodiment, application layer 104 communicates with a casino back end server 124. Command queues 105 and 106 communicate with protocol layer 107. Server108 communicates (receiving messages from independent networks) with the interface 110 and is a web server in one embodiment capable of XML parsing. Client 109 is coupled to interface 110 and is used to send messages to independent networks.

[0029] Interface 110 is used to bridge between the native and the independent networks. As referred to herein, an independent network is a network using a different protocol than the native protocol of the central server 101. These independent protocols may or may not include some mixture of proprietary and standard protocols. An example of a proprietary protocol is the Slot Data Transport (SDT) by Bally Gaming Systems, or RSM, a binary TCP messaging structure that is socket based, used for near real-time communication, also by Bally Gaming Systems. The interface provides the ability to translate commands and data from one network protocol to another. In this case, it provides a way to translate from one or more independent protocols to the native protocol and vice-versa. Communication is not limited to only between the central server and the independent networks. It should be realized that the native and independent networks may communicate with each other, using the interface to facilitate communication.

[0030] In one embodiment of the invention, the interface is implemented using the S2S (System to System) protocol defined by the Gaming Standards Association (GSA)

[0031] In the example of Figure 1, there are a number of independent networks 111, 112, and 113 coupled to the central server 101 via the interface 110. Independent network 111 is shown by way of example as an IGT Class II system. The network 111 comprises a plurality of gaming machines 150 coupled to system processor 114. Network 111 in one embodiment communicates using web services via SOAP ("Simple Object Access Protocol") over HTTPS. Network 112 comprises a network (e.g. an NRT network) with a plurality of bill acceptors 152 or other coin-in devices 153, 154 coupled to network system processor 115. Network 112 communicates with direct posts using HTTP in one embodiment of the invention. Network 113 is an anticipated future network that uses a protocol not currently in use and includes a plurality of gaming machines 156 and network system processor 116. Network 113 in one embodiment communicates via socket or web services. The invention contemplates use with existing network protocols and future network protocols.

[0032] In the example of Figure 1, protocol layer 107 is capable of communicating with a number of independent networks using a variety of protocols. Protocol layer 107 can use web services to communicate using SOAP, use direct posts using HTTP, or via sockets.

[0033] Figure 2 is a block diagram illustrating a communication model in an embodiment of the interface 110. The native network system is represented on the left side of Figure 2 and an independent network is represented on the right side. Application layer 201 and business layer 202 respond to commands in messages directed to the native network. In one embodiment, business layer 104 communicates with a casino back end server 220. Application layer 218 and Business layer 217 respond to commands in messages directed to the independent network.

[0034] Application layer 201 receives messages (and commands) via inbound command queue 203. Application layer 201 sends messages to the network via outbound command queue 204. The queues 203 and 204 are coupled to the protocol layer 207. At the protocol level, messages are transmitted between the network hosts.

[0035] Application layer 218 receives messages and commands via inbound command queue 216 and sends messages to the network via outbound command queue 215. Queues 215 and 216 are coupled to protocol layer 214.

[0036] Protocol layer 207 includes a web server 205 and client connection 206. Similarly, protocol layer 214 includes a client connection 212 and web server 213. In the embodiment shown in Figure 2, communication is accomplished via a request-response method. Therefore, there are separate channels for inbound and outbound communication of messages. The inbound channel for protocol layer 207 (and correspondingly the outbound channel for protocol layer 214) is represented by the connection between server 205 of protocol layer 207 and client connection 212 of protocol layer 214. Client connection 212 sends a message 208 to server 205. An acknowledgement (ACK) 209 is sent back by server 205 to client connection 212 when a message has been received.

[0037] The outbound channel for protocol layer 207 (and inbound channel for protocol layer 214) is represented by the connection between client connection 206 of protocol layer 207 and web server 213 of protocol layer 214. Client connection 206 sends a message 210 to web server 213. An ACK 211 is sent back by server 213 to client connection 206 when a message has been received.

[0038] In the embodiment illustrated, the ACK only indicates that a message and the commands have been received and does not represent an acknowledgement that the commands will be executed. The processing of commands in the system of Figure 2 is illustrated in the flow diagram of Figure 3. At step 301 the native network A sends a message to independent network B. The message includes a command. At step 302 network B sends an ACK to network A acknowledging the receipt of the message and command. At some later time at step 303 network B processes the command. (The embodiment illustrated contemplates asynchronous operation so that the commands do not have to be processed immediately upon receipt). At step 304 network B may send a message to network A that confirms that the original command has been processed and perhaps including a command from network B to network A. At step 305 network A sends an ACK to network B. At step 306, network A processes the command.

[0039] <u>Message Format</u>

[0040] In one embodiment of the invention, a standard message format is used to enhance the ability to promote cross network and cross protocol communication. In one embodiment, a message element forms the outer wrapper of an original message and the acknowledgement. An acknowledgement is composed of a single element "Acknowledge". A message comprises a header and a body. The header identifies the sender, the receiver and information about the message. The body of the message contains application level commands.

[0041] The header includes attributes that identify the message, including a system identifier of the originator, the system identifier of the recipient, a unique message identifier, and a date stamp. The system identifier permits the appropriate translation to take place for ease of communication between native and non-native systems.

[0042] The application command consists of a class-level element and a command-level element. A message element may contain multiple commands comprising the class-level/command-level pair.

[0043] Gaming Machine

[0044] The invention contemplates the ability to interact with gaming devices on different networks. The following figures give examples of embodiments of gaming machines that may be used with the invention. Figure 4A is a block diagram of an example gaming machine configuration in an embodiment of the invention. The gaming machine 415 communicates with its associated native or non-native network via communications path 414 which may be Ethernet, wireless Ethernet, wire, fiber, wireless, or any other suitable communication link. The gaming machine 415 may include a communications interface 401 that handles communication between the gaming machine and its associated devices and the remainder of the gaming network. Communication interface 401 is coupled to an MPU 402. The MPU serves as the processor of the gaming machine. An interface referred to as a "SMIB" 403 (smart interface board or slot machine interface board) is coupled to the MPU and to the communication interface 401. SMIB 403 is coupled to one or more peripherals or other devices connected to the gaming machine 415, such as devices 404A to 404N of Figure 4A. In one embodiment of the invention, SMIB 403 uses an Ethernet or other high-speed communications link to the communication interface 401, MPU 402, and devices 404A through 404N. In one embodiment, the SMIB includes switching capabilities. In one embodiment, the SMIB is implemented with a Mastercom 300 manufactured by Bally Technologies.

[0045] Figure 4B illustrates an alternate embodiment of a gaming machine and peripherals. The gaming machine communicates with its associated network via communications path 414 (which may be an Ethernet connection). Communication is handled by network distribution device 405. This device could be an Ethernet hub, for example, or any other suitable communications interface. The game machine includes an MPU 402 that provides processing for the game. A number of peripherals are included in the gaming machine and are coupled directly to the network distribution device 405 or to the MPU 402. In the embodiment of Figure 4B, such peripheral devices include lights 406, keypad 407, card reader 408, primary display 409, lights 410, button deck 411, printer 412, hopper 413, coin acceptor 414, bill acceptor 415, and secondary display 416. It is understood that not every gaming machine will have this exact configuration of peripherals. A gaming machine may have fewer or more peripherals, and different peripherals, without departing from the scope of the invention.

