



(51) International Patent Classification:
H04L 45/28 (2022.01) *H04L 69/40* (2022.01)
H04L 43/0811 (2022.01)

(21) International Application Number:
PCT/IN2024/050957

(22) International Filing Date:
27 June 2024 (27.06.2024)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
202321045206 05 July 2023 (05.07.2023) IN

(71) Applicant: **JIO PLATFORMS LIMITED** [IN/IN]; OF-
FICE-101, SAFFRON, NR. CENTRE POINT, PANCH-
WATI 5 RASTA, AMBAWADI, GUJARAT, AHMED-
ABAD 380006 (IN).

(72) Inventors: **BHATNAGAR, Aayush**; Office-101, Saffron,
Nr. Centre Point, Panchwati 5 Rasta, Ambawadi, Gujarat,
Ahmedabad 380006 (IN). **BISHT, Birendra**; Office-101,
Saffron, Nr. Centre Point, Panchwati 5 Rasta, Ambawa-
di, Gujarat, Ahmedabad 380006 (IN). **SINGH, Harbinder**;
Office-101, Saffron, Nr. Centre Point, Panchwati 5 Ras-
ta, Ambawadi, Gujarat, Ahmedabad 380006 (IN). **SOREN,**
Rohit; Office-101, Saffron, Nr. Centre Point, Panchwati
5 Rasta, Ambawadi, Gujarat, Ahmedabad 380006 (IN).
AGGARWAL, Pravesh; Office-101, Saffron, Nr. Centre
Point, Panchwati 5 Rasta, Ambawadi, Gujarat, Ahmedabad
380006 (IN). **SAHU, Bidhu**; Office-101, Saffron, Nr. Cen-
tre Point, Panchwati 5 Rasta, Ambawadi, Gujarat, Ahmed-
abad 380006 (IN). **SINGH, Priyanka**; Office-101, Saf-
fron, Nr. Centre Point, Panchwati 5 Rasta, Ambawadi, Gu-
jarat, India., Ahmedabad 380006 (IN). **JASWANI, Tikam**;
Office-101, Saffron, Nr. Centre Point, Panchwati 5 Ras-
ta, Ambawadi, Gujarat, Ahmedabad 380006 (IN). **SWAMI,**

(54) Title: METHOD AND SYSTEM FOR MANAGING NETWORK CONNECTIVITY

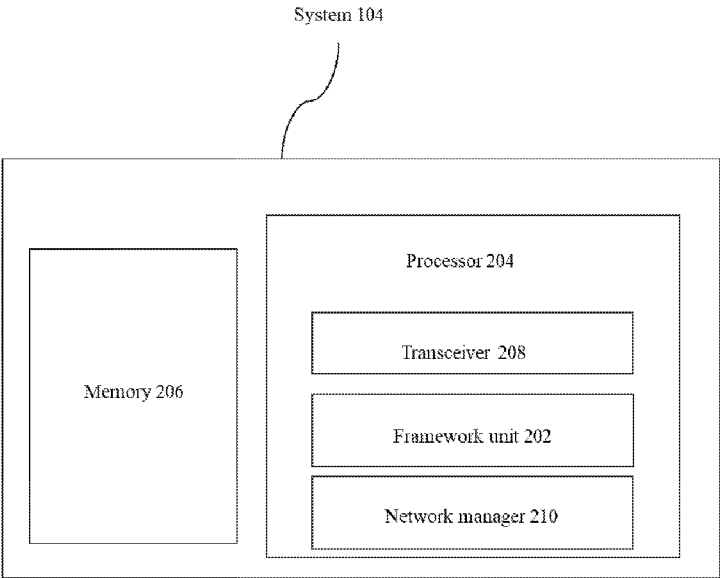


FIG. 2

(57) Abstract: The present disclosure relates to a system (104) and a method (400) for managing network connectivity during a process failure in a communication network (106). The system (104) includes a transceiver (208) configured to receive a failure trigger from a first node (102). The system (104) further includes a network manager (210) configured to disconnect a connection between the first node (102) and the system (104) on receiving the failure trigger in real time. The framework unit (202) is further configured to initiate one or more failover procedures, and transfer tasks to be performed by the first node (102) to a second node (108), and thereby manage network connectivity.

Mukul; Office-101, Saffron, Nr. Centre Point, Panchwati 5
Rasta, Ambawadi, Gujarat, Ahmedabad 380006 (IN).

(74) **Agent: JAPHET, Chinthan;** K Law (Krishnamurthy and Co.), 4th Floor, Prestige Takt, No 23, Kasturba Road Cross, Bangalore, Bangalore 560 001 (IN).

