Office de la Propriété Intellectuelle du Canada
Un organisme d'Industrie Canada

Canadian Intellectual Property Office
An agency of Industry Canada

CA 2980033 A1 2016/09/29

(21) 2 980 033

(12) DEMANDE DE BREVET CANADIEN
CANADIAN PATENT APPLICATION

(13) A1

(86) Date de dépôt PCT/PCT Filing Date: 2016/03/18
(87) Date publication PCT/PCT Publication Date: 2016/09/29
(85) Entrée phase nationale/National Entry: 2017/09/15
(86) N° demande PCT/PCT Application No.: US 2016/023225
(87) N° publication PCT/PCT Publication No.: 2016/154036
(30) Priorité/Priority: 2015/03/25 (US14/668,602)

(51) Cl.Int./Int.Cl. *H04L 29/06* (2006.01)

(71) Demandeur/Applicant:
SIERRA NEVADA CORPORATION, US

(72) Inventeurs/Inventors:
FISCHER, PETER, US;
FELDKAMP, ANDREW, US;
RODRIGUEZ, NELSON, US;
EDWARDS, JOSHUA, US

(74) Agent: SMART & BIGGAR

(54) Titre : SECURITE DE DONNEES BIDIRECTIONNELLE POUR RESEAUX SCADA
(54) Title: BI-DIRECTIONAL DATA SECURITY FOR SUPERVISOR CONTROL AND DATA ACQUISITION NETWORKS

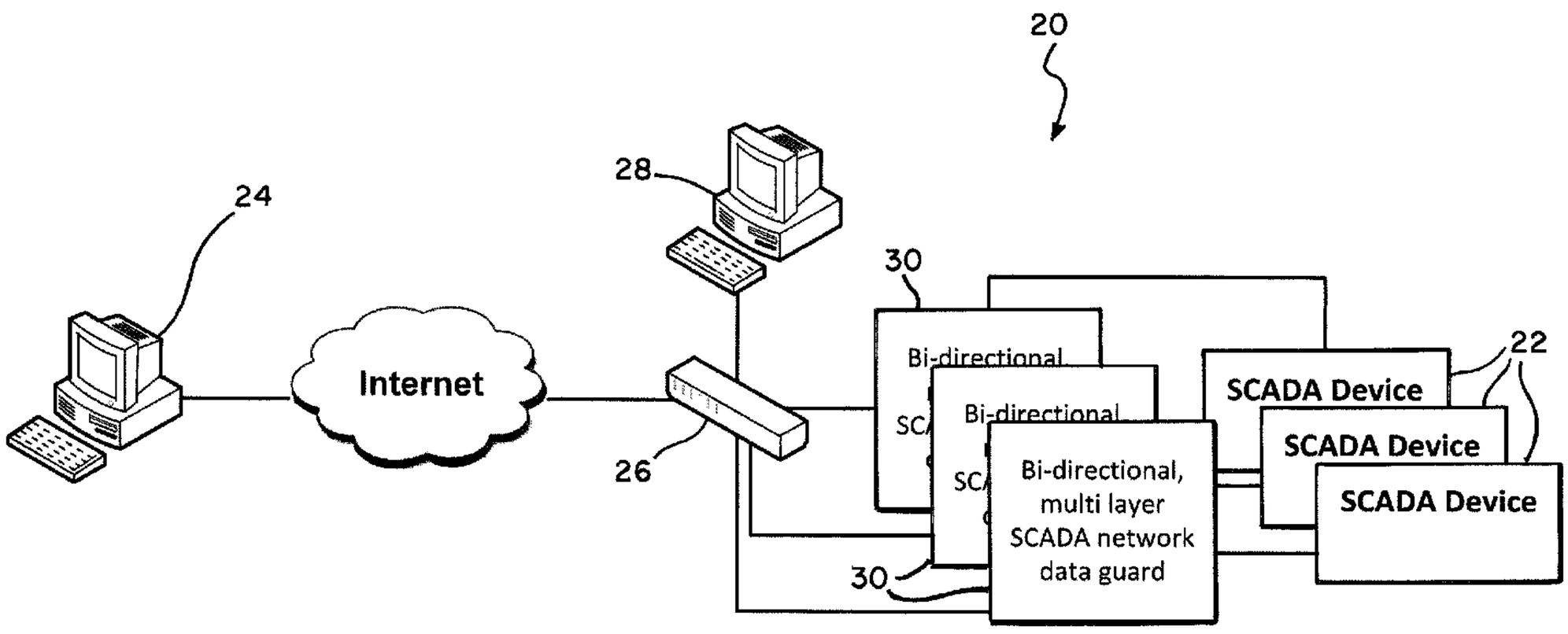

FIG. 2

(57) Abrégé/Abstract:
A cyber-security system, including a device and associated method, provides secure communications bi-directionally between an external network and an internal network, including a supervisor control and data acquisition (SCADA) device. The device includes a processor in data communication with the external and internal networks that is programmed with a rule-set establishing validation criteria configured to validate data received from the external and internal networks. The processor is operable in an operational mode to pass between the external and internal networks only data that are compliant with the validation criteria. The processor may be configured to save certain validated data indicating a system state that can inform the application of the rule-set to data. The processor is re-programmable with a new rule-set only in a programming mode. The device includes a switch that is manually operable to switch the processor from the operational mode to the programming mode.

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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
29 September 2016 (29.09.2016)

WIPO | PCT

(10) International Publication Number
WO 2016/154036 A1

(51) International Patent Classification:
H04L 29/06 (2006.01)

(21) International Application Number:
PCT/US2016/023225

(22) International Filing Date:
18 March 2016 (18.03.2016)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
14/668,602 25 March 2015 (25.03.2015) US

(71) Applicant: **SIERRA NEVADA CORPORATION** [US/US]; 444 Salomon Circle, Sparks, Nevada 89434 (US).

(72) Inventors: **FISCHER, Peter**; 444 Salomon Circle, Sparks, Nevada 89434 (US). **FELDKAMP, Andrew**; 444 Salomon Circle, Sparks, Nevada 89434 (US). **RODRIGUEZ, Nelson**; 444 Salomon Circle, Sparks, Nevada 89434 (US). **EDWARDS, Joshua**; 444 Salomon Circle, Sparks, Nevada 89434 (US).

(74) Agents: **KLEIN, Howard J.** et al.; Klein, O'Neill & Singh, LLP, 16755 Von Karman Avenue, Suite 275, Irvine, California 92606 (US).

(81) Designated States *(unless otherwise indicated, for every kind of national protection available)*: AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JP, KE, KG, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, 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, LR, LS, MW, MZ, NA, RW, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Declarations under Rule 4.17:

— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*

— *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))*

Published:

— *with international search report (Art. 21(3))*

— *with amended claims (Art. 19(1))*

(54) **Title**: BI-DIRECTIONAL DATA SECURITY FOR SUPERVISOR CONTROL AND DATA ACQUISITION NETWORKS

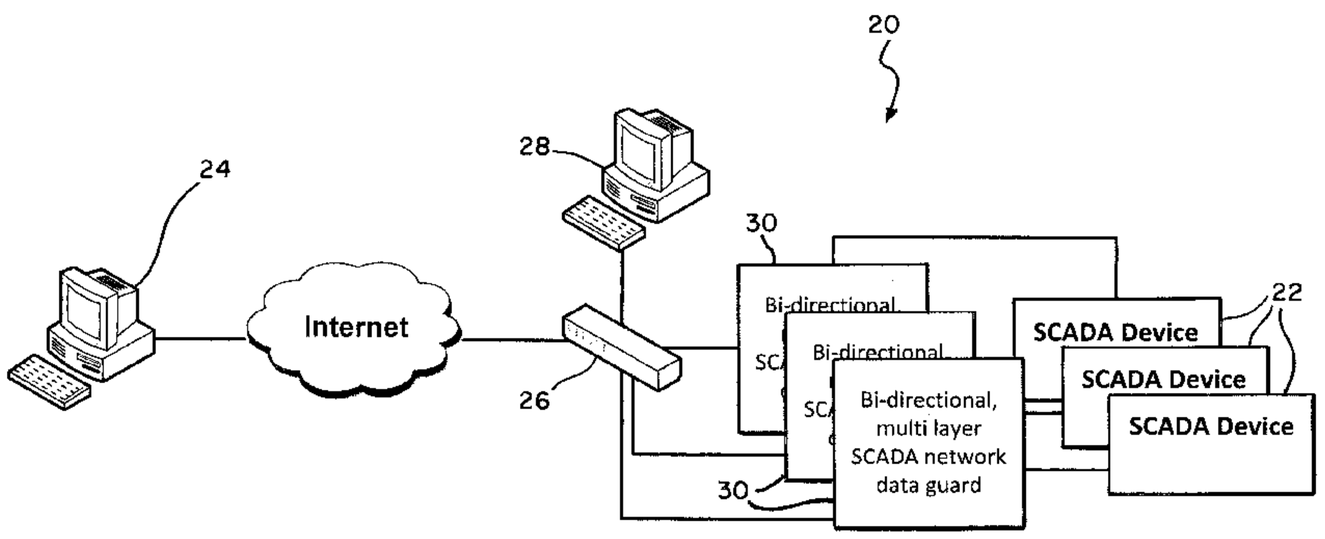

FIG. 2

(57) **Abstract**: A cyber-security system, including a device and associated method, provides secure communications bi-directionally between an external network and an internal network, including a supervisor control and data acquisition (SCADA) device. The device includes a processor in data communication with the external and internal networks that is programmed with a rule-set establishing validation criteria configured to validate data received from the external and internal networks. The processor is operable in an operational mode to pass between the external and internal networks only data that are compliant with the validation criteria. The processor may be configured to save certain validated data indicating a system state that can inform the application of the rule-set to data. The processor is re-programmable with a new rule-set only in a programming mode. The device includes a switch that is manually operable to switch the processor from the operational mode to the programming mode.