[0046] An alternate embodiment of a gaming machine and peripherals is illustrated in Figure 4C. Here the gaming machine remains coupled to its associated network via communication path 414 and network distribution device 405. The gaming machine includes an MPU 402 and a number of peripheral devices. In this embodiment, the devices are coupled to the network distribution device 405 and/or the MPU 402 via a plurality of protocols. These protocols could include parallel connections (e.g. lights 406), I^2C connections (e.g. keypad 407), USB (e.g. card reader 408), LVDS (e.g. primary display 409) and other protocols. In this embodiment, the MPU 402 and/or the network distribution device 405 convert from the non-Ethernet protocol to Ethernet protocol for communication via the network.

[0047] Each gaming network may use a number of network services for administration and operation. Dynamic Host Configuration Protocol (DHCP) allows central management and assignment of IP addresses within the gaming network. The dynamic assignment of IP addresses is used in one embodiment instead of statically assigned IP addresses for each network component. A DNS (domain name service) is used to translate between the domain names and the IP addresses of network components and services. DNS servers are well known in the art and are used to resolve the domain names to IP addresses on the Internet.

[0048] Similarly, Network Time Protocol (NTP) may be used to synchronize time references within the network components for security and audit activities. It is important to have a consistent and synchronized clock so that the order and the timing of transactions within the gaming network can be known with reliability and certainty. Network information can be gathered centrally at a single workstation by using the Remote Monitoring (RMON) protocol. SNMP (simple network management protocol) allows network management components to remotely manage hosts on the network, thus providing scalability. In one embodiment of the gaming network, SNMPv3 is used to take advantage of embedded security mechanisms to mitigate malicious attacks made against the configuration management function. Still further, TFTP (trivial file transfer protocol) is used by servers to boot or download code to network components.

[0049] In one embodiment, the network may be implemented using the IPv6 protocol designed by the IETF (Internet Engineering Task Force). When using IPv6, the network may take advantage of the Quality of Service (QoS) features available with IPv6. QoS refers to the ability of a network to provide a guaranteed level of service (i.e. transmission rate, loss rate, minimum bandwidth, packet delay, etc). QoS may be used as an additional security feature in that certain transactions may request a certain QoS as a rule or pursuant to some schedule. Any fraudulent traffic of that nature that does not request the appropriate QoS is considered an attack and appropriate quarantine and counter measures are taken.

[0050] Similarly, the Type of Service (ToS) capabilities of IPv4 may also be used in a similar manner to provide additional security cues for validation of transactions. Again, certain types of transactions may be associated with a particular specific ToS or a rotating schedule of ToS that is known by network monitors.

[0051] <u>Traffic Content</u>

[0052] In an embodiment of the gaming network, the traffic content varies in size and sensitivity. Messages may comprise transactional messages related to game play, such as coin-in. Other messages may be related to management, administration, or sensitive information, such as administrator passwords, new game code, pay tables, win rates, patron personal data, or the like.

[0053] <u>Device Addressing</u>

[0054] In one embodiment of the invention, each device, including each peripheral device in a game machine, could have its own MAC address. The central server could be responsible for assigning IP addresses to each device and gathering the devices into bindings that represent a physical location of a device. For example, all of the bindings of a gaming machine may be in a single binding. Alternatively, the network distribution device 405 may be responsible for assigning IP addresses to the devices within the game machine.

[0055] In one embodiment, there is a common IP address (e.g. 10.5.5.32) for the gaming machine and that is owned by network distribution device 405. (In some cases, the MPU 402 and/or other devices, such as the printer, may have their own IP addresses). The remaining devices in the gaming machine are addressable by a unique port assignment attached to the common IP address. For example, PORT 5020 could be associated with the card reader 408, PORT 5030 with the hopper 413, and the like. Even though the physical connection with the peripheral may be, for example, an RS232 connection, the MPU 402 and network distribution device 405 communicate using the IP and PORT address. In some cases the communication itself may use a proprietary protocol yet still use the IP and PORT address as a destination.

[0056] In cases where multiple protocols are used in the system, the network distribution service can act as an Internet Protocol Exchange (IPX) to facilitate the translation of TCP/IP network traffic to the native protocol of a device and vice-versa. In one embodiment, the device enablement may take advantage of a USB sharing hub such as the USB Multi-switch manufactured by Standard Microsystems, Corp.

[0057] Initialization of Gaming Machine Devices

[0058] An embodiment of the invention provides a process for identifying devices coupled to a game machine. This process is described in Figure 5. At step 501, during initialization, each device (e.g. devices 404A – 404N) attempts to communicate with the network and transmits its MAC/IP address. The address is received by a switch in the game machine (e.g. the SMIB 403, network distribution device 405, or the like) and a table of addresses of associated devices is assembled. This table is made available to the devices in the game machine so that the IP addresses of other devices within the gaming machine become available to each device.

[0059] At the identification step 502 each device identifies itself to other devices in the gaming machine. At step 503 a verification process is initiated so that it can be determined if the devices are valid devices on the network. At step 504 devices may begin to transmit data between themselves and to the core layer or other back-end server of the network.

[0060] MAC/IP Transmission

[0061] A description of one embodiment of the MAC/IP transmission of step 501 is illustrated in the flow diagram of Figure 6. During a boot or initialization sequence 601, any network-connected device inside the gaming machine will attempt to communicate with the network at step 602 by sending its MAC/IP address via the SMIB or other switching device. The nature of this initial communication may be for a DHCP or BOOTP configuration, an ARP request, or any other attempt to identify itself to the back-end system. The MAC/IP addresses that are part of these communication attempts are added at step 603 to a table. This table is managed by the SMIB 403 in one embodiment, or by the MPU 402 in another embodiment. Eventually at step 604, a table will be generated that contains the MAC/IP addresses of all of the devices in the gaming machine.

[0062] In one embodiment, the devices send only their MAC addresses but the switch or other management device associates an IP address with each MAC address to populate a table. This embodiment may be used when IP addresses are assigned dynamically as described above.

[0063] At step 605, the switch or MPU, or whichever device is managing the address table, periodically transmits raw Ethernet frames, USB packets, or TCP packets that include a list of the attached MAC/IP addresses associated with that game machine. In one embodiment, the frame is sent on a regular basis (e.g. every three to five seconds) so that other devices can expect that frame and react appropriately if it is not received. The transmitted frame is sent to switches and game machines on the network. In one embodiment, the transmission is via User Datagram Protocol (UDP) but any suitable protocol may be used without departing from the spirit and scope of the invention. In this manner, game machine devices need only be able to recognize the frame to take action. Eventually all of the MAC/IP addresses of game machine devices are published throughout the network. In this embodiment, there is no necessity of flooding the network with broadcasts frames with address information. This information is distributed organically throughout the network.