(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, CV, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IQ, IR, IS, IT, JM, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, MG, MK, MN, MU, 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, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW.

(84) **Designated States** (*unless otherwise indicated, for every kind of regional protection available*): ARIPO (BW, CV, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SC, 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, ME, 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:

— *of inventorship (Rule 4.17(iv))*

Published:

— *with international search report (Art. 21(3))*
— *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments (Rule 48.2(h))*

METHOD AND SYSTEM FOR MANAGING NETWORK CONNECTIVITY

FIELD OF THE INVENTION

[0001] The present invention relates to wireless communication networks, more particularly relates to managing network connectivity during a process failure in a communication network.

BACKGROUND OF THE INVENTION

[0002] In a network, large volume of requests, are received and processed within a minute time frame. Processing such large volume of requests in a satisfactory manner indicates a positive user experience. The network utilizes various network components such as servers and network functions in form of a cluster-based application, to process the received requests. However, sometimes the components of the network experience some downtime due to some errors or due to some hardware malfunctioning. Sorting of such errors and failures is essential to maintain the quality of service of the network.

[0003] In the cluster-based application of the network, a high availability framework (HA) is provided which detects process failures based on TCP connection failure or by heartbeat mechanism. The heartbeat mechanism includes sending a communication packet from one node to another node in the network in order to monitor the health of the nodes, networks and network interfaces, and to prevent cluster partitioning which occurs when communication is lost between one or more nodes in the cluster and a failure of the lost nodes cannot be confirmed.

[0004] It is well known in the art that the cluster is a group of inter-connected nodes or hosts that work together to support one or more applications and a middleware. In a cluster mode, the HA launches a driver inside the cluster. In a single cluster, there is indeed a single driver node responsible for managing the one or more applications. The cluster mode is a good choice for production of workloads pertaining to one or more applications that require high availability, scalability, and security.

[0005] Due to this, HA experiences delay in detecting process failures and initiating failover. These delays occur due to the time taken by the Linux kernel to clean up resources like TCP connections, RAM, open files etc. after a process exits due to crashes, resource exhausts, deadlock etc. Additionally, periodic heartbeat mechanisms introduce further delays. Therefore, the development of a rapid failure detection and recovery framework is crucial to improve the overall reliability and performance time sensitive systems.

SUMMARY OF THE INVENTION

[0006] One or more embodiments of the present disclosure provide a system and method for managing network connectivity during a process failure in a communication network.

[0007] In one aspect of the present invention, a method of managing network connectivity during the process failure in a communication network is provided. The method includes receiving a failure trigger from a first node. In an embodiment, the failure trigger is one of a crash signal, a resource exhaust signal, and a deadlock. The method further includes disconnecting, a connection between the first node and a one or more processors on receiving the failure trigger in real time. In an embodiment, upon disconnection the one or more processors is informed of the process failure in real time owing to the disconnection. The method further includes initiating one or more failover procedures. The method further includes invoking one or more Application Programming Interface (API) functions. Upon invoking one or more API functions, the method further initiates a kernel cleanup. The method further includes transferring tasks to be performed by the first node to a second node. In an embodiment, the first node is an active node, and the second node is a standby node. In an embodiment, the second node is selected simultaneously when the one or more API functions are invoked, thereby transitioning from the first node to the second node. In an embodiment, the method further transfers tasks to be performed by the first

node to the second node, enables takeover of restarting procedure and managing network connectivity during process failure.

[0008] In another aspect of the present invention a system for managing network connectivity during a process failure in a communication network is disclosed. The system includes a transceiver configured to receive, a failure trigger from a first node. In an embodiment, the failure trigger is one of a crash signal, a resource exhaust signal, and a deadlock. Further, the system includes a network manager configured to disconnect, a connection between the first node and the system on receiving the failure trigger in real time. In an embodiment, upon disconnection the framework unit is informed of the process failure in real time owing to the disconnection. The framework unit is further configured to initiate one or more failover procedures by invoking, one or more Application Programming Interface (API) functions. Upon invoking one or more API functions, the framework unit initiates a kernel cleanup. The framework unit is further configured to transfer tasks to be performed by the first node to a second node. In an embodiment, the first node is an active node, and the second node is a standby node. In an embodiment, the second node is selected simultaneously when the one or more API functions are invoked, thereby transitioning from the first node to the second node. In an embodiment, the framework unit transfers tasks to be performed by the first node to the second node, enables takeover of restarting procedure and managing network connectivity during process failure.