WO 2016/154036 A1

BI-DIRECTIONAL DATA SECURITY FOR SUPERVISOR CONTROL AND DATA ACQUISITION NETWORKS

BACKGROUND

[0001] This disclosure relates generally to the field of cybersecurity for automation and control systems that are monitored and/or controlled over public or private internet protocol (IP) networks. More specifically, it relates to devices and methods for providing secure communications to and from one or more supervisor control and data acquisition (SCADA) devices in such automation and control systems.

[0002] Supervisor Control and Data Acquisition (SCADA) devices, such as switches, meters, and sensors, enable real-time monitoring and control of automation and control systems over public or private internet protocol (IP) networks. Most SCADA devices utilize industrial communication protocols, such as Modbus or Distributed Network Protocol V3.0 (DNP3), which are not designed with robust built-in security. Thus, these networked devices may be susceptible to cyber-terrorism and other malicious attacks that can shut down operations and cause damage to physical equipment. Potential attacks include social engineering, malware, buffer overflow, input validation and man-in-the-middle attacks, from internal or external networks, whether connected to private local networks or the public Internet. This poses a serious cyber-security challenge and risk to critical infrastructure in many industries, such as, for example, petroleum refining, natural gas processing, chemical processing, power generation and distribution, water purification, and even financial institutions.

[0003] Figure 1 illustrates a conventional SCADA network 10 of automation and control devices. The network 10 includes a plurality of SCADA devices 12 linked to an external or remote SCADA control terminal 14 by a network communication device 16, such as, for example, a router. Communication between the remote terminal 14 and the communication device 16 may be through the Internet (as shown), or alternatively, through a Local Area Network (LAN) or a Wide Area Network (WAN). A local or internal SCADA control terminal 18 may also be linked to the SCADA devices 12 and the external control terminal 14 by the communication device 16.

[0004] Commercially available options exist for providing a degree of security for networks such as the network 10 shown in Fig. 1. Some options, that operate at the transport

layer or higher, such as firewalls, security proxies, intrusion detection systems, and application layer solutions, do not provide physical network segregation. Physical and data-link layer specific solutions, such as network segregation, do not protect against attacks that originate from the internal\segregated network, and they present additional integration issues for SCADA control networks. None of these solutions are able to provide real-time protection for SCADA devices that can be customized for the industrial process and system for which they are installed.

[0005] Accordingly, a solution has been sought to the problem of ensuring cyber-security for SCADA networks by protecting them from both internal and external attacks and threats. Moreover, it would be advantageous to provide such protection bi-directionally; that is, by protecting both incoming data (data coming into the SCADA devices from the network with which the device is linked) and outgoing data (data communicated from the SCADA devices to the network).

SUMMARY

[0006] Broadly, in accordance with at least some embodiments of this disclosure, a bi-directional cyber-security ("data guard") device for a SCADA network is installed in-line between each SCADA device and a network (either internal or external) to protect each SCADA device from attack, and to validate the integrity of all data and commands sent to each SCADA device and all information sent from each SCADA device to the network for further dissemination. The bi-directional SCADA network cyber-security or "data guard" device provides protection across all seven OSI model layers by installing a physical hardware isolation barrier between each SCADA device and the network, with a customizable rule-set programmed into the device for processing inbound and outbound data (commands and messages). A cyber-security or "data guard" device according to these embodiments is configured to pass only validated data, and to delete and/or block data that do not conform to validation criteria established by the rule-set. The rule-set can be customized for each SCADA device and the control network. Advantageously, in an aspect, separate rule-sets are provided to define how inbound and outbound data are processed, and either deleted/blocked or validated.

[0007] More specifically, a cyber-security device in accordance with embodiments of this disclosure comprises a processor programed with a data validation rule-set (preferably, but not necessarily, separate data validation rule-sets for inbound and outbound data); an external communication interface configured for bi-directional data communication between the

processor and an external network; and an internal communication interface configured for bi-directional data communication between the processor and at least one SCADA device, wherein the data received by the processor via either the external or internal communication interface is either deleted/blocked or passed by the appropriate rule-set, depending on whether the data conform to validation criteria established by the rule-set. The processor analyzes the data, preferably byte-by-byte, with the data in each byte being required to conform to the rule-set validation criteria before being passed from the processor to the appropriate interface. The processor may also be configured to be re-programmed with new rule-sets during start-up ("re-booting"), but only when a programming switch, preferably a physical hardware switch, is actuated during start-up, thereby assuring that the security provided by the rule-set validation criteria cannot be defeated or compromised without physical access to the cyber-security device.

[0008] In accordance with another aspect of the disclosure, a method of validating data transmitted between an internal network including a SCADA device and an external network comprises, in at least some embodiments, (a) providing a processor programmed with a rule-set establishing data validation criteria; (b) communicating data to the processor from one of the internal and external networks; (c) operating the processor to determine if the data conform to the data validation criteria established by the rule-set; and (d) communicating the data from the processor to the other of the internal and external networks only if the data conform to the validation criteria.

BRIEF DESCRIPTION OF THE DRAWINGS

[0009] Figure 1 is a simplified semi-schematic representation of a conventional unsecured SCADA network, as described above under "BACKROUND;"

[0010] Figure 2 is a simplified semi-schematic representation of a SCADA network including cyber-security devices according to the present disclosure;

[0011] Figure 3 is a generalized, high-level flow chart representing the flow of data outward from the SCADA devices to the remote terminal in the network of Fig. 2;

[0012] Figure 4 is a generalized, high-level flow chart representing the flow of data from the remote terminal to one or more of the SCADA devices in the network of Fig. 2;

[0013] Figure 5 is a diagrammatic representation of an embodiment of a cyber-security device in accordance with the present disclosure;

[0014] Figure 6 is a flow chart representing the steps of validating data in accordance with one embodiment of the present disclosure; and

[0015] Figure 7 is a flow chart representing the steps of validating data in accordance with another embodiment of the present disclosure that takes into account the industrial process or system in which the device is installed.

DETAILED DESCRIPTION

[0016] The following detailed description describes the present aspects with reference to the drawings. In the drawings, reference numbers label elements of the present aspects. These reference numbers are reproduced below in connection with the discussion of the corresponding drawing features.

[0017] It will be understood that any of the aspects described with reference to the figures may be implemented using software, firmware, hardware (e.g., fixed logic circuitry), or a combination of these implementations. The terms "logic," "module," "component," "system," and "functionality," as used herein, generally represent software, firmware, hardware, or a combination of these elements. For instance, in the case of a software implementation, the terms "logic," "module," "component," "layer," "system," and "functionality" represent executable instructions that perform specified tasks when executed on a hardware-based processing device or devices (e.g., CPU or CPUs). The program code can be stored in one or more non-transitory, computer readable memory devices.

[0018] More generally, the illustrated separation of logic, modules, components, systems, and functionality into distinct units may reflect an actual physical grouping and allocation of software, firmware, and/or hardware, or it can correspond to a conceptual allocation of different tasks performed by a single software program, firmware program, and/or hardware unit. The illustrated logic, modules, components, systems, and functionality may be located at a single site (e.g., as implemented by a processing device), or may be distributed over a plurality of locations. The term "machine-readable media" and the like refers to any kind of medium for retaining information in any form, including various kinds of storage devices (magnetic, optical, static, etc.).

[0019] The aspects disclosed herein may be implemented as a computer process (method), a computing system, or as an article of manufacture, such as a computer program product or computer-readable media. The computer program product may be embodied or

implemented as non-transitory, computer storage media, readable by a computer device, and encoding a computer program of instructions for executing a computer process. The computer program product may also be readable by a computing system, and encoding a computer program of instructions for executing a computer process.

[0020] Figure 2 shows a SCADA network 20 that includes a plurality of SCADA devices 22 linked to an external or remote SCADA control terminal 24 by a network communication device 26, such as, for example, a router. Communication between the remote terminal 24 and the communication device 26 may be through the Internet (as shown), or alternatively, through a Local Area Network (LAN) or a Wide Area Network (WAN). A local or internal SCADA control terminal 28 may also be linked to the SCADA devices 22 and the external control terminal 24 by the communication device 26.

[0021] The network 20 differs from the network 10 of Fig. 1, as described above, primarily by the provision of one or more cyber-security devices or "data guards" 30, in accordance with this disclosure. The data guards 30, which will be described in detail below, are installed between the SCADA devices 22 and the communication device 26, whereby all data communicated to or from each SCADA device passes through, and is processed by, one data guard 30, as will be described below. Preferably, a single data guard device 30 is operationally associated with each SCADA device 22, so that each SCADA device 22 has its own dedicated data guard device 30, but other configurations may be suitable, depending on the particular application. Thus, a network may be configured, for example, with one data guard device 30 operationally associated with two or more SCADA devices 22. In an aspect, the data guard device(s) 30 is (are) transparent to the network and would not require changes to or be detected in the network's normal operation.

[0022] In an aspect, a data guard 30 includes a programmable rule-set that provides one or more rules for handling network messages that are sent through it. In an aspect, a rule-set comprises a configuration file loaded into a specific location in memory or device storage. Accordingly, the rule-set can be customized for each SCADA device and the control network with which the data guard is associated. A rule-set may include static rules, such as "allow" or "deny" based on the message protocol and content. A rule-set may also include dynamic rules that use variables that can be assigned to message data fields or registers. In this manner, large numbers of "static rules" may be condensed into smaller rule-sets. For example, a dynamic rule

may allow messages over a range of message data addresses or command sequences only during certain times of day and another range of message data addresses or command sequences during a different time of day. The data guard 30 is configured, in an aspect, to allow only limited changes to its configuration, such as the rule-set and network settings. In an aspect, all other data locations may be restricted from change to provide more security to the operation of the data guard 30.