[0064] The process in one embodiment is an ongoing process, shown by the return path from step 605 to step 602 in Figure 6. The tables are rebroadcast periodically by the switch. This rebroadcast allows devices to learn about other new devices that have been added to the network. It also allows a device to know when another device has left the network.

[0065] At this point in the process the information being collected is pre-authentication. It allows a list of possible devices to be known and addressable so that if the device is valid and authenticated, it can participate on the network.

[0066] <u>Identification</u>

[0067] The identification process 502 is described in conjunction with Figure 7. A device receives a MAC/IP transmission frame from the switch at step 701. This is an ongoing process during runtime as the switch periodically transmits Ethernet frames containing updated and new MAC/IP address information as described above. At step 702 the device identifies other devices within the same game machine or cabinet from information in the Ethernet frame. At step 703 the device initiates an identification communication with one or more other devices in the game machine. The form of this transmission at step 704 may be as simple as sending an "I'm here" message. In other embodiments, the identification message may include identification information about the device at step 704. This information may include information such as the port address, device ID, a preferred communication protocol, and the like. In other embodiments, such information is provided during communication negotiations.

[0068] Verification

[0069] Once two devices have identified themselves to each other, a verification procedure can take place. The verification procedure is intended to establish that the device with which another device is communicating is a valid gaming device. In one embodiment of the invention, verification may be accomplished by using the protocol described herein in connection with Figures 3 and 4. Any suitable verification protocol may be utilized without departing from the scope and spirit of the invention. In-cabinet devices have similar security concerns as other network devices described herein. For example a device may not only need to identify itself to a central server, but may also need to identify itself to another device within the same cabinet.

[0070] In one embodiment, a verification method is used such as is described in pending U. S. Patent application number 10/243,912, filed on September 13, 2002, and entitled "Device Verification System and Method", assigned to the assignee of the invention, and incorporated by reference herein in its entirety. The invention provides a system and method for verifying a device by verifying the components of that device. The components may comprise, for example, software components, firmware components, hardware components, or structural components of an electronic device. These components include, without limitation, processors, persistent storage media, volatile storage media, random access memories, read-only memories (ROMs), erasable programmable ROMs, data files (which are any collections of data, including executable programs in binary or script form, and the information those programs operate upon), device cabinets (housings) or cathode ray tubes

(CRTs). Identification numbers or strings of the components are read and then verified. The process of verifying may comprise matching each identification number in a database to determine whether each identification number is valid. In the case where a data file comprises one of a plurality of operating system files, verification of that file, in effect, comprises verifying part of an operating system. For data files, the file names may comprise the identification numbers.

[0071] The database may comprise a relational database, object database, or may be stored in XML format, or in a number of other formats that are commonly known. The database may also comprise an independent system stack of bindings, which comprise numbers, identification strings or signatures in the database for matching or authenticating the components, from manufacturers of the components, each identification number being verified using the binding from the manufacturer of the respective component to verify the component. Especially in the context of smaller devices such as personal digital assistants (PDAs), such a system stack may comprise a subset of one or more global component databases containing bindings from manufacturers of the components, each binding of the subset being associated with at least one of the identification numbers of one of the components in the device.

[0072] Structural components, such as cabinets, may contain an electronic identification chip embedded within them, such as a so-called Dallas chip or an IBUTTON device manufactured by Dallas Semiconductor of Dallas, Texas. These devices allow a unique identifier, placed within a semiconductor or chip, to be placed on a component that may or may not be electronic, such as a computer or gaming machine cabinet. The IBUTTON device is a computer chip enclosed in a 16 mm stainless steel can. The steel button can be mounted, preferably permanently or semi-permanently, on or in the structural component. Two wires may be affixed to the IBUTTON device, one on the top, and one on the bottom, to exchange data between the IBUTTON device and a processor, serial port, universal serial bus (USB) port, or parallel port.

[0073] The matching process may comprise matching each identification number based on the type of component that the identification number identifies. The identification number and the type of component are matched in the database in order to verify that the identification number is valid. Operation of the device may be stopped if any one of the identification numbers is not matched in the database. In the case of a game or gaming machine type of device, a tilt condition message is generated if any one of the identification numbers is not matched in the database.

[0074] The database may consist of a set of signatures, also called bindings. At least with respect to the components that comprise data files or firmware, a well-known hash function, the Secure Hash Function -1, also known as SHA-1, may be used to compute a 160-bit hash value from the data file or firmware contents. This 160-bit hash value, also called an abbreviated bit string, is then processed to create a signature of the game data using an equally well-known, one-way, private signature key technique, the Digital Signature Algorithm (DSA). The DSA uses a private key of a private key/public key pair, and randomly or pseudorandomly generated integers, to produce a 320-bit signature of the 160-bit hash value of the data file or firmware contents. This signature is stored in the database in addition to the identification number.

[0075] Either contained in the device, or in communication with the device, is a processor and a memory containing executable instructions or a software program file for verification of the components (verification software), which may itself be one of the components to verify. The verification software may be stored on a persistent storage media such as a hard disk device, read only memory (ROM), electrically erasable programmable read-only memory (EEPROM), in the aforementioned CMOS memory, battery-backed random access memory, flash memory or other type of persistent memory. Preferably, the verification software is stored in a basic input/output system (BIOS) on a solid-state persistent memory device or chip. BIOS chips have been used for storing verification software, such as the BIOS+ chip used by Bally Gaming Systems, Inc. of Las Vegas, NV in their EVO gaming system. Placing the verification software in the BIOS is advantageous because the code in the BIOS is usually the first code executed upon boot or start-up of the device, making it hard to bypass the verification process.

[0076] Alternatively, the verification software may be stored in a firmware hub, which may comprise the part of an electronic device or computer that stores BIOS information. In personal computer hub technology, such as that manufactured by the Intel Corporation of Santa Clara, California, a hub is used in place of a peripheral component interconnect (PCI) bus to connect elements of chipsets.

[0077] The persistent storage media may be a removable storage unit such as a CD-ROM reader, a WORM device, a CD-RW device, a floppy disk device, a removable hard disk device, a ZIP disk device, a JAZZ disk device, a DVD device, a removable flash memory device, or a hard card device. However, the database is preferably stored in a non-removable, secure device either within the device being verified, or remotely on a server, in order to enhance security.

[0078] The verification software executes a DSA verification of the data files and firmware components. Also stored in the database is the public key of the private key/public key pair. For each data file and firmware component, as part of the DSA verification, the processor and verification software first computes the hash value of the digital contents of the component using the SHA-1 algorithm. The verification software then processes or authenticates this computed hash value, using the DSA signature verification algorithm, which also takes, as input, the aforementioned public key stored in the database. The verification part of the DSA produces a Boolean result (yes or no) as to whether the inputs solve the algorithm. If the algorithm is not solved by the inputs, then an unexpected result is produced, thereby failing to verify the particular component. This may cause a fault tilt to occur to prohibit the loading operation of the device. Otherwise, use of the device is permitted. A detailed description of the DSA can be found in the U.S. government's Federal Information Processing Standards Publication (FIPS) 186-2. That publication describes each step of the DSA signature generation and verification.