[0009] Other features and aspects of this invention will be apparent from the following description and the accompanying drawings. The features and advantages described in this summary and in the following detailed description are not all-inclusive, and particularly, many additional features and advantages will be apparent to one of ordinary skill in the relevant art, in view of the drawings, specification, and claims hereof. Moreover, it should be noted that the language used in the specification has been principally selected for readability and instructional purposes and not have been selected to delineate or circumscribe the inventive subject matter, resort to the claims being necessary to determine such inventive subject matter.

BRIEF DESCRIPTION OF THE DRAWINGS

[0010] The accompanying drawings, which are incorporated herein, and constitute a part of this disclosure, illustrate exemplary embodiments of the disclosed methods and systems in which like reference numerals refer to the same parts throughout the different drawings. Components in the drawings are not necessarily to scale, emphasis instead being placed upon clearly illustrating the principles of the present disclosure. Some drawings indicates the components using block diagrams and possibly not represent the internal circuitry of each component. It will be appreciated by those skilled in the art that disclosure of such drawings includes disclosure of electrical components, electronic components or circuitry commonly used to implement such components.

[0011] **FIG. 1** is an exemplary block diagram of an environment for managing network connectivity during a process failure in a communication network, according to various embodiments of the present disclosure;

[0012] **FIG. 2** is a block diagram of a system for managing network connectivity during the process failure in the communication network, according to various embodiments of the present disclosure;

[0013] **FIG. 3** is an exemplary schematic representation of the system of **FIG. 1** in which various entities operations are explained, according to various embodiments of the present system;

[0014] **FIG. 4** shows a flow diagram of a method for managing network connectivity during the process failure in the communication network, according to various embodiments of the present disclosure.

[0015] The foregoing shall be more apparent from the following detailed description of the invention.

DETAILED DESCRIPTION OF THE INVENTION

[0016] Some embodiments of the present disclosure, illustrating all its features, will now be discussed in detail. It must also be noted that as used herein and in the appended claims, the singular forms "a", "an" and "the" include plural references unless the context clearly dictates otherwise.

[0017] Various modifications to the embodiment will be readily apparent to those skilled in the art and the generic principles herein are applied to other embodiments. However, one of ordinary skill in the art will readily recognize that the present disclosure including the definitions listed here below are not intended to be limited to the embodiments illustrated but is to be accorded the widest scope consistent with the principles and features described herein.

[0018] A person of ordinary skill in the art will readily ascertain that the illustrated steps detailed in the figures and here below are set out to explain the exemplary embodiments shown, and it should be anticipated that ongoing technological development will change the manner in which particular functions are performed. These examples are presented herein for purposes of illustration, and not limitation. Further, the boundaries of the functional building blocks have been arbitrarily defined herein for the convenience of the description. Alternative boundaries can be defined so long as the specified functions and relationships thereof are appropriately performed. Alternatives (including equivalents, extensions, variations, deviations, etc., of those described herein) will be apparent to persons skilled in the relevant art(s) based on the teachings contained herein. Such alternatives fall within the scope and spirit of the disclosed embodiments.

[0019] As per various embodiments depicted, the present invention discloses the system and method for managing network connectivity during a process failure in a communication network. Further the invention provides a framework unit, configured to detect the process failure in a cluster, and initiate a recovery process. In another aspect of the present invention, an immediate notification of the process failure is

provided to the framework unit based on a failure trigger. Further the immediate termination ensures that the framework unit is promptly notified about the active process going down, allowing the framework unit to initiate failover procedure without any delay.

[0020] In an embodiment, by terminating the connection between the first node and the system immediately upon process failure, the mechanism eliminates the need to rely solely on the operating system (such as, for example, but are not limited to Linux) kernel's cleanup process. This approach bypasses any potential delays caused by the kernel cleanup, ensuring faster and more efficient detection of the process failures. Further in accordance with an example an `exit()` or an `_exit()` Application Programming Interface (API) function is initiated to sever the TCP connection upon detecting process failure to perform the kernel cleanup.

[0021] Referring to **FIG. 1**, **FIG. 1** illustrates an exemplary block diagram of an environment 100 for managing network connectivity during a process failure in a communication network 106, according to various embodiments of the present disclosure. The environment 100 comprises a first node 102 configured to connect to a system 104 via a communication network 106. The first node 102 is an active node configured to actively perform process in real-time. Further the first node 102 communicably connects to the system 104.

[0022] The first node 102 communicates with the system 104 via the communication network 106 over a protocol for example a Transmission Control Protocol (TCP).