[0023] Figures 3 and 4 show generally the flow of data into and out of a data guard device 30. In Fig. 3, outbound data (information and/or messages) are generated (step 101) by one of the SCADA devices 22. The data may represent status information and/or messages acquired by the SCADA device 22 from the automation and control system (not shown) in which the SCADA device 22 is installed. The SCADA-generated data are received by the data guard device 30 (step 102) through a protected internal communication interface, preferably an Ethernet interface. The data are then processed (step 103) by validation software in the data guard device 30, first to screen out any data deemed invalid or malformed ("unqualified" data) based on a rule-set in accordance with a predefined SCADA protocol that is established by the security software installed in the data guard device 30. Any unqualified data are blocked or deleted (dropped), and a corresponding event log entry may advantageously be created. Well-formed ("qualified") data are then validated against the rule-set established by the security software to assure compliance with the data validation criteria established by the rule-set. Any data that are not in compliance with the validation criteria are deleted or blocked, and an event log entry may advantageously be created. Only data that are validated by compliance with the validation criteria are passed (step 104) to a protected external communication interface (e.g., an Ethernet interface) and then to a control network (e.g., a LAN or WAN), and finally to one or more designated recipients (step 105), which may include the remote SCADA control terminal 24 and, if present, the local SCADA control terminal 28.

[0024] The processing of step 103 may also serve another purpose with respect to SCADA sent messages. In some aspects, the messages may be scanned to understand the state of the SCADA device that sent the message. In general, in some aspects, a SCADA device may operate as a state machine, meaning that it must be in one of a finite number of conditions or states at any given time. In such a system, the SCADA device operates under a particular set of rules allowed in that current state until conditions cause the device to transition to a new state. In

that new state, a different set of rules may apply. As such, in an aspect, the data guard device may use data from within SCADA device messages to understand the current state and to process network messages, such as commands, accordingly.

[0025] For example, certain messages may communicate the values of variables associated with the SCADA device. When these variables are deemed important, the data guard device 30 may maintain and update a copy of that variable for use in processing other messages. These state variables may be stored in registers, volatile and/or non-volatile memory, or the like. In one aspect, it is preferred that any state variable data be updated only after the message is qualified and validated. In other aspects, however, state variable data may be processed simultaneously with other processing or in another order. In an aspect, therefore, the state-based rule-sets operate at the OSI layer 7 (application layer) by looking at the value of variables indicating the device or system state.

[0026] As will be described in more detail below with reference to Figure 7, the data guard device 30 may seek to store this state data in order to factor in the state of a device when processing and validating messages, as some messages may be allowed or denied only in specific situations. Updating the state variables based on processing conforming messages allows the data guard to remain transparent in the sending and receiving of messages. Typically in such systems, the SCADA device 22 is being polled by one or both of the remote SCADA control terminal 24 and the local SCADA control terminal 28 at regular intervals during normal operation, whereby the data guard 30 is likely to have relatively accurate state information simply from reading the contents of the responses of the SCADA device 22. In another aspect, a data guard 30 may poll a SCADA device 22 or other connected device for its state by generating its own data request messages. In some cases, this is a less desirable—but still functional—method of operation.

[0027] In Fig. 4, inbound data, from, for example, the remote SCADA control terminal 24 and/or the local SCADA control terminal 28 (step 106) are received by the protected external communication interface (step 107) and then processed (step 108) as described above with reference to Fig. 3. Only data that are validated by compliance with the validation criteria are passed (step 109) to the protected internal communication interface (described below) and then to the designated SCADA device 22 via an appropriately configured network connection (step 110).

[0028] While the state of one or more SCADA devices 22 is more likely to be of importance in processing messages through a data guard device 30 than the state of a device on the communication device 26 side (or "public" side) of the data guard 30, state variable data information may also be processed from messages directed to a SCADA device 22 in a manner as described above with respect to Figure 3.

[0029] Figure 5 schematically illustrates an instantiation of the cyber-security or "data guard" device 30 as installed in an automation and control network. In an aspect, as illustrated, the device 30 includes at least one processor 32 that communicates with an external network 34 by means of an external communication interface, e.g., an external Ethernet interface 36, and with an internal network 38 by means of an internal communication interface, e.g., an internal Ethernet interface 40. In an aspect, the processor 32 may be understood as including memory 33 and non-volatile storage 35 with which a processor module communicates. For example, either or both of the memory 33 and the non-volatile storage 35 may optionally be included with the processor 32 in a microcomputer, as is well known, or they may be separate components. In an aspect, power is provided by a DC power supply 42 that is connectable to any suitable external power source 44 that delivers electrical power at, for example, 5 to 32 VDC, and that converts the voltage from the source 44 to a voltage suitable for operating the electronic components of the device 30. Ethernet is a common network protocol used for network communication. The original Ethernet bus or star topology was developed for LANs to transfer data at 10 Mbps (megabits per second). Newer Ethernet standards (for example, Fast Ethernet (100 Base-T) and Gigabit Ethernet) support data transfer rates that are greater than 1 gigabit (Gb). The various aspects described herein may use Ethernet (which includes 100 Base-T and/or Gigabit Ethernet) as the network protocol. However, the adaptive aspects disclosed herein are not limited to any particular protocol, as long as the functional goals are met by an existing or new network protocol. For example, Fibre Channel (FC) or Fibre Channel over Ethernet (FCoE) or DNP3, as mentioned above, are other communications protocols that may also be used, among others, in various aspects.

[0030] The one or more processors 32, also known as central processing units (CPUs), may be, or may include, one or more programmable general-purpose or special-purpose microprocessors, digital signal processors (DSPs), programmable controllers, application specific integrated circuits (ASICs), programmable logic devices (PLDs), or the like, or a

combination of such hardware devices. In an aspect, the processor 32, non-volatile storage 35 and/or memory 33 may be combined in a system-on-a-chip (SoC) configuration, such as those commercially available based on ARM or x86 designs. In other aspects, memory 33 and/or storage 35 may be separate components.

[0031] Each of the processors 32 executes machine-implemented instructions (or process steps/blocks) out of memory 33. In an aspect, processor 32 communicates with the other components through one or more interconnects (unlabeled) that may be referred to as a computer bus or set of computer buses, as is well-known. A computer bus may be, for example, a system bus, a Peripheral Component Interconnect (PCI) bus, a PCI-Express (PCIe) bus, a HyperTransport or industry standard architecture (ISA) bus, a SCSI bus, a universal serial bus (USB), an Institute of Electrical and Electronics Engineers (IEEE) standard 1394 bus (sometimes referred to as "Firewire"), or any other type of bus. It is preferable that each processor 32 sits between separate buses to connect to the external Ethernet interface 36 and the internal Ethernet interface 40, such that the processor cannot be bypassed by any direct path between the external network 34 and internal network 38.

[0032] The storage device 35, which may be or include, for example, a hard disk (HDD), a CD-ROM, a non-volatile memory device such as flash, a hybrid drive (sometimes referred to as SSHD), or any other storage device for storing persistent, structured or unstructured data. Storage 35 may store operating system program files (or data containers), application program files, and one or more rule-sets in the form of scripts, functions, programs, configuration files or other file types. In an aspect, storage 35 may also include a data file or data structure that maintains indications of device states as described herein.

[0033] Memory 33 also interfaces with the processor(s) 32 with access to memory storage. Memory 33 may include any suitable type of random access main memory (RAM) for example. When executing stored computer-executable process steps from storage 35, the processor(s) 32 may store and execute the process steps out of memory 33. Read only memory (ROM, not shown) may also be used to store invariant instruction sequences, such as startup instruction sequences or basic input/output system (BIOS) sequences for operation of a keyboard (not shown). In an aspect, memory 33 may include a data structure storing device state indications as described herein for use when processing messages with a state-varied rule-set, as described below with reference to Figure 7.

[0034] As discussed above, each processor 32 is programmable with a rule-set that validates both inbound data that is received from the external Ethernet interface 36, and outbound data that is received from the internal Ethernet interface 40. In an aspect, as mentioned above, inbound and outbound data are processed by their respective rule-sets that may be simultaneously or separately programmed into the processor(s) 32. Programming and re-programming are accomplished via the Ethernet through the external Ethernet interface 36 only when the device 30 is in a programming mode initiated by the activation of a programming switch 46 during boot-up of the processor(s) 32. In another aspect, programming and reprogramming are accomplished via the internal Ethernet interface 40—also when the device 30 is in a programming mode initiated by the activation of a programming switch 46. In a preferred embodiment, the programming switch 46 is a physical (i.e., hardware) switch that can be actuated manually. For example, the programming switch 46 may comprise a button, lever, plunger, blade, or the like that can be accessed by a tool (not shown) inserted through an aperture in the housing (not shown) containing the electronic components. In other aspects, the programming switch may include a fingerprint scanner or other biometric security device. Thus, any alteration, whether benign or malicious, of the operational software of the device 30 preferably requires physical access to the device 30.

[0035] Initiation of the programming mode allows the processor(s) 32 to upload a digitally-authenticated rule-set file received in an encrypted programming signal. In an aspect, digital authentication may occur through the use of public and private keys. For example, when a data guard device is built or set up initially, the process may include burning in a public key. Preferably this key is located in ROM or other memory that cannot be overwritten. In an aspect, updating a rule-set then may require both knowledge of an associated private key to complement the public key and physical access to the data guard 30 and its programming switch 46. If a rule-set is uploaded without the correct private key signature, the device 30 may generate an error, abort the upload process, delete the attempted rule-set upload, and/or the like.

[0036] After uploading the rule-set file, in an aspect, the device 30 is allowed to go through a complete power cycle to enter its operational mode as programmed with the new rule-set. A status indicator 48 (preferably a visual indicator such as an LED) may optionally be employed to indicate whether the device 30 is in the programming mode or the operational mode.

[0037] As can be seen from Figure 5, preferably the external Ethernet interface 36 and the internal Ethernet interface 40 are physically and electrically isolated from each other, and can communicate with each other only through the processor(s) 32. This assures that data cannot pass to or from the SCADA devices 22 on the internal network 38 without being validated by the rule-set(s) programmed into the processor(s) 32, thereby providing data security that encompasses all seven OSI model layers (physical, data link, network, transport, session, presentation, and application).