[0079] Alternatively, the set of executable instructions may use the Rivest-Shamir-Adleman (RSA) algorithm to verify the components. Using the RSA algorithm, a first abbreviated bit string or hash value is computed from each component's digital contents and encrypted into a digital signature. The digital signature is stored in the database along with the identification number for the component. When the device is verified, the component is verified by computing a second abbreviated bit string computed from the component's digital contents. The signature is retrieved from the database by searching the database for the identification number. The signature is decrypted to recover the first abbreviated bit string. The component is then verified by comparing the second abbreviated bit string with the first abbreviated bit string. If the first and second abbreviated bit strings do not match, then the component is not verified. As discussed below, this may cause a fault tilt to occur to prohibit the loading operation of the device. Otherwise, use of the device is permitted.

[0080] Instead of creating a digital signature for, or signing, each data file individually, collections of data files may be signed together in order speed up processing. The abbreviated bit strings, hash values, or signatures, also called digests, of the collection of data files are collected into a catalog file, and the catalog is signed as described above.

[0081] Communication

[0082] After verification between devices has been completed, they may begin communication. At step 801 of Figure 8, a device initiates a communication with another device. The sending device may include a section of the first message to provide needed information to the intended recipient. This information may include at step 802 the type of device, the protocol the device is using, any restrictions related to QOS, and other communication related information. At step 803 the recipient determines if it can communicate with the sender directly or if an interface is needed at decision block 804. If an interface is needed at step 806, the sender and receiver may need to communicate through the MPU or interface 110, for example, if the MPU includes software or firmware for translating appropriately for the devices. If the devices can communicate directly, then messages are sent back and forth using an accepted protocol at step 805.

[0083] The invention allows devices to be aware of each other's presence through MAC/IP transmissions. This permits the use of a single network port for each device to use to communicate with each other and with a back-end system. The devices do not need pre-knowledge of the MAC/IP addresses of other devices but can learn them at start up and during run-time. The system also allows a new device to be added to a game cabinet and have it be integrated and identified to the system without extensive IT effort.

[0084] Although the invention has been described in connection with in-cabinet devices identifying themselves to each other, it is not limited to such an application. The invention may be used to provide identification of any network devices by organically updating identification information periodically in Ethernet frames. In addition, the invention is not limited to the specific network configuration described herein. Rather, the system can work with any number of network configurations without departing from the scope and spirit of the invention.

[0085] It will be apparent from the foregoing that, while particular forms of the claimed invention have been illustrated and described, various modifications can be made without departing from the spirit and scope of the claimed invention. Accordingly, it is not intended that the claimed invention be limited, except as by the appended claims.

CLAIMS

What is Claimed is:

1. A gaming network comprising:

a first sub-network using a first network protocol;

a second sub-network using a second network protocol;

an interface coupled to the first and second sub-networks to translate messages between the first and second sub-networks.

2. The gaming network of claim 1 wherein the second network protocol is SOAP over HTTPS.

3. The gaming network of claim 1 wherein the first network protocol is a proprietary system.

4. The gaming network of claim 1 wherein the interface comprises an S2S transport protocol.

5. The gaming network of claim 1 further including a first inbound command queue in said first sub-network for receiving messages containing commands from the second sub-network.

6. The gaming network of claim 5 further including a first outbound command queue in said first sub-network for sending messages containing commands to the second sub-network.

7. The gaming network of claim 6 further including a second inbound command queue in said second sub-network for receiving messages containing commands from the first sub-network.

8. The gaming network of claim 7 further including a second outbound command queue in said second sub-network for sending messages containing commands to said first sub-network.

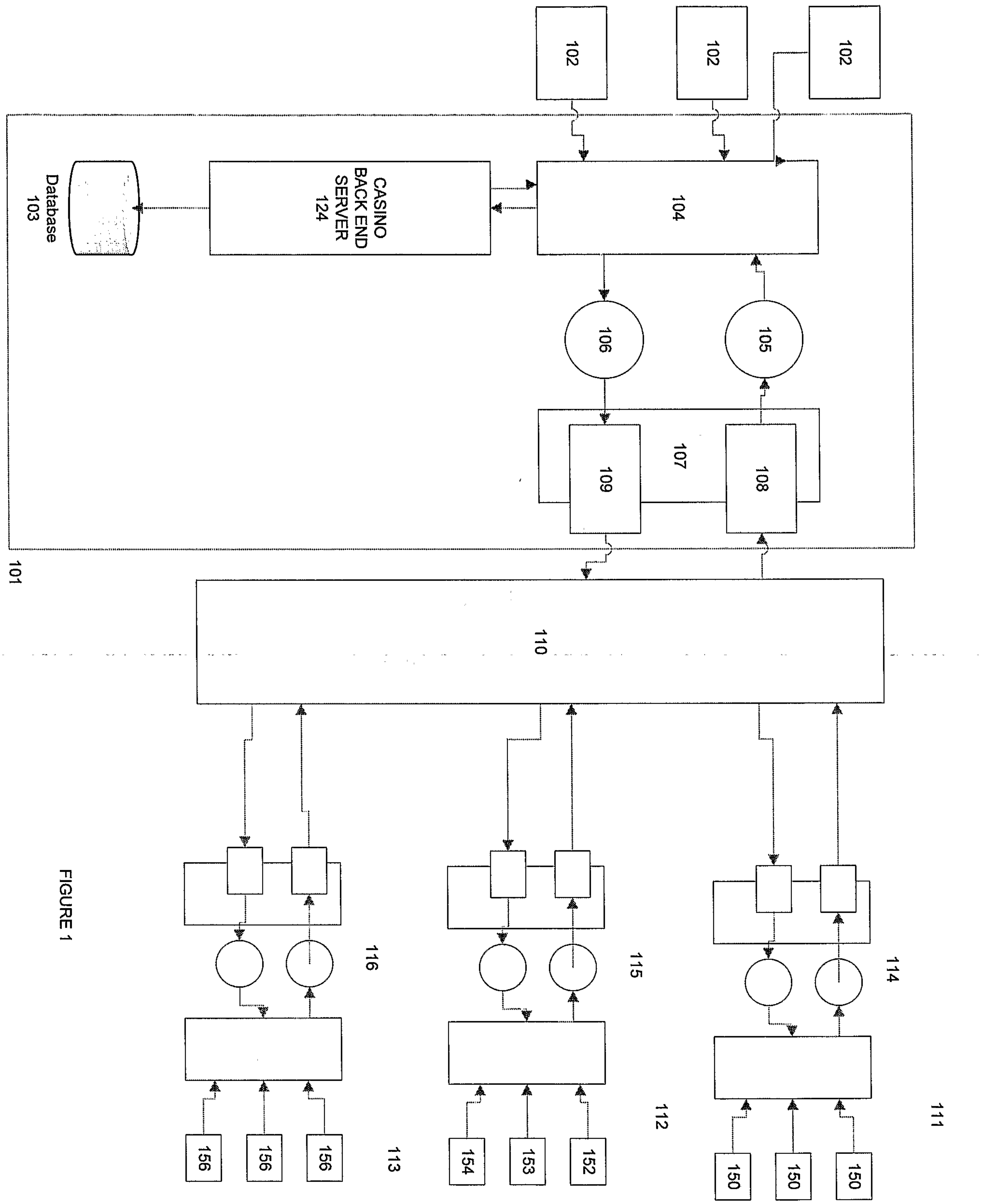

102
102
102
104
CASINO
BACK END
SERVER
124
Database
103
105
106
107
108
109
101
110
114
115
116
111
112
113
150
150
150
152
153
154
156
156
156