[0023] The environment 100 further comprises a second node 108 configured to connect with the system 104 via the communication network 106. In an embodiment, the second node 108 is a standby node configured to recover the process in real-time. Further the second node 108 is communicably connected to the system 104. The second node 108 communicates with the system 104 via the communication network 106 over the TCP protocol.

[0024] In an embodiment, the first node 102 and the second node 108 include at least one of, but not limited to, a computer system with one or more communication ports coupled to one or more communication bus, an user equipment with an interface, a wireless device including, by the way of example not limitation, a handheld wireless communication device (such as a mobile phone, a smart phone, and a phablet device), a wearable computer device (such as a head-mounted display computer device, a head-mounted camera device, and a wristwatch computer device), a Global Positioning System (GPS) device, a laptop computer, a tablet computer, or another type of portable computer, a media playing device, a portable gaming system, and/or any other type of computer device with wireless communication capabilities.

[0025] In various embodiments, the system 104 is integrated with any application including a System Management Facility (SMF), an Access and Mobility Management Function (AMF), a Business Telephony Application Server (BTAS), a Converged Telephony Application Server (CTAS), any SIP (Session Initiation Protocol) Application Server which interacts with core Internet Protocol Multimedia Subsystem(IMS) on Industrial Control System (ISC) interface as defined by 3GPP to host a wide array of cloud telephony enterprise services, a System Information Blocks (SIB)/ and a Mobility Management Entity (MME).

[0026] The communication network 106 includes, by way of example but not limitation, one or more of a wireless network, a wired network, an internet, an intranet, a public network, a private network, a packet-switched network, a circuit-switched network, an ad hoc network, an infrastructure network, a Public-Switched Telephone Network (PSTN), a cable network, a cellular network, a satellite network, a fiber optic network, or some combination thereof. The communication network 106 includes, but is not limited to, a Third Generation (3G), a Fourth Generation (4G), a Fifth Generation (5G), a Sixth Generation (6G), a New Radio (NR), a Narrow Band Internet of Things (NB-IoT), an Open Radio Access Network (O-RAN), and the like.

[0027] The communication network 106 uses one or more communication interfaces/protocols such as, for example, Transmission Control Protocol (TCP),

Voice Over Internet Protocol (VoIP), 802.11 (Wi-Fi), 802.15 (including Bluetooth™), 802.16 (Wi-Max), 802.22, Cellular standards such as Code Division Multiple Access (CDMA), CDMA2000, Wideband CDMA (WCDMA), Radio Frequency Identification (e.g., RFID), Infrared, laser, Near Field Magnetics, etc.

[0028] In an embodiment, the system 104 is described as an integral part of the server 110, without deviating from the scope of the present disclosure. In an alternate embodiment, the system 104 is a standalone device.

[0029] The server 110 can be, for example, but not limited to a standalone server, a server blade, a server rack, a bank of servers, a business telephony application server (BTAS), a server farm, a cloud server, an edge server, home server, a virtualized server, one or more processors executing code to function as a server, or the like. In an implementation, the server 110 is operated at various entities or a single entity (include, but is not limited to, a vendor side, service provider side, a network operator side, a company side, an organization side, a university side, a lab facility side, a business enterprise side, a defense facility side, or any other facility) that provides service.

[0030] With reference to **FIG. 2**, the system 104 includes one or more processors 204 coupled with a memory 206, wherein the memory 206 stores instructions which when executed by the one or more processors 204 causes the system 104 to detect the process failure in the communication network 106. The one or more processor(s) 204 is implemented as one or more microprocessors, microcomputers, microcontrollers, edge or fog microcontrollers, digital signal processors, central processing units, logic circuitries, and/or any devices that process data based on operational instructions. Among other capabilities, the one or more processor(s) 204 is configured to fetch and execute computer-readable instructions stored in a memory 206 of the system 104. The memory 206 is configured to store one or more computer-readable instructions or routines in a non-transitory computer readable storage medium, which is fetched and executed to manage network connectivity during the process failure. The memory 206 includes for example, Random Access Memory (RAM), a non-volatile memory (e.g.,

disk memory, FLASH memory, EPROMs, etc.), an unalterable memory, and/or other types of memory. In one implementation, the memory 206 might be configured or designed to store data pertaining to the process failure.

[0031] In an embodiment, the system further includes an interface(s). The interface(s) comprises a variety of interfaces, for example, interfaces for data input and output devices, referred to as input/output (I/O) devices, storage devices, and the like. The interface(s) facilitates communication for the system. The interface(s) also provides a communication pathway for one or more components of the system. Examples of such components include, but are not limited to, processing unit/engine(s) and a database. The processing unit/engine(s) is implemented as a combination of hardware and programming (for example, programmable instructions) to implement one or more functionalities of the processing engine(s).