[0038] The functional components of the data guard device 30, as described above, are housed in an enclosure (not shown) that is advantageously made of a suitable metal alloy, such as, for example, aircraft grade 6061-T6 aluminum alloy. The above-described electronic components can be advantageously potted or otherwise protected to provide a certain level of tamper protection.

[0039] Figure 6 is a data flow diagram for exemplary data validation software used in an embodiment of the cyber-security device 30. A data message is read (step 201) from a first input/output (I/O) port operationally associated with either the external Ethernet interface 36 or the internal Ethernet interface 40, depending on whether the data message is inbound or outbound. After the data inputted to the I/O port are read, the data are queued for processing (step 202), and then qualified (step 203) (preferably byte-by-byte) by a rule-set, as described above, to determine the presence of malformed or unexpected data ("unqualified" data). If any unqualified data are found ("YES" in decision step 204), such data are deleted, and a log entry is created (step 205). If no such unqualified data are found ("NO" in decision step 204), the content of the qualified data is examined in accordance with the rule-set (step 206) to determine compliance with the validation criteria. If non-compliance is determined ("NO" in decision step 207), the data are deleted, and a log entry is created (step 208). If the data are found to be compliant with the validation criteria, i.e., the data are determined to be valid ("YES" in decision step 207), the data may optionally be modified (as needed) in accordance with any further criteria that may be established by the rule-set (step 209), then written (step 210) on a second I/O port operationally associated with whichever of the Ethernet interfaces 36, 40 was not used in the reading step (step 201), and finally output from the I/O port to the appropriate Ethernet interface.

[0040] An exemplary rule-set that may be used in some embodiments of this disclosure may be generically described as including the following logical processes operating on message data read from an I/O port:

[0041] In the first process, the message header is read to determine and verify the message type and the expected message header length and version to validate integrity of the message. This process includes (a) reading the Start of Message byte sequence where applicable; (b) reading N bytes (where N is the number of bytes defined in the rule-set for that message format), indicating the start of the message that comprises the header; and (c) verifying that the header is valid, that there are no illegal values or extra characters in the message, and that all required fields are present and match requirements defined in the rule-set. In another aspect, ASCII messages such as XML may be processed—in a first process—by verifying delimiters or custom delimiters or message criteria to determine message validity as described above.

[0042] In an aspect, the first process also advantageously includes comparing the total size of the data read to the message packet size specified in the header to assure that no extra data have been inserted and that no potential data overflows are possible.

[0043] In the second process, the contents of the message payload data are looped through to assure that only allowed fields are present in the message and that they conform to limits defined in the rule-set. This process includes repeating a sequence of sub-steps through the entire contents of the message data payload or until an invalid message is detected, or the total amount of data read matches or exceeds the expected message packet size. The sequence of sub-steps comprises: (1) reading M binary bytes that comprise a data field identifier; (2) reading the value and contents of the data field; (3) assuring that the data field is allowed by rule-set; and (4) if allowed, assure that the values of that data are within limits and ranges defined in the rule-set.

[0044] In the second process, for example, the message data may be processed to determine if the type of message is allowed and whether or not variables within the message are allowed for that message.

[0045] For example, in a particular application, a SCADA device may control water pressure through a given pipe. In an aspect, the SCADA device may allow various commands such as “increase pressure,” “decrease pressure,” “report pressure,” and “emergency stop.” In such an application, a data guard device 30 may be programmed to review incoming messages and determine that they are properly formed, are of the right size, and the like. Furthermore, the

actual contents of the messages may also be analyzed for compliance with a rule-set. For example, if an "increase pressure" message comes in, the data guard device 30 may ensure that no extra data are tacked onto the message that could be interpreted improperly by the SCADA device according to a first process. According to a second process, the data guard device 30 may also determine if variables, such as function parameters, are within operating limits according to the rule-set. For example, in one rule-set, changes in pressure must occur in increments less than 5 PSI. In such a case, if a "decrease pressure" message is processed with a parameter indicating lowering the pressure by 3PSI, it is allowed as within the rule-set. On the other hand, if the "decrease pressure" message included a parameter of 35, it may be dropped as seeking a change that is too great for the system's rule-set.

[0046] The above-described generic rule-set—and the specific example—are exemplary only and are not limiting. Variations and modifications of a rule-set will readily suggest themselves for particular applications. Rule-sets may be based on any of a variety of message processing rules, including message type, message size, message contents, message source, message destination, message protocol, data rate, system state, the data type and values of message contents, and the like. Moreover, rule-sets may allow for variance based on outside input apart from the message contents itself.

[0047] In an aspect, for example, the rule-set can further be programmed to take device state into account when processing messages. While this will often come from the state reported by the SCADA device 22, it can also take into account the states of multiple SCADA devices 22, a SCADA device state, external network device states, internal network device states, combinations or the same, and the like. Figure 7 illustrates a sample data flow process that includes analyzing messages in light of device states.

[0048] As illustrated, in an aspect, a data message is read from an I/O port (step 301). This may comprise a message from an I/O port operationally associated with either an external Ethernet interface 36 or an internal Ethernet interface 40, depending on whether the data message is inbound or outbound. After the data message input to the I/O port is read, the data message is analyzed in a first process (step 302) (preferably byte-by-byte) by a rule-set, as described above, to determine the presence of malformed or unexpected data ("unqualified" data). If any malformed or unqualified data are found ("NO" in decision step 303), the message is deleted, and a log entry may be created (step 309). If the data are well-formed ("YES" in

decision step 303), the process continues to step 304 and step 305. In step 304, system state data are read from memory 33 or storage 35 as needed. As described above, in an aspect, the state data are gleaned from previous messages that are processed through the data guard device 30. In another aspect, the data guard device 30 may further be able to poll connected devices for state information, but this additional network traffic may be less desirable or unnecessary.

[0049] The message or command type of the qualified data are then examined in accordance with the rule-set and in light of the system state data read from memory (step 305) to determine whether the message/command type is allowed at that time. If the message is not allowed at that time ("NO" in decision step 305), the message and data are deleted, and a log entry is created (step 309). If the message is allowed, ("YES" in decision step 305), the content of the allowed message is examined in accordance with the rule-set and the current system state data (step 306) to determine compliance with the validation criteria. If non-compliance is determined ("NO" in decision step 306), the message and data are deleted, and a log entry is created (step 309). If the data are found to be compliant with the validation criteria in light of the system state data, i.e., the message data are determined to be valid ("YES" in decision step 306), the message data may optionally be used to update the system state data stored in the data guard device 30 (step 307). Then the message is written (step 308) to a second I/O port operationally associated with whichever of the Ethernet interfaces 36, 40 was not used in the reading step (step 301), and finally outputted from the I/O port to the appropriate Ethernet interface.

[0050] To return to the example set forth above, a SCADA device controlling water pressure in a pipe has, as a state, the current water pressure. In normal operation, for example, an external control terminal 24 may poll the SCADA device 22 for the water pressure at periodic intervals. This will cause a message to be returned from the SCADA device 22 that includes the current water pressure reading. As the data guard device 30 processes this return message, it may make a copy of the current pressure reading as a part of the current system state. The rule-set may then be set up to allow or deny messages based on the water pressure state variable. For example, a rule-set may include a rule that indicates "increase pressure" or "decrease pressure" messages are only acceptable when the current pressure is outside of a normal pressure range, such as between 70 and 120 PSI. When a proper "report pressure" message is sent through the data guard device 30 to the SCADA device 22, the data guard device 30 passes along the request and receives the response message from the SCADA device 22. In an aspect, the data guard

device 30 may process the response message and save the current reading for PSI that is being reported—for example, 75 PSI. If the data guard device 30 then receives an "increase pressure" message, it can review the current pressure state reading and deny the message.

[0051] Any of a large number of systems, states, and rule-sets are contemplated herein, and the water pressure example is simply one such possible application. One will understand from the disclosure herein that the data guard systems and methods may be implemented in a variety of applications and situations, such as, for example, industrial control systems, energy management and distribution systems, remote monitoring systems, and the like. Other concrete examples include power stations, oil pipelines, and building HVAC, alarm, fire and other safety systems.

[0052] From the foregoing description, it will be appreciated that data guard device 30 cannot be configured or otherwise modified by users over an internal or external network without physical access to the device (due to needing access to the programming switch 46). Therefore, the security provided by the data guard device 30 cannot be overridden or by-passed, even if other protections, such as a firewall or IDS, are compromised.

[0053] Significantly, the data guard device 30 provides bi-directional protection across all seven OSI model layers in an aspect. This is achieved through the use of two segregated network interfaces providing physical and data-link layer protection between each SCADA device 22 and the control network. Furthermore, the data guard device 30 protects the network and transport OSI model layers by limiting network data traffic to only the configured IP addresses and ports to and from each individual SCADA device. In addition, the data guard device 30 protects the session, presentation, and application OSI model layers through data validation and rule-sets that define what data can be sent to and from each SCADA device 22 based at least in part on the data content of network traffic. Additionally, these layers can also be protected through encryption, which is supported in at least some aspects. Moreover, the data guard device 30 does not modify the message protocol, utilize a proxy, or require any modification to existing software or hardware on the SCADA network. Finally, the re-programmable feature described above allows the data guard device 30 to support custom rule-sets and configurations to tailor it to any SCADA device and network.

[0054] Although the present disclosure has been described with reference to specific aspects, these aspects are illustrative only and not limiting. For example, although the description

above has been described with respect to a data guard device, any other device may be configured to perform the foregoing function. In an aspect, for example, data guard functionality may be built into a SCADA device. Many other applications and aspects of the present disclosure will be apparent in light of this disclosure and the following claims. References throughout this specification to "one aspect" or "an aspect" means that a particular feature, structure or characteristic described in connection with the aspect is included in at least one aspect of the present disclosure. Therefore, it is emphasized and should be appreciated that two or more references to "an aspect" or "one aspect" or "an alternative aspect" in various portions of this specification are not necessarily all referring to the same aspect. Furthermore, the particular features, structures or characteristics being referred to may be combined as suitable in one or more aspects of the disclosure, as will be recognized by those of ordinary skill in the art

AMENDED CLAIMS

received by the International Bureau on 4 August 2016 (04.08.2016)

1. (Cancelled)

2. The cyber-security device of claim 42, further comprising memory configured for storing system state data derived from validated data messages received from one or both of the external network and the internal network.