FIGURE 1

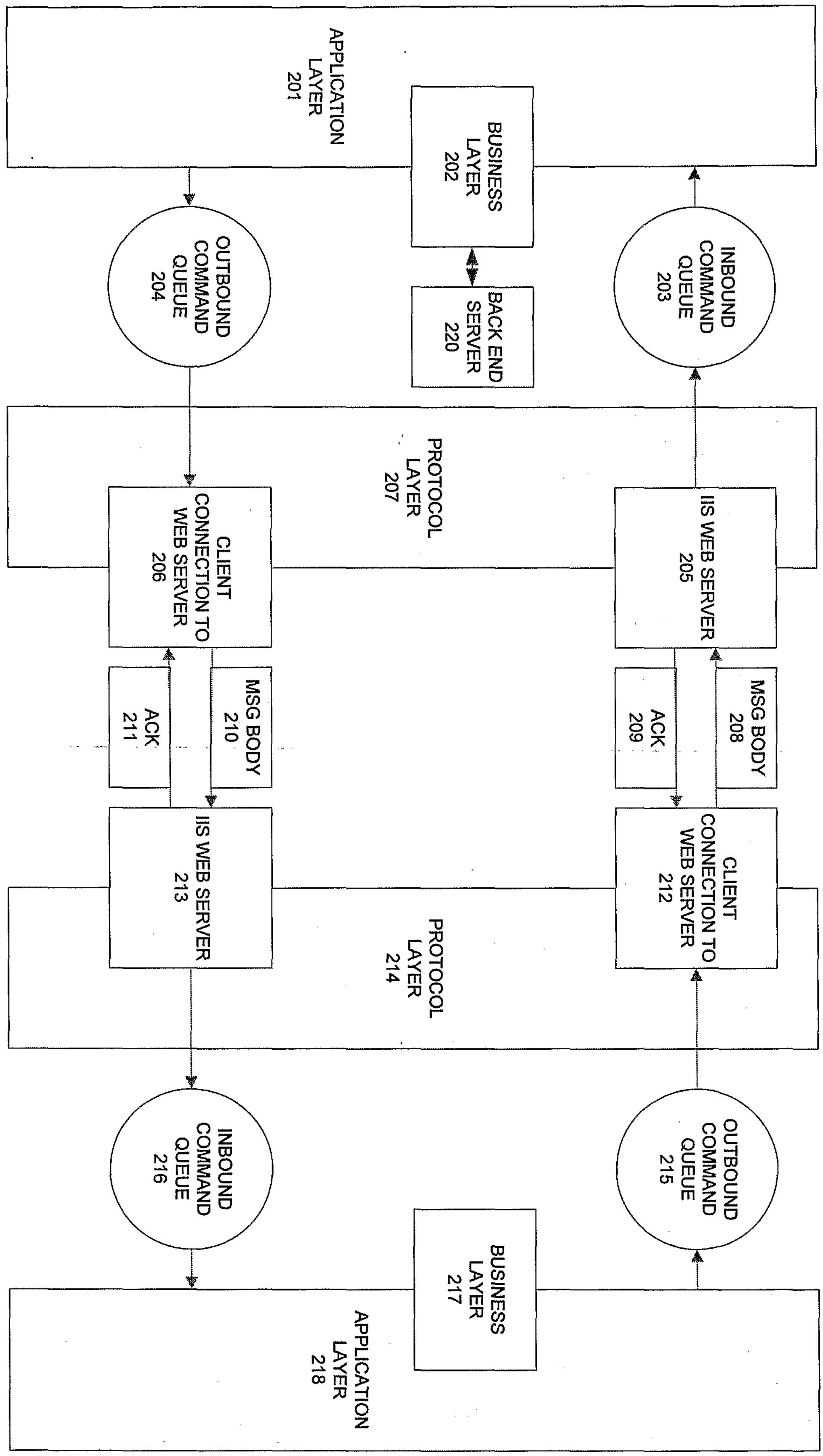

APPLICATION LAYER 201
BUSINESS LAYER 202
INBOUND COMMAND QUEUE 203
OUTBOUND COMMAND QUEUE 204
BACK END SERVER 220
PROTOCOL LAYER 207
CLIENT CONNECTION TO WEB SERVER 206
IIS WEB SERVER 205
MSG BODY 210
ACK 211
MSG BODY 208
ACK 209
IIS WEB SERVER 213
CLIENT CONNECTION TO WEB SERVER 212
PROTOCOL LAYER 214
INBOUND COMMAND QUEUE 216
OUTBOUND COMMAND QUEUE 215
BUSINESS LAYER 217
APPLICATION LAYER 218