[0032] Operational and construction features of the system 104 will be explained in detail with respect to the following figures.

[0033] Referring to **FIG. 2**, **FIG. 2** illustrates a block diagram of the system 104 provided for managing network connectivity during the process failure in the communication network 106, according to one or more embodiments of the present invention.

[0034] As per the illustrated embodiment, the system 104 includes one or more processors 204, and a memory 206. The one or more processors 204, is implemented as one or more microprocessors, microcomputers, microcontrollers, digital signal processors, central processing units, state machines, logic circuitries, single board computers, and/or any devices that manipulate signals based on operational instructions. As per the illustrated embodiment, the system 104 includes one or more processor 204. However, it is to be noted that the system 104 includes multiple processors as per the requirement and without deviating from the scope of the present disclosure.

[0035] The information related to the process failure is provided or stored in the memory 206 of the system 104. Among other capabilities, the one or more processors 204 is configured to fetch and execute computer-readable instructions stored in the memory 206. The memory 206 is configured to store one or more computer-readable instructions or routines in a non-transitory computer-readable storage medium, which is fetched and executed to manage network connectivity during the process failure.

[0036] Further, the one or more processors 204, in an embodiment, is implemented as a combination of hardware and programming (for example, programmable instructions) to implement one or more functionalities of the one or more processors 204. In the examples described herein, such combinations of hardware and programming is implemented in several different ways. For example, the programming for the one or more processors 204 is processor-executable instructions stored on a non-transitory machine-readable storage medium and the hardware for the one or more processors 204 comprises a processing resource (for example, one or more processors), to execute such instructions. In the present examples, the memory 206 stores instructions that, when executed by the processing resource, implement the one or more processors 204. In such examples, the system 104 comprises the memory 206 storing the instructions and the processing resource to execute the instructions, or the memory 206 is separate but accessible to the system 104 and the processing resource. In other examples, the one or more processors 204 is implemented by electronic circuitry.

[0037] In order for the system 104 to manage the network connectivity during the process failure, the processor 204 includes a transceiver 208, a network manager 210 and the framework unit 202. The framework unit 202 is at least one of, but not limited to, a layered structure, a set of instructions or functions embedded within the processor 204 of the system 104.

[0038] The transceiver 208, the framework unit 202 and the network manager 210 in an embodiment, are implemented as a combination of hardware and programming

(for example, programmable instructions) to implement one or more functionalities of the one or more processors 204. The transceiver 208 of the processor 204 is communicably connected to the first node 102 and the second node 108 via the communication network 106. In an embodiment, the first node 102 is the active node, and the second node is the standby node. Accordingly, the transceiver 208 is configured to receive a failure trigger from the first node 102. In an embodiment the failure trigger is one of a crash signal, a resource exhaust signal, a deadlock, and a closure notification pertaining to closing of a task at the first node 102.

[0039] The network manager 210 of the processor 204 is configured to disconnect, a connection between the first node 102 and the framework unit 202 upon receiving the failure trigger. The connection between the first node 102 and the framework unit 202 includes at least one of, the Transmission Control Protocol (TCP) connection. In an embodiment, the TCP is a connection-oriented protocol for communications that facilitates the exchange of data or messages between the first node 102 and the system 104 in the communication network 106.

[0040] In an embodiment, the framework unit 202 is configured to initiate one or more failover procedures and transfer one or more tasks to be performed by the first node 102 to the second node 108. In an embodiment, the one or more failover procedures includes at least one of, kernel cleanup process, restarting the process, and transferring the process to standby second node 108 from the first node 102. Subsequent to disconnecting the connection between the first node 102 and the framework unit 202 by the network manager 210 based on the failure trigger, the framework unit 202 initiates the one or more failover procedures by initiating the kernel cleanup process of the first node 102 or restarting the process performed by the first node. By performing the kernel cleanup process, the framework unit 202 deallocates the various types of resources utilized by the first node 102 while performing the one or more tasks.

[0041] In an embodiment, the one or more tasks performed by at least one of, the first node 102 and the second node 108 include, but is not limited to, data transmission, data processing, data testing, and data analyzing.

[0042] In an alternate embodiment, the network manager 210 is configured to inform the framework unit 202 of the process failure in real time owing to the disconnection. Subsequent to invoking one or more Application Programming Interface (API) functions at the first node 102 Upon invoking the one or more API functions, the framework unit 202 is configured to initiate a kernel cleanup process. The one or more API functions include at least one of an exit function such as `_exit()` and `exit()`.