3. (Cancelled)

4. The cyber-security device of claim 2, further comprising a status indicator configured to provide an indication of whether the processor is in the operational mode or in the programming mode.

5. The cyber-security device of claim 2, wherein at least one of the external communication interface and the internal communication interface is an Ethernet interface.

6. The cyber-security device of claim 2, wherein the processor is configured to store the system state data based on data content received from one of the internal network and the external network without having to request the data content.

Claims 7-9: (Cancelled)

10. The method of claim 46, further comprising re-programming the processor with a new rule-set only when the processor is in the programming mode

11. The method of claim 10, wherein the re-programming is performed by:

switching the processor from the operational mode to the programming mode;

loading a new rule-set into the processor; and

cycling the processor back to the operational mode.

12. The method of claim 11, wherein the loading is performed via an Ethernet interface.

13. The method of claim 11, wherein the switching is performed manually with a hardware switch operably associated with the processor.

Claims 14-17: (Cancelled)

18. The system of claim 51, wherein the processor is in data communication with the external network via an external Ethernet interface, and wherein the processor is in data communication with the internal network via an internal Ethernet interface.

19. (Cancelled)

20. The system of claim 18, wherein the processor is re-programmable via the external Ethernet interface.

21. The system of claim 51, wherein the processor is configured to cycle back to the operational mode after being switched to the programming mode.

22. The system of claim 51, further comprising a power supply operatively associated with the processor and configured for connection to an external power source providing a DC voltage within a predetermined voltage range, and for supplying the processor with a fixed operational DC voltage converted from the voltage provided by the external power source.

23. The system of claim 22, wherein the predetermined voltage range is 5 to 32 volts.

24. The system of claim 51, further comprising a status indicator configured to provide an indication of whether the processor is in the operational mode or in the programming mode.

25. (Cancelled)

26. A cyber-security device for providing secure communication of automation and control data between first and second networks in a system operable in one or more system states, the cyber-security device comprising:

a first network interface configured to accept messages destined for a supervisor control and data acquisition (SCADA) device or an automation and control device in the second network and transmitting qualified and validated messages to the first network;

a second network interface configured to accept messages destined for the first network and transmitting qualified and validated messages to the second network;

a processor configured to qualify and validate messages from the first and second network interfaces on a byte-by-byte basis; and

memory configured to store system state information and a rule-set comprising rules for qualifying and validating messages on a byte-by-byte basis, wherein at least one of the rules is a system state-dependent rule;

wherein the processor is operable to:

accept a message from the first network interface;

retrieve the rule-set and the system state information from memory;

qualify the message from the first network interface, on a byte-by-byte basis, based on the rule-set and the system state information;

only if the message is qualified, validate the message, on a byte-by-byte basis, based on the rule-set and the system state information; and

transmit the message to the second network interface only if the message is validated and in compliance with the system state-dependent rule.

27. The cyber-security device of Claim 26, wherein the processor is further operable to drop the message when the message cannot be validated and to create an error log entry based on the dropped message.

28. The cyber-security device of Claim 26 wherein the processor is further operable to update the system state when the processor receives a message including system state data that the processor can validate based on the rule-set.

29. The cyber-security device of Claim 28, wherein the message including system state data is accepted at the second network interface and comes from the SCADA device or the automation and control device in the second network.

30. The cyber-security device of Claim 26, further comprising:

a physical switch operable manually to switch the processor between at least an operational mode and a programming mode, wherein the processor is operable to replace the rule-set with a new rule-set only when in the programming mode.

31. The cyber-security device of Claim 30, wherein the physical switch is selected from the group consisting of at least one of a button, a switch, a pin, a lever, a plunger, a blade, and a fingerprint scanner.

32. A method of providing secure communications to a supervisor control and data acquisition (SCADA) device or an automation and control device, the method comprising:

accepting messages bound for the SCADA device or the automation and control device at a network interface that is in data communication with a processor programmed with a programmable rule-set that includes rules for qualifying the accepted messages for message size and type, and for validating message contents in the accepted messages;

processing each accepted message bound for the SCADA device or the automation and control device by operating the processor to implement the rule-set so as to qualify and validate, on a byte-by-byte basis, each accepted message bound for the SCADA device or the automation and control device in accordance with the rules for message type, message size, and message contents;

sending only the messages that are qualified and validated based on the rule-set to the SCADA device or the automation and control device; and

dropping messages that cannot be qualified and validated based on the rule-set.

33. The method of Claim 32, further comprising:

accepting messages from the SCADA device or the automation and control device at a second network interface that is in data communication with the processor;

processing each accepted message from the SCADA device or the automation and control device by operating the processor to implement the rule-set so as to qualify and validate, on a byte-by-byte basis, each accepted message from the SCADA device or the automation and control device in accordance with the rules for message type, message size, and message contents;

sending only the messages that are qualified and validated based on the rule-set to the network interface for transmission to the network; and

dropping messages that cannot be qualified and validated based on the rule-set.

34. The method of Claim 32, further comprising:

updating a system state indication based on the contents of a qualified and validated message from the SCADA device or the automation and control device;

wherein at least one of the rule-set's validation rules is dependent on the system state indication.

35. The method of Claim 32, wherein the rule-set is a first rule-set, the method further comprising:

accepting an input from a physical switch to put the processor into a programming mode;

accepting a second rule-set at the network interface only while the processor is in the programming mode;

replacing the first rule-set with the second rule-set;

exiting the programming mode; and

processing future messages bound for the SCADA device or the automation and control device based on the second rule-set by operating the processor to implement the second rule set so as to qualify and validate, on a byte-by-byte basis, each accepted massage bound for the SCADA device or the automation and control device in accordance with the rules for message type, message size, and message contents.

36. The method of Claim 32, wherein the validation provided by the rule-set is dependent on a system state.

37. A non-transitory computer-readable medium including instructions that, when executed by a processor, cause the processor to:

accept messages bound for a SCADA device or an automation and control device from a network interface;

process, on a byte-by-byte basis, each accepted message bound for the SCADA device or the automation and control device based on a rule-set that includes qualification rules for message type and message size, and validation rules for message contents;

send only the messages that are qualified and validated based on the processing using the rule-set to the SCADA device or the automation and control device; and

drop messages that cannot be qualified or that cannot be validated based on the processing using the rule-set.

38. The non-transitory computer-readable medium of Claim 37, further comprising instructions to:

accept messages from the SCADA device or the automation and control device from a second network interface;

process, on a byte-by-byte basis, each accepted message from the SCADA device or the automation and control device based on the rule-set;

send only the messages that are qualified and validated based on the processing using the rule-set to the network interface for transmission to the network; and

drop messages that cannot be qualified or that cannot be validated based on the processing using the rule-set.

39. The non-transitory computer-readable medium of Claim 37, further comprising instructions to:

update a system state indication based on the contents of a validated message from the SCADA device or the automation and control device;

wherein at least one of the rule-set's validation rules is dependent on the system state indication.

40. The non-transitory computer-readable medium of Claim 37, wherein the rule-set is a first rule-set, the medium further comprising instructions to:

accept an input from a physical switch to put the processor into a programming mode;

accept a new rule-set at the network interface only while the processor is in the programming mode;

replace the first rule-set with a second rule-set;

exit the programming mode; and

process future messages bound for the SCADA device or the automation and control device based on the second rule-set by operating the processor to implement the second rule set so as to qualify and validate, on a byte-by-byte basis, each accepted massage bound for the SCADA device in accordance or the automation and control device with the rules for message type, message size, and message contents.

41. The non-transitory computer-readable medium of Claim 37, wherein the validation of the messages by the rule-set is dependent on a system state.

42. A cyber-security device for providing secure data communication of SCADA or automation protocol data messages between an external network and an internal network of an automation and control system operable in one or more system states, wherein the data messages include system state information, at least one of the internal and external networks including at least one of a SCADA device and an automation and control device, the cyber-security device comprising:

an external communication interface configured to send data messages to, and receive data messages from, the external network;

an internal communication interface configured to send data messages to, and receive data messages from, the internal network;

a processor in communication with the external communication interface and the internal communication interface, the processor being operable in an operational mode to process the data messages in accordance with a processor-implemented rule-set including a system state-dependent rule, the rule set being configured for (a) qualifying, byte-by-byte, (i) the content of each of the data messages received from the external network via the external communication interface as conforming to qualification criteria defined by the rule-set for external-to-internal

communications, and as including system state information in compliance with the system state-dependent rule, and (ii) the content of each of the data messages received from the internal network via the internal communication interface as conforming to qualification criteria defined by the rule-set for internal-to-external communications, and as including system state information in compliance with the system state-dependent rule; and (b) validating, byte-by-byte, the content of each qualified data message in accordance with validation criteria defined by the rule-set including the system state-dependent rule, wherein the processor has a programming mode in which the processor is re-programmable with a new rule-set; and

a manually-operable physical switch operable to transition the processor between at least the operational mode and the programming mode;

43. The cyber security device of claim 42, wherein the processor-implemented rule set includes a first rule set configured to process the data messages received from the external network, and a second rule set configured to process the data messages received from the internal network.