FIGURE 2

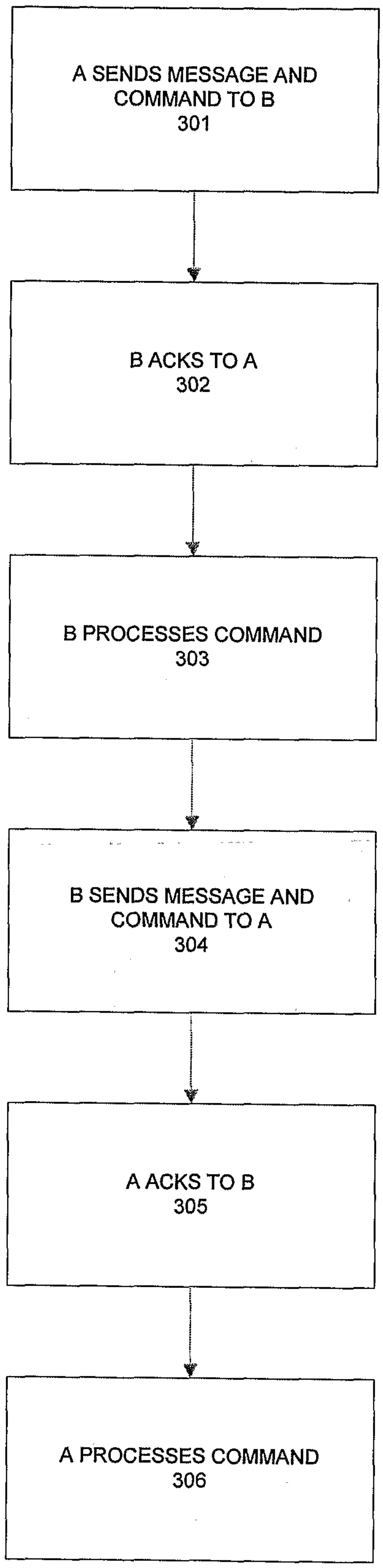
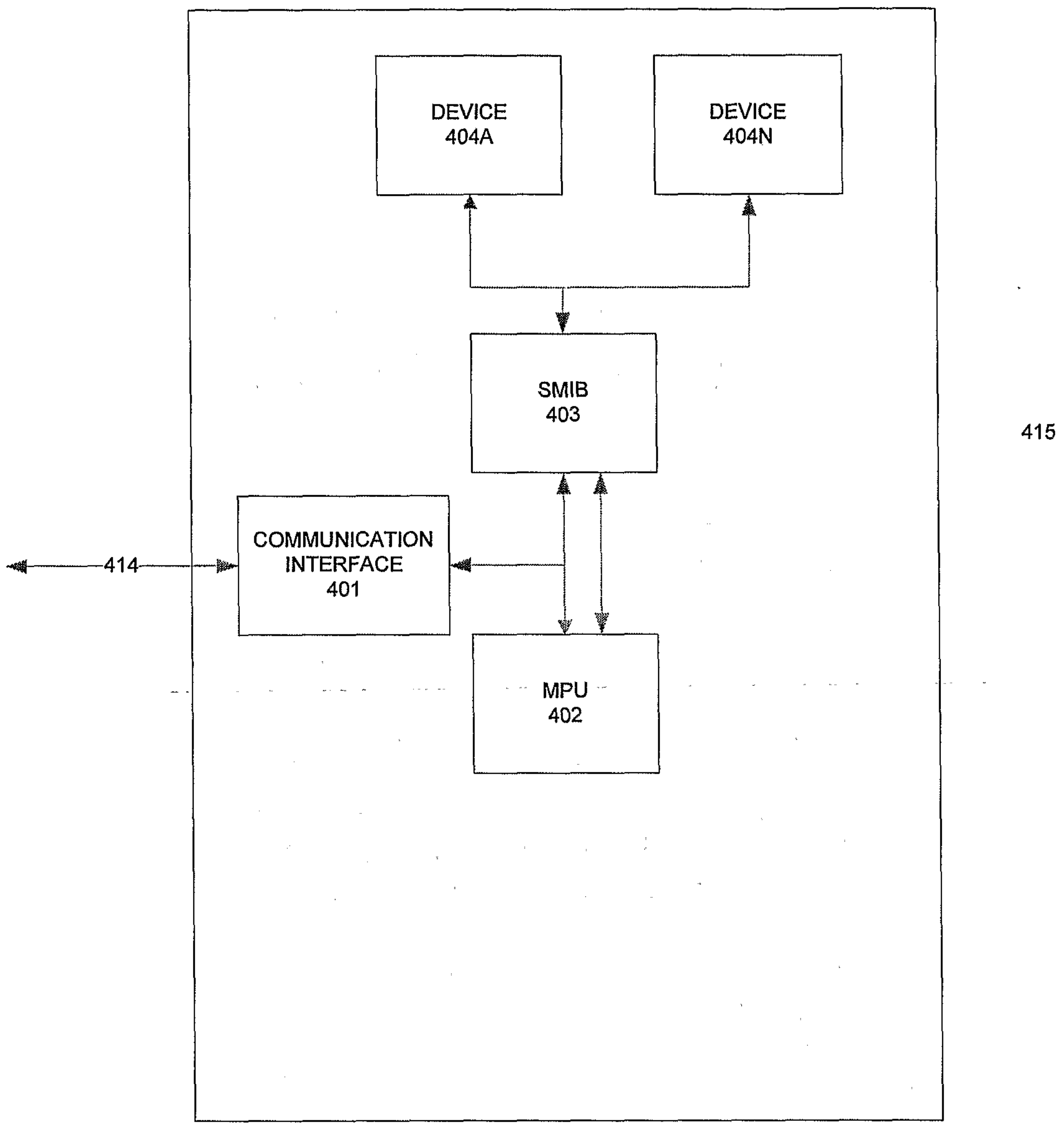

A SENDS MESSAGE AND COMMAND TO B
301
B ACKS TO A
302
B PROCESSES COMMAND
303
B SENDS MESSAGE AND COMMAND TO A
304
A ACKS TO B
305
A PROCESSES COMMAND
306

FIGURE 3

FIGURE 4A


DEVICE
404A
DEVICE
404N
SMIB
403
415
COMMUNICATION
INTERFACE
401
414
MPU
402

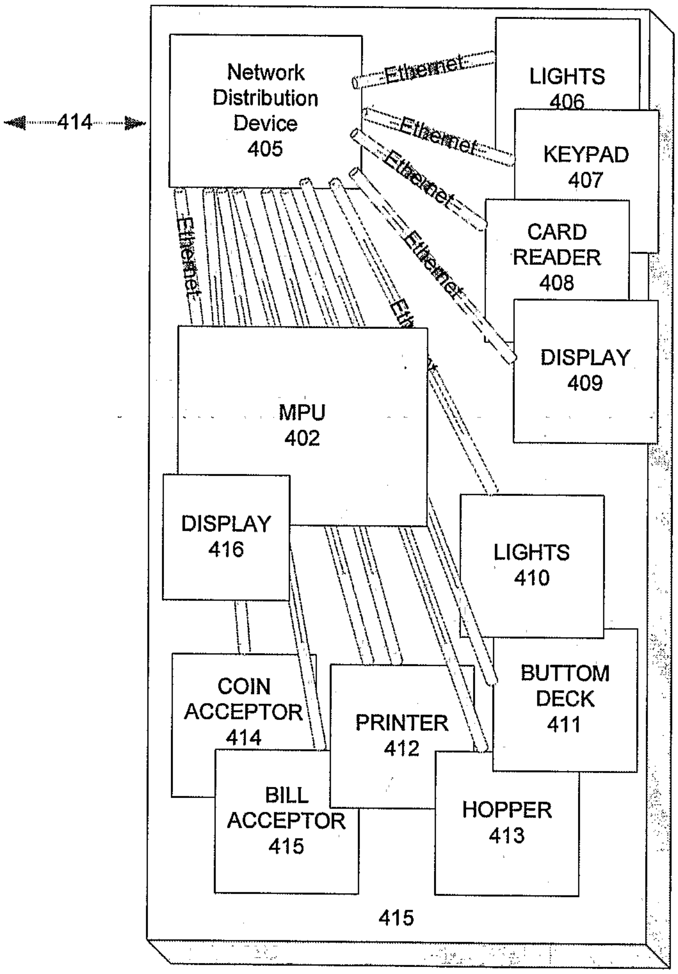

414
Network
Distribution
Device
405
Ethernet
LIGHTS
406
KEYPAD
407
CARD
READER
408
DISPLAY
409
MPU
402
DISPLAY
416
LIGHTS
410
BUTTOM
DECK
411
COIN
ACCEPTOR
414
PRINTER
412
BILL
ACCEPTOR
415
HOPPER
413
415