[0043] In an embodiment, the kernel 308 is a core component serving as the primary interface between hardware components of the first node 102 and the plurality of applications hosted thereon. The plurality of applications includes at least one of, streaming application, and messaging application. The kernel 308 is configured to provide the plurality of applications hosted on the first node 102 and access to resources available in the communication network 106. The resources include one of a Central Processing Unit (CPU), memory components such as Random Access Memory (RAM) and Read Only Memory (ROM).

[0044] In an embodiment, the API is defined as medium of communication in between the system 104 and at least one of, the first node 102 and the second node 108. The framework unit 202 initiates one or more failover procedures based on the invoking the API functions at the first node 102 . The API functions include at least one of an exit function such as `_exit()` and `exit()`. Utilizing the API functions, the framework unit 202 initiates the kernel cleanup process. The API is operable by providing a set of instructions in suitable formats like JSON (JavaScript Object Notation), Python or any other such compatible formats.

[0045] Referring to **FIG. 3**, **FIG. 3** describes the system 104 for managing network connectivity during the process failure in the communication network 106 along with reference to the first node and the second node. It is to be noted that the embodiment with respect to **FIG. 3** will be explained with respect to the first node 102, the system 104 and the second node 108 for the purpose of description and illustration and should nowhere be construed as limiting the scope of the present disclosure.

[0046] The first node 102 includes one or more primary processors 304 communicably coupled to the framework unit 202 of the one or more processors 204. The one or more primary processors 304 are coupled with a memory unit 306 storing instructions which are executed by the one or more primary processors 304. Execution of the stored instructions by the one or more primary processors 304 enables the first node 102 to execute one or more tasks utilizing the system 104. The execution of the stored instructions by the one or more primary processors 304 further enables the first node 102 to transmit the failure trigger pertaining to the process failure to the one or more processors 204 subsequent to invoking the one or more API functions including at least one of an exit function such as `_exit()` and `exit()`.

[0047] In one embodiment, the exit function terminates the currently running one or more tasks/process on the first node 102. In particular, the `exit()` function flushes a plurality of buffers used by the one or more tasks/process, closes each program associated with the one or more tasks/process, and deletes temporary files associated with the one or more tasks/process.

[0048] In one embodiment, one or more API functions are invoked at the first node 102 when the one or more tasks/process fails due to at least one of, but not limited to, a crash signal, a resource exhaust signal, and a deadlock between the system 104 and the first node 102. The invoking of the one or more API functions indicates that the one or more tasks/process are going down and requires one or more failover procedures to be initiated promptly.

[0049] In an embodiment, to detect the process failure, the one or more processors 204 continuously monitors the one or more tasks performed by the first node 102. The one or more processors 204 detects the process failure when there is at least one of, but not limited to, a crash signal, a resource exhaust signal, and a deadlock between the system 104 and the first node 102. Subsequent to the detection of the process failure, the one or more primary processors 304 of the first node 102 generates failure trigger. The generated failure trigger is transmitted to the one or more processors 204 included in the system 104.

[0050] In an embodiment, the crash signal is a scenario when several nodes in the communication network 106 transmits data at the same time due to which a collision is detected, at that time a crash signal is detected at the system 104 and the several nodes stop transmitting data for a random time period before attempting again.

[0051] In an embodiment, a deadlock is a situation where a set of task/process are blocked because each process is holding a resource and waiting for another resource acquired by some other task/process. For example, Process 1 is holding resource 1 and waiting for resource 2 which is acquired by process 2, and process 2 is waiting for resource 1.

[0052] In an embodiment, the resource exhaustion happens when the system 104 uses each of the available resource such that the system 104 the resources are completely drained. Resource exhaustion occurs when the system's 104 resources like memory, processing power, or disk space are fully used and can't cater to additional demands due to which the resource exhaust signal is provided to the system 104 by the server 110.

[0053] In the preferred embodiment, the transceiver 208 of the processor 204 is configured to receive the failure trigger from the one or more primary processors 304 of the first node 102. In an embodiment the failure trigger is triggered due to the

process failure which includes at least of, a crash signal, a resource exhaust signal, and a deadlock.

[0054] In an embodiment, the framework unit 202 initiates one or more failover procedures without any delay based on the failure trigger. The one or more failover procedures includes a recovery process. The framework unit 202 during the recovery process, activates the second node 108 which is the standby node. Further, during the recovery process, the framework unit 202 transfers the one or more tasks being executed by the one or more primary processors 304 of the first node 102 to one or more primary processors 314 of the second node 108.