44. A method for providing secure communication of supervisor control and data acquisition (SCADA) data messages or automation protocol data messages, wherein the data messages have content including system state information in a system comprising an external network and an internal network, at last one of the internal and external networks including at least one of a SCADA device and an automation and control device, wherein the system is operable in any of several system states, the method comprising:

providing a processor programmed with a processor-implemented rule-set configured for qualification and validation of the content of the data messages, the rule-set defining data qualification and validation criteria, the rule-set including a system state-dependent rule;

accepting incoming data messages from the external network into the processor;

determining a current system state of the system;

qualifying, on a byte-by-byte basis, the content of each incoming data message received from the external network by compliance with the data qualification criteria defined by the rule-set and compliance with the system state-dependent rule;

validating, on a byte-by-byte basis, the content of each qualified incoming data message by compliance with the data validation criteria defined by the rule-set and compliance with the system state-dependent rule;

outputting from the processor to the internal network only those incoming data messages the content of which has been qualified and validated;

accepting outgoing data messages from the internal network into the processor;

qualifying, on a byte-by-byte basis, the content of each outgoing data message received from the internal network by compliance with the data qualification criteria defined by the rule-set and compliance with the system state-dependent rule;

validating, on a byte-by-byte basis, the content of each qualified outgoing data message by compliance with the data validation criteria defined by the rule-set and compliance with the system state-dependent rule;

outputting from the processor to the external network only those outgoing data messages the content of which has been qualified and validated; and

updating the system state information based on at least some of the qualified and validated content of one or more of the incoming or outgoing data messages.

45. The method of claim 44, wherein the rule-set includes a first rule-set configured to process the incoming data messages received from the external network, and a second rule-set configured to process the outgoing data messages received from the internal network.

46. The method of claim 44, wherein the processor has an operational mode and a programming mode, and wherein the rule-set is implemented only when the processor is in the operational mode.

47. The method of claim 44, wherein the qualification of the content of the incoming and outgoing data messages comprises:

determining the presence, in each data message, of unqualified content that is not in compliance with the qualification criteria defined by the rule-set; and

deleting any unqualified content determined to be present.

48. The method of claim 47, wherein the validation of the content of the incoming and outgoing data messages comprises:

examining, in accordance with the validation criteria defined by the rule-set, the qualified content of the data messages that has not been deleted, to determine compliance of the qualified content with the validation criteria defined by the rule-set; and

deleting any content determined to be non-compliant with the validation criteria defined by the rule-set.

49. The method of claim 48, wherein the validation further comprises, after examining the qualified content of the data messages, modifying at least some of the qualified content that has not been deleted so as to be compliant with further validation criteria defined by the rule-set.

50. The method of claim 44, further comprising creating a log entry in response to the deletion of unqualified content and/or non-compliant content.

51. An automation and control system that is operable in one or more system states and that includes a cyber-security functionality, the automation and control system comprising:

an external network including an external control terminal;

an internal network including at least one of a supervisor control and data acquisition (SCADA) device and an automation and control device;

a processor in data communication with the external network and the internal network;

a memory associated with the processor and configured to store an indication of the system state and a rule-set defining qualification and validation criteria for data contents of incoming data messages received from the external network and data contents of outgoing data messages received from the internal network, wherein the rule-set includes a system state-dependent rule;

wherein the processor is operable in an operational mode to process the data messages in accordance with the rule-set so as to (a) qualify, byte-by-byte, (i) the content of each of the data messages received from the external network via the external communication interface as conforming to qualification criteria defined by the rule-set for external-to-internal communications, and (ii) the content of each of the data messages received from the internal

network via the internal communication interface as conforming to qualification criteria defined by the rule-set for internal-to-external communications; (b) validate, byte-by-byte, the content of each qualified data message in accordance with validation criteria defined by the rule-set; and (c) pass between the internal network and the external network only data content that has been qualified and validated and that is deemed proper based on the indication of system state and compliance of system state information in the data message with the system state-dependent rule, and wherein the processor is re-programmable with a new rule-set only in a programming mode; and

a switch operable manually to switch the processor between the operational mode and the programming mode.

52. The system of claim 51, wherein the rule-set includes a first rule-set configured to process the incoming data messages received from the external network, and a second rule-set configured to process the outgoing data messages received from the internal network.

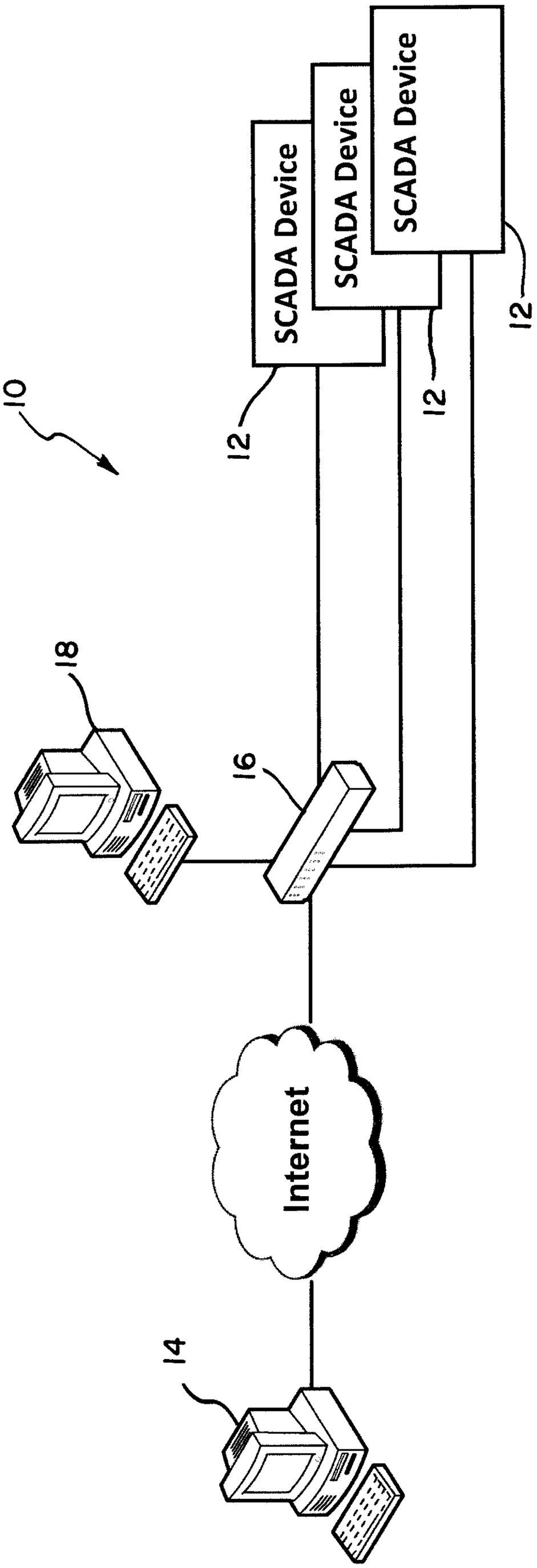

FIG. 1
(PRIOR ART)

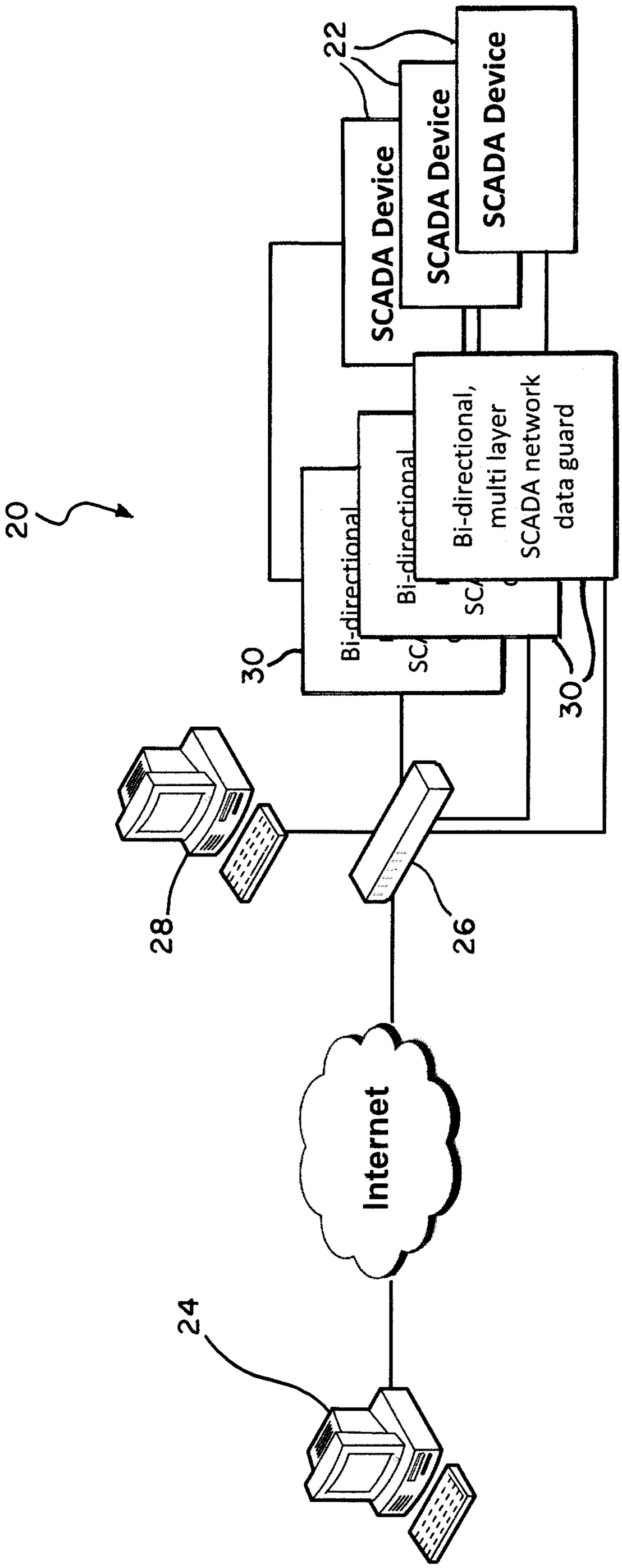
20
22
SCADA Device
SCADA Device
SCADA Device
Bi-directional
Bi-directional
Bi-directional,
multi layer
SCADA network
data guard
30
30
28
26
Internet
24