FIGURE 4B

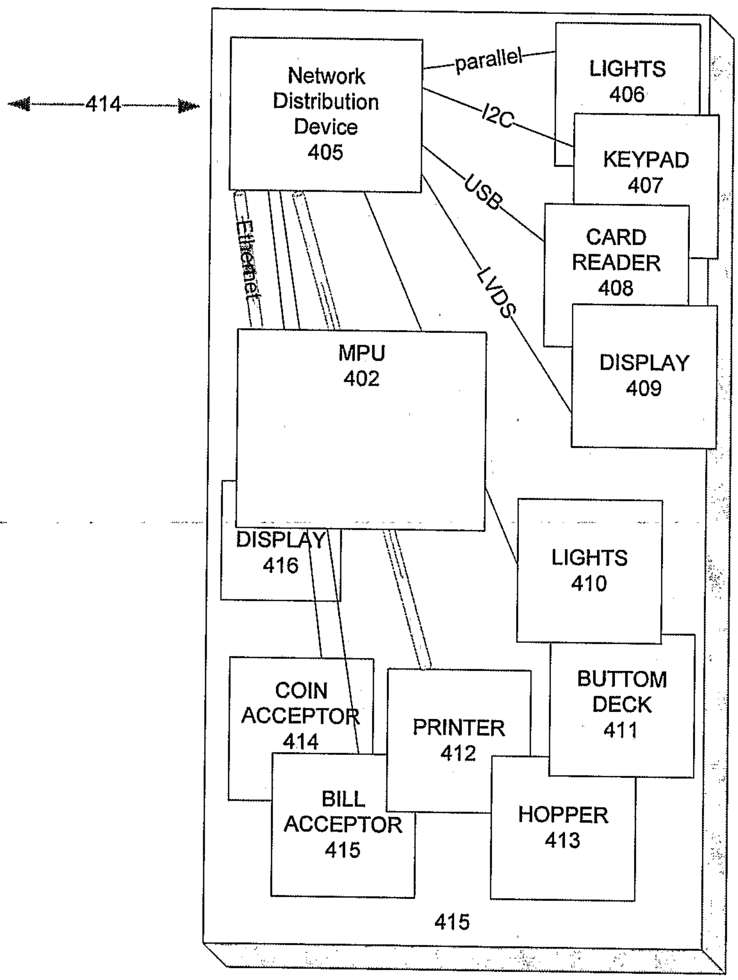

414
Network Distribution Device 405
parallel
I2C
USB
LVDS
Ethernet
LIGHTS 406
KEYPAD 407
CARD READER 408
DISPLAY 409
MPU 402
DISPLAY 416
LIGHTS 410
COIN ACCEPTOR 414
PRINTER 412
BUTTOM DECK 411
BILL ACCEPTOR 415
HOPPER 413
415

FIGURE 4C

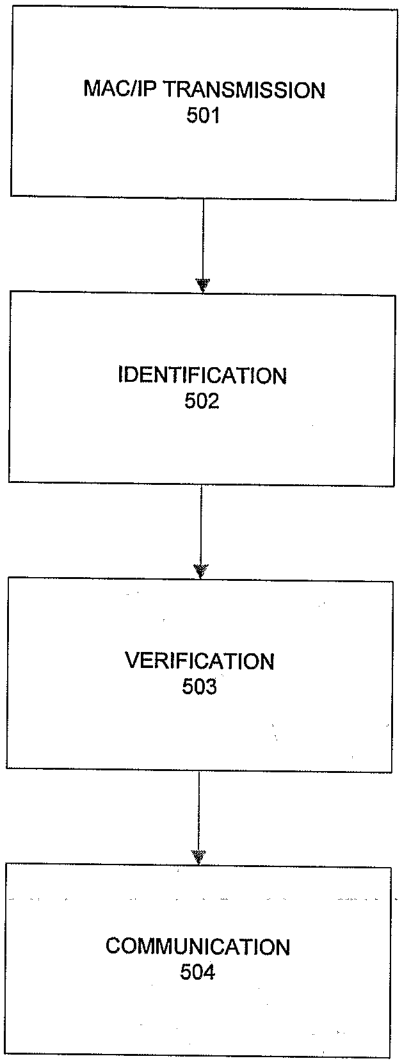

MAC/IP TRANSMISSION
501
IDENTIFICATION
502
VERIFICATION
503
COMMUNICATION
504

FIGURE 5

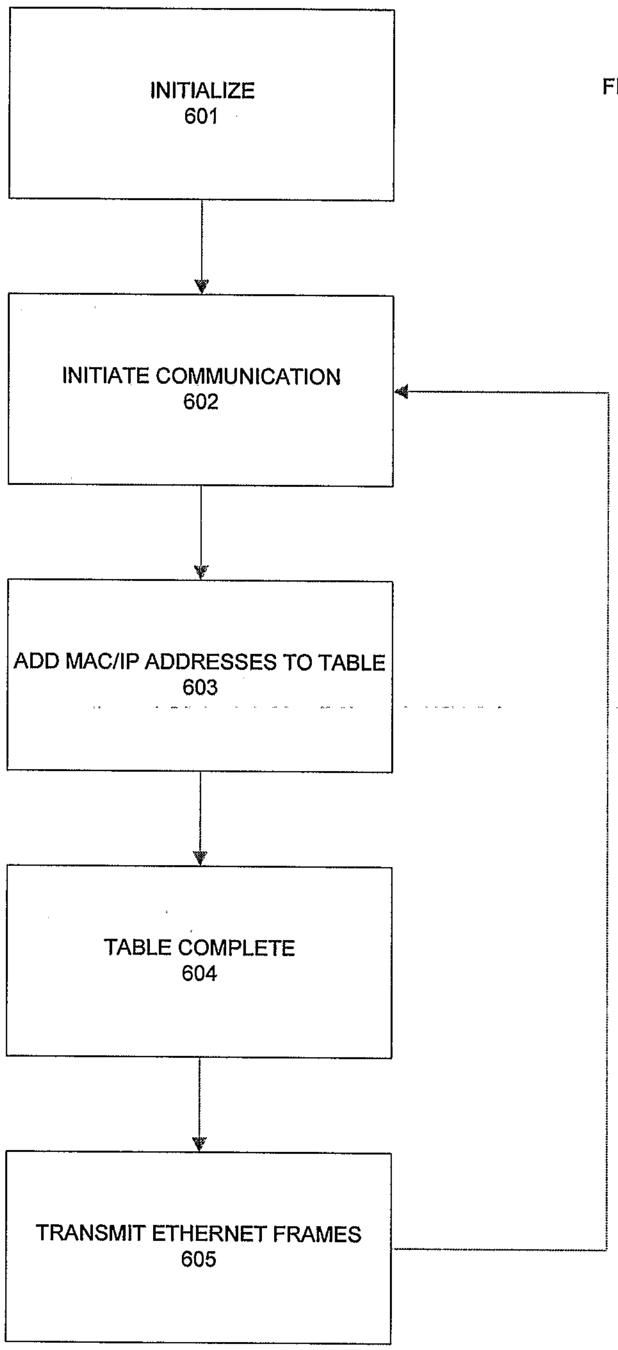

INITIALIZE
601
INITIATE COMMUNICATION
602
ADD MAC/IP ADDRESSES TO TABLE
603
TABLE COMPLETE
604
TRANSMIT ETHERNET FRAMES
605