[0055] In an alternate embodiment, the second node 108 includes one or more primary processors 314 communicably coupled to the framework unit 202 of the one or more processors 204. The one or more primary processors 314 are coupled with a memory unit 316 storing instructions which are executed by the one or more primary processors 314. Execution of the stored instructions by the one or more primary processors 314 enables the second node 108 to execute one or more tasks transferred by the one or more processors 204, utilizing the system 104, from the first node 102 to the second node 108 subsequent to the process failure between the first node 102 and the system 104.

[0056] **FIG. 4** shows a flow diagram of a method 400 for managing network connectivity during the process failure in the communication network 106, according to various embodiments of the present disclosure. More specifically, the method recovers the process in the communication network 106. For the purpose of description, the method 400 is described with the embodiments as illustrated in **FIG. 2** and should nowhere be construed as limiting the scope of the present disclosure.

[0057] At step 402, the method 400 includes the step of receiving a failure trigger from a first node 102. In an embodiment the transceiver 208 receives the failure trigger. In an embodiment the first node 102 is an active node. The failure trigger is

due to the process failure which includes at least of, a crash signal, a resource exhaust signal, and a deadlock.

[0058] At step 404, the method 400 includes the step of disconnecting, a connection between the first node 102 and a processor 204 on receiving the failure trigger in real time by the network manager 210. The connection between the first node 102 and the processors (204) includes at least one of, a TCP connection. In an embodiment, upon disconnection, the framework unit 202 is informed of the process failure in real time owing to the disconnection and the framework unit 202 receives specific details about the failure trigger.

[0059] At step 406, the method 400 includes the step of initiating, one or more failover procedures by the framework unit 202. In an embodiment, upon specific details about the failure trigger, the framework unit 202 immediately recognizes that an active process is going down. Thereby initiating one or more failover procedures without any delay. The one or more failover procedures are initiated within milliseconds of the process failure. The one or more failover procedures includes a recovery process by activating the second node 108 which is the standby node subsequent to invoking one or more Application Programming Interface (API) functions at the first node 102. In an embodiment, upon invoking the one or more API functions, the framework unit 202 initiates a kernel cleanup. In an embodiment, in order to clean up the kernel 308 the one or more API functions such as are `_exit()` or `exit()` are invoked at the first node 102.

[0060] At step 408, the method 400 includes the step of transferring, tasks to be performed by the first node 102 to the selected second node 108. The second node 108 is selected by the framework unit 202 simultaneously when one or more API functions are invoked, thereby transitioning from the first node 102 to the second node 108. In an embodiment, by transferring, tasks to be performed by the first node 102 to the second node 108, the framework unit 202 enables takeover of recovery procedure and managing network connectivity during process failure. Advantageously, due to which

the downtime is reduced and the one or more tasks are performed without any interruption.

[0061] The present invention further discloses a non-transitory computer-readable medium having stored thereon computer-readable instructions. The computer-readable instructions are executed by a processor 204. The processors 204 is configured to receive a failure trigger from the first node 102. The processors 204 is further configured to disconnect the connection between the first node 202 and the processor 204 on receiving the failure trigger in real time. In an embodiment, upon disconnection the processor 204 is informed of the failure in real time owing to the disconnection. The processors 204 is further configured to initiate one or more failover procedures by invoking one or more Application Programming Interface (API) functions. Upon invoking one or more API functions, the processors 204 initiates a kernel cleanup. The processors 204 is further configured to transfer one or more tasks to be performed by the first node 102 to a second node 108 thereby restarting the one or more tasks and managing network connectivity during process failure.

[0062] A person of ordinary skill in the art will readily ascertain that the illustrated embodiments and steps in description and drawings (FIG.1-4) are set out to explain the exemplary embodiments shown, and it should be anticipated that ongoing technological development will change the manner in which particular functions are performed. These examples are presented herein for purposes of illustration, and not limitation. Further, the boundaries of the functional building blocks have been arbitrarily defined herein for the convenience of the description. Alternative boundaries can be defined so long as the specified functions and relationships thereof are appropriately performed. Alternatives (including equivalents, extensions, variations, deviations, etc., of those described herein) will be apparent to persons skilled in the relevant art(s) based on the teachings contained herein. Such alternatives fall within the scope and spirit of the disclosed embodiments.

[0063] The present disclosure incorporates technical advancement of promptly notifying the high availability framework about the process failure. The optimized mechanism enables immediate failover procedures which minimizes the downtime and service disruptions experienced by the telecom product, ensuring uninterrupted operation. The optimization reduces the dependency on the Linux kernel's cleanup process for TCP connections after process failure. This results in significantly shorter detection delays, allowing the high availability framework to swiftly respond to process failures. The enhanced process monitoring capabilities enable real time detection of failures. By optimizing the process failure detection and recovery framework, failover procedures can be initiated within milliseconds of a process failure. This level of responsiveness aligns with the stringent requirements of telecom products, minimizing the impact on critical operations and providing seamless user experience. The optimized mechanism enhances the overall resilience and responsiveness of the high availability framework. It reduces the risk of extended service disruptions, improves fault tolerance, and ultimately contributes to a more reliable and robust telecom product. Further, the proposed method can proactively optimise process failure detection and initiate the recovery process immediately after the process failure detection, so as to reduce delays in detecting process failure, enhance process monitoring, enhance service reliability and also saves a lot of time.

[0064] The present invention offers multiple advantages over the prior art and the above listed are a few examples to emphasize on some of the advantageous features. The listed advantages are to be read in a non-limiting manner.

REFERENCE NUMERALS

- [0065]** Environment - 100;
- [0066]** First Node - 102;
- [0067]** System – 104;
- [0068]** Communication Network - 106;

- [0069] Second Node - 108;
- [0070] Server - 110;
- [0071] Framework unit – 202;
- [0072] One or more processors – 204;
- [0073] Memory – 206;
- [0074] Transceiver – 208;
- [0075] Network manager – 210;
- [0076] Primary processors of first node - 304;
- [0077] Memory unit of first node - 306;
- [0078] Kernel – 308;
- [0079] Primary processors of second node - 314;
- [0080] Memory unit of second node -316; and
- [0081] Processor of a non-transitory computer-readable medium – 204.

We Claim:

1. A method (400) of managing network connectivity during failure, the method comprising the steps of:
 - receiving, (402) by one or more processors (204), a failure trigger from a first node (102);
 - disconnecting, (404) by the one or more processors (204), a connection between the first node (102) and the one or more processors (204) on receiving the failure trigger in real time;
 - initiating, (406) by the one or more processors (204), one or more failover procedures;
 - transferring, (408) by the one or more processors (204), tasks to be performed by the first node (102) to a second node (108).
2. The method (400) as claimed in claim 1, wherein the failure trigger is one of a crash signal, a resource exhaust signal, a deadlock, and a closure notification pertaining to closing of the tasks at the first node (102).
3. The method (400) as claimed in claim 1, wherein upon disconnection, the one or more processors (204) is informed of the failure in real time owing to the disconnection.
4. The method (400) as claimed in claim 1, wherein the first node (102) is an active node, and the second node (108) is a standby node.
5. The method (400) as claimed in claim 1, wherein the step of, initiating one or more failover procedures, includes the step of:
 - invoking, by the one or more processors (204), one or more Application Programming Interface (API) functions.

6. The method (400) as claimed in claim 5, wherein upon invoking the one or more API functions, the one or more processors (204) is configured to initiate a kernel cleanup.
7. The method (400) as claimed in claim 5, wherein the second node (108) is selected by the one or more processors (204) simultaneously when the one or more API functions are invoked, thereby transitioning from the first node (102) to the second node (108).
8. The method (400) as claimed in claim 1, wherein the one or more processors (204), by transferring, tasks to be performed by the first node (102) to the second node (108), enables takeover of restarting procedure and managing network connectivity during failure.
9. A system (104) for managing network connectivity during failure, the system comprising:
 - a transceiver (208) configured to, receive, a failure trigger from a first node;
 - a network manager(210) configured to:
 - disconnect, a connection between the first node and the system (104) on receiving the failure trigger;
 - a framework unit (202) configured to:
 - initiate, one or more failover procedures; and
 - transfer, tasks to be performed by the first node to a second node.
10. The system (104) as claimed in claim 9, wherein the failure trigger is one of a crash signal, a resource exhaust signal, a deadlock and a closure notification pertaining to closing of the tasks at the first node (102).

11. The system (104) as claimed in claim 9, wherein upon disconnection, the framework unit (202) is informed regarding the failure in real time owing to the disconnection.
12. The system (104) as claimed in claim 9, wherein the first node (102) is an active node and the second node (108) is a standby node.
13. The system (104) as claimed in claim 9, wherein the framework unit (202) initiates one or more failover procedures by invoking, one or more Application Programming Interface (API) functions.
14. The system (104) as claimed in claim 13, upon invoking the one or more API functions, the framework unit (202) is configured to initiate a kernel cleanup