FIG. 2

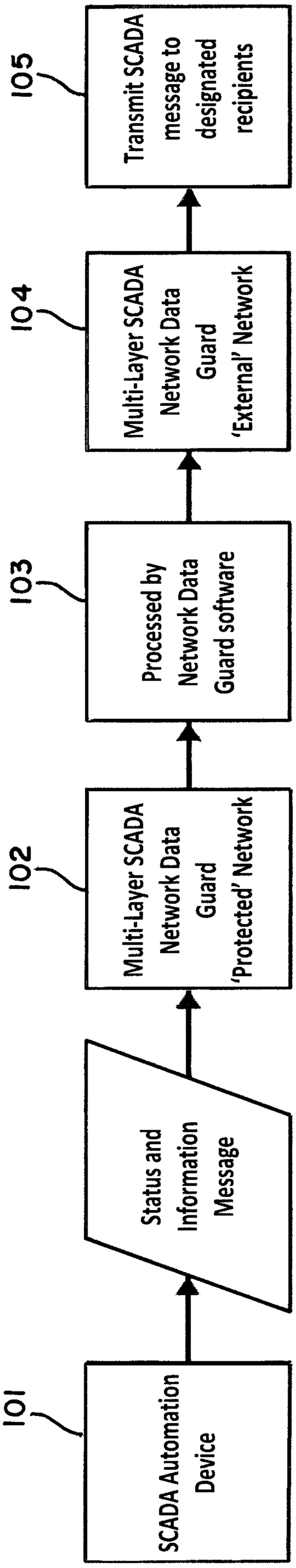
101
SCADA Automation Device
Status and Information Message
102
Multi-Layer SCADA Network Data Guard 'Protected' Network
103
Processed by Network Data Guard software
104
Multi-Layer SCADA Network Data Guard 'External' Network
105
Transmit SCADA message to designated recipients

FIG. 3

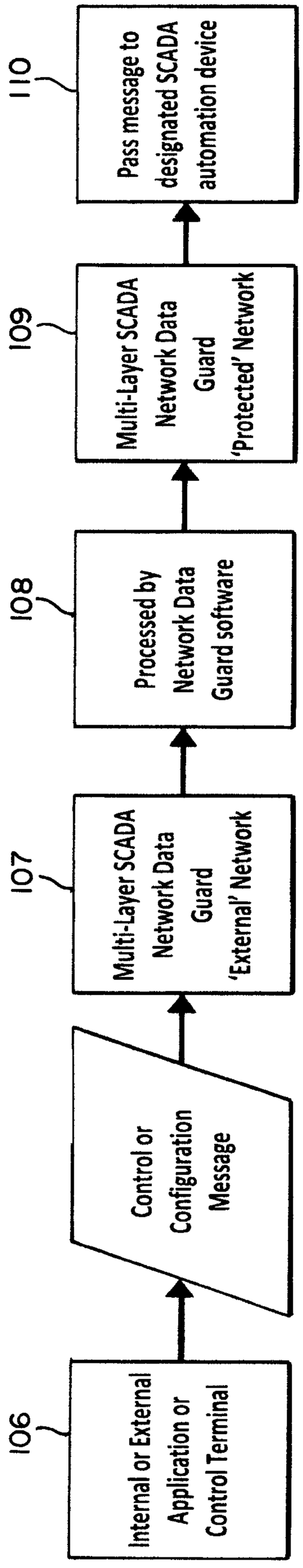
110
Pass message to designated SCADA automation device
109
Multi-Layer SCADA Network Data Guard 'Protected' Network
108
Processed by Network Data Guard software
107
Multi-Layer SCADA Network Data Guard 'External' Network
Control or Configuration Message
106
Internal or External Application or Control Terminal

FIG. 4

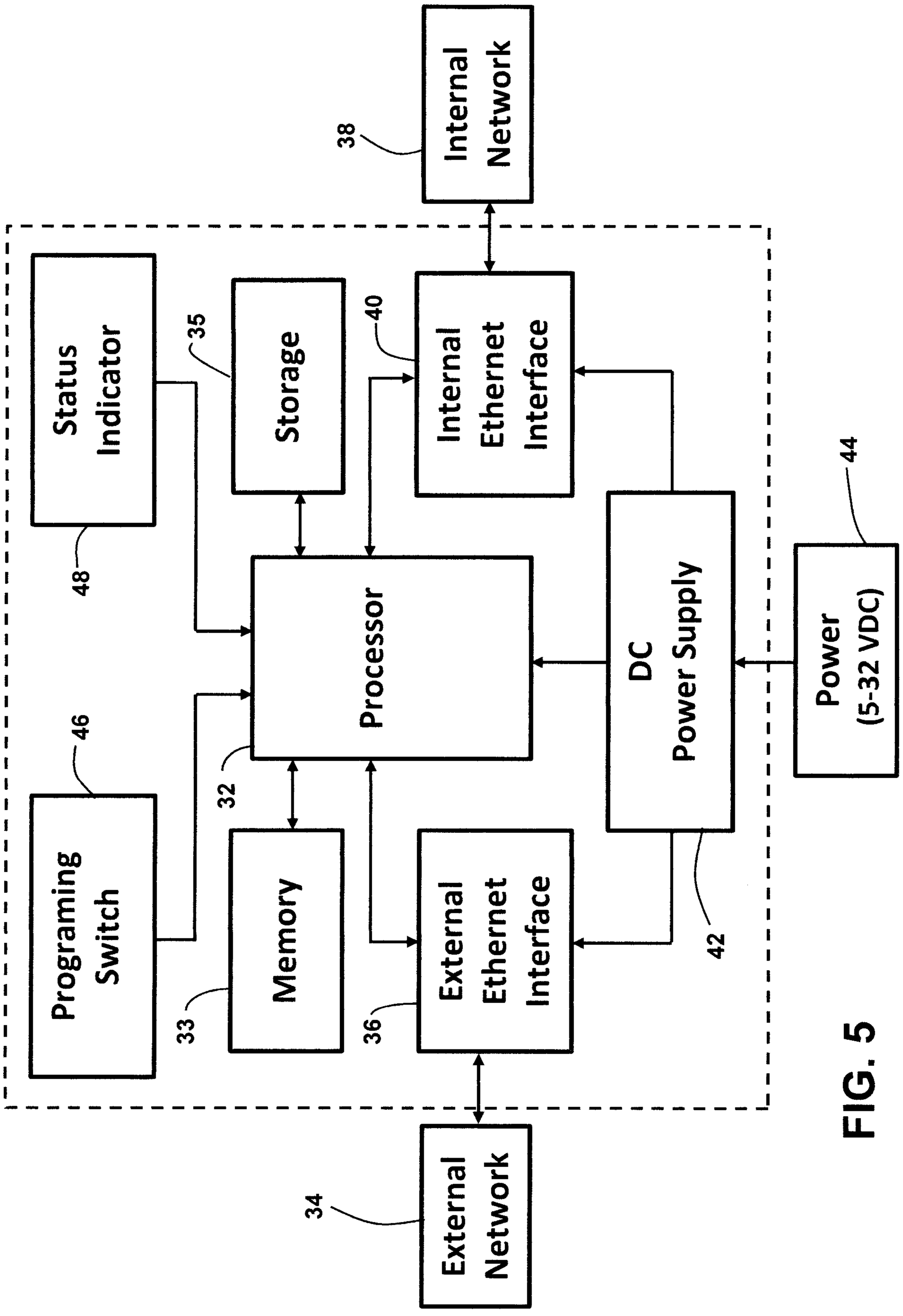
Internal Network
38
Status Indicator
35
Storage
40
Internal Ethernet Interface
44
48
Processor
DC Power Supply
Power (5-32 VDC)
46
32
Programing Switch
Memory
External Ethernet Interface
33
36
42
External Network
34

FIG. 5

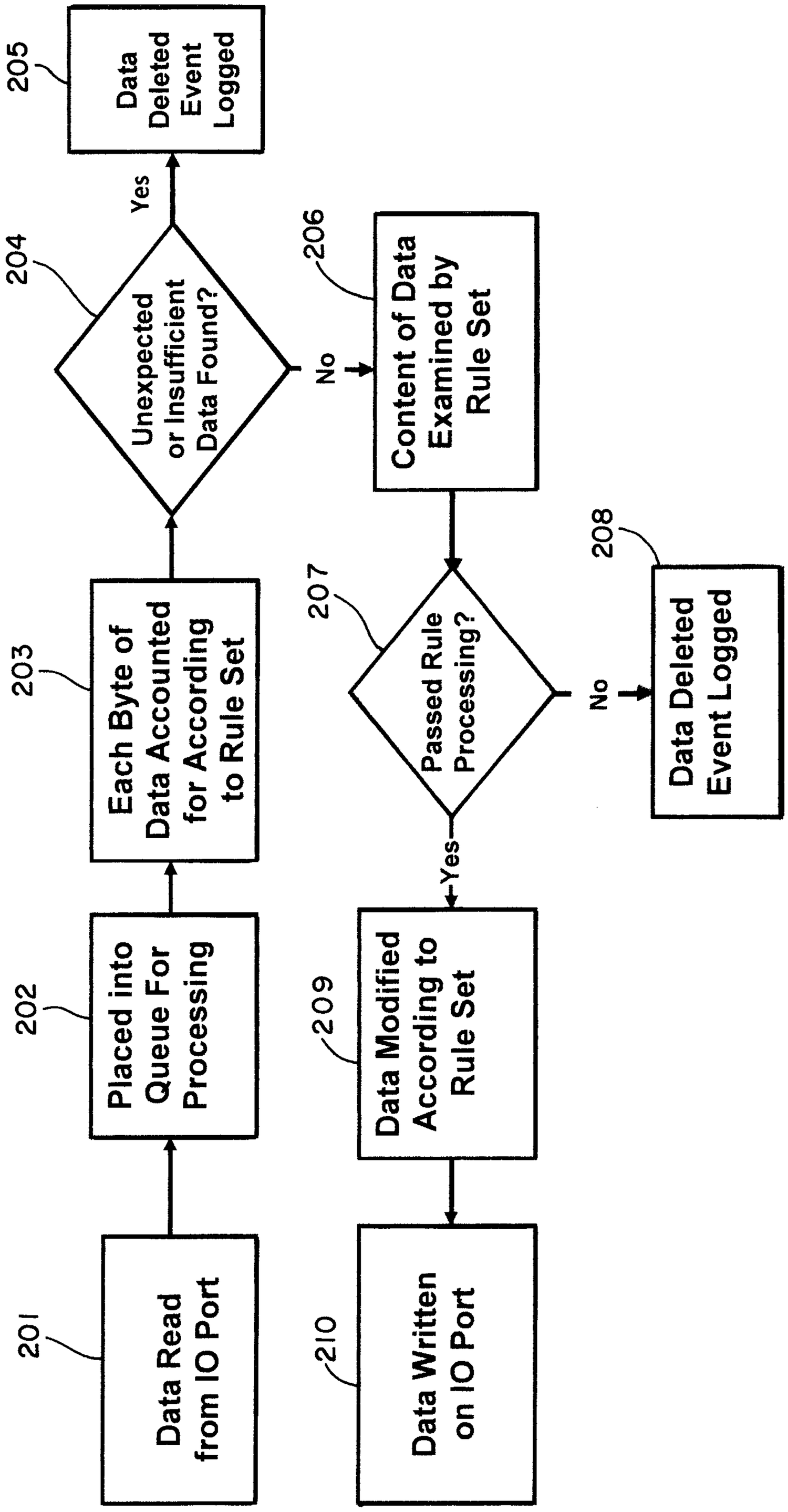
Data Read from IO Port
201
Placed into Queue For Processing
202
Each Byte of Data Accounted for According to Rule Set
203
Unexpected or Insufficient Data Found?
204
Yes
Data Deleted Event Logged
205
No
Content of Data Examined by Rule Set
206
Passed Rule Processing?
207
No
Data Deleted Event Logged
208
Yes
Data Modified According to Rule Set
209
Data Written on IO Port
210

FIG. 6

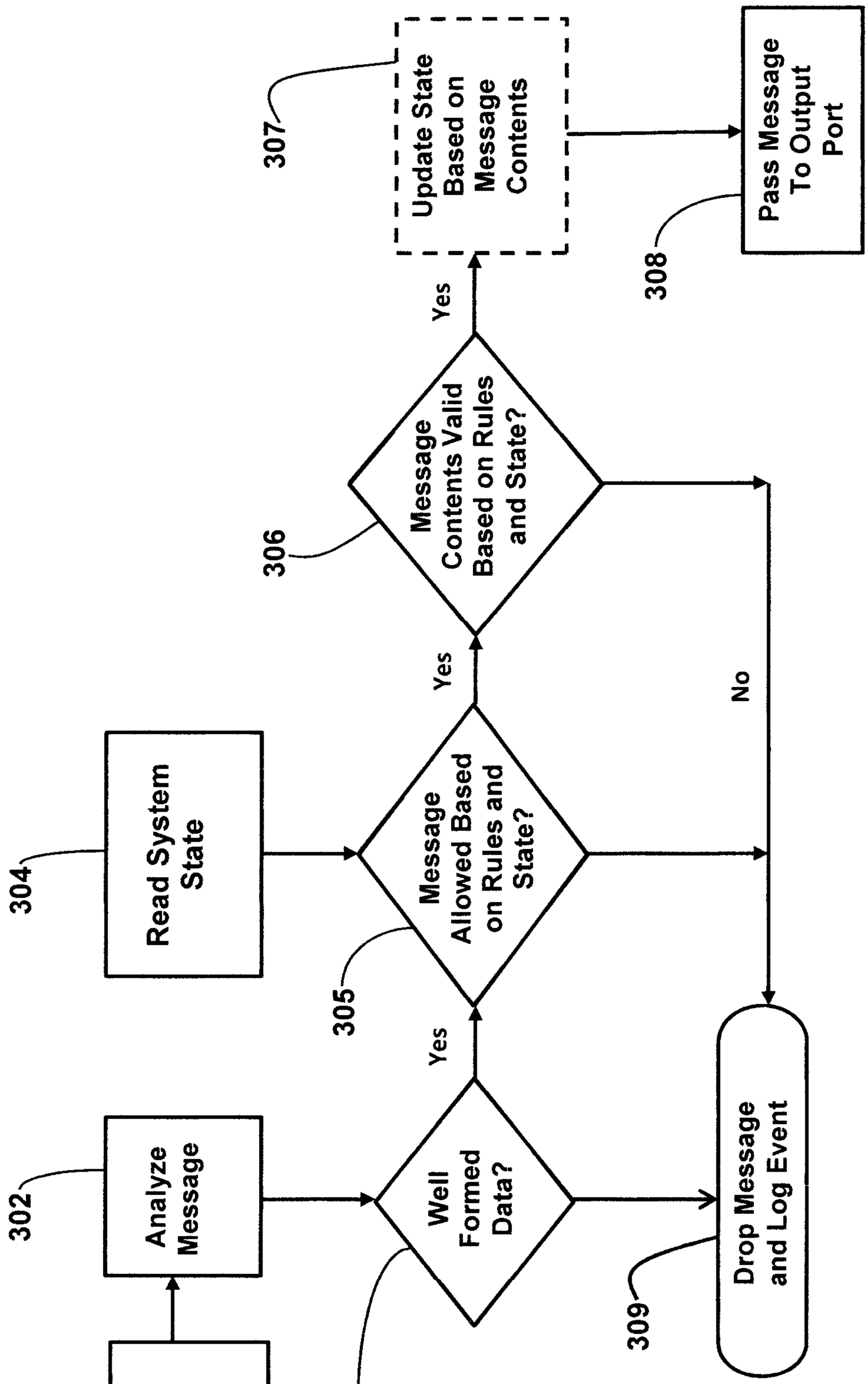
Read System State
304
Analyze Message
302
Message Data Read from IO Port
301
Well Formed Data?
303
Message Allowed Based on Rules and State?
305
Message Contents Valid Based on Rules and State?
306
Update State Based on Message Contents
307
Pass Message To Output Port
308
Drop Message and Log Event
309
Yes
No
Yes
Yes
Yes

FIG. 7

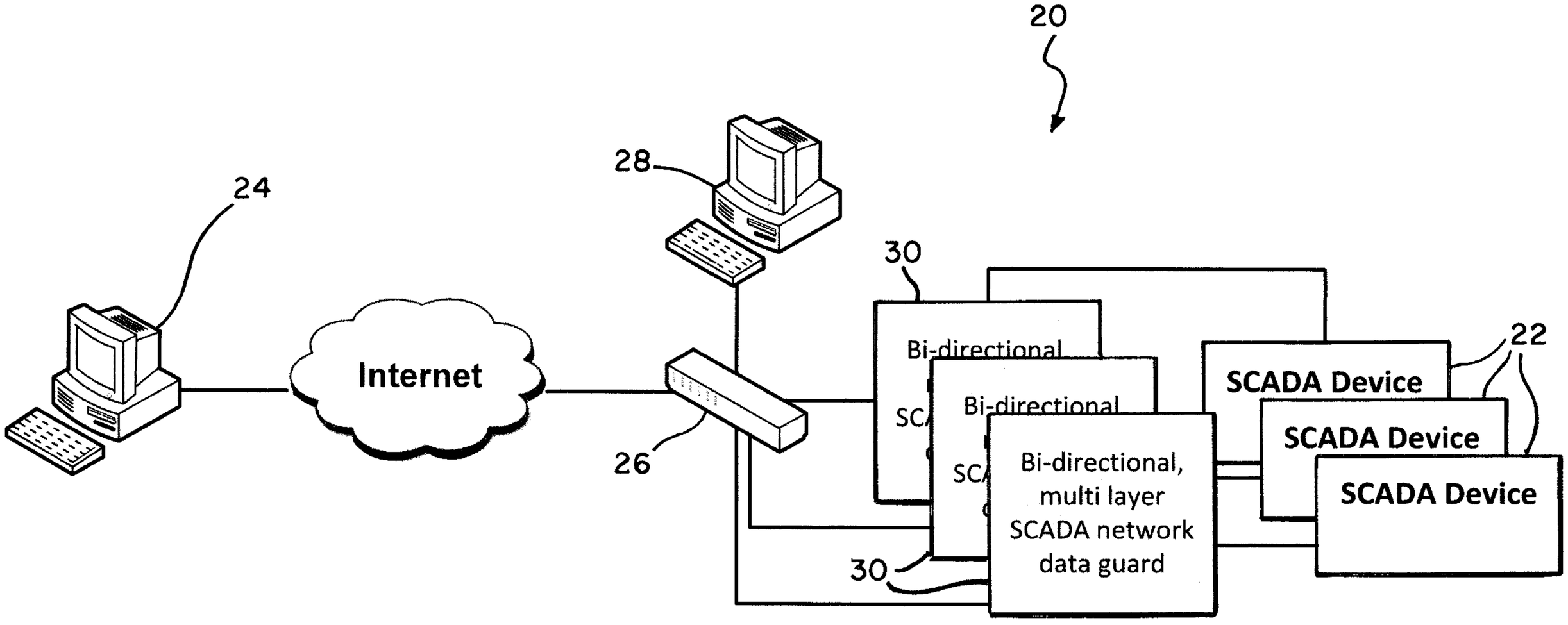
20
24
28
30
22
Internet
26
Bi-directional
Bi-directional
Bi-directional,
multi layer
SCADA network
data guard
SCADA Device
SCADA Device
SCADA Device
30

FIG. 2