FIGURE 6

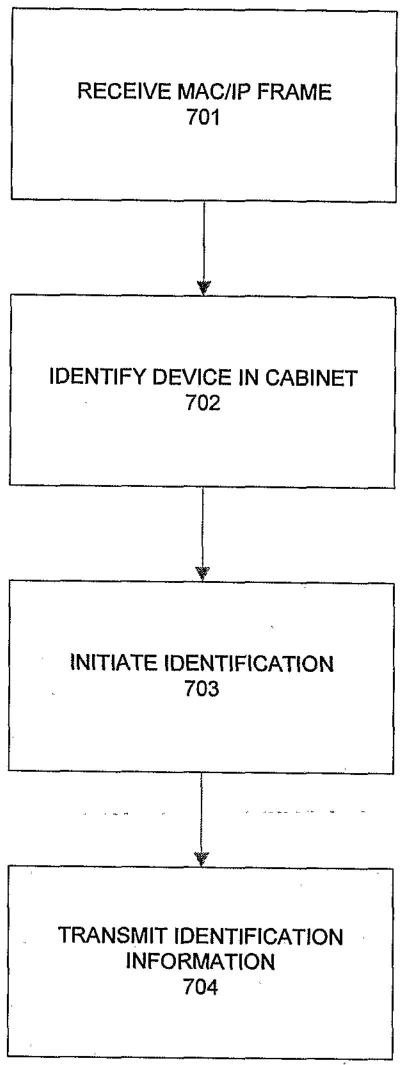

RECEIVE MAC/IP FRAME
701
IDENTIFY DEVICE IN CABINET
702
INITIATE IDENTIFICATION
703
TRANSMIT IDENTIFICATION INFORMATION
704

FIGURE 7

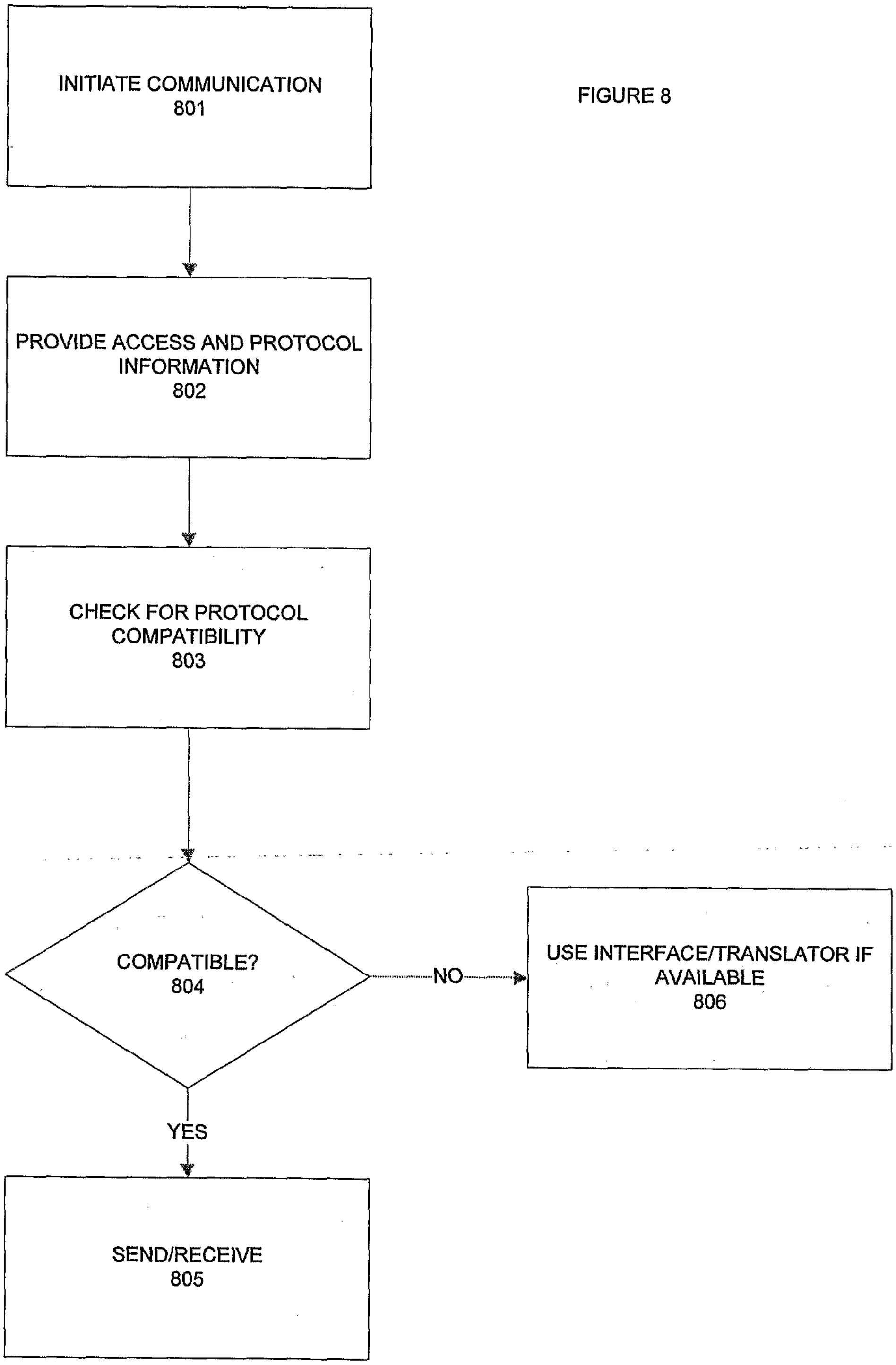

INITIATE COMMUNICATION
801
PROVIDE ACCESS AND PROTOCOL INFORMATION
802
CHECK FOR PROTOCOL COMPATIBILITY
803
COMPATIBLE?
804
NO
USE INTERFACE/TRANSLATOR IF AVAILABLE
806
YES
SEND/RECEIVE
805

FIGURE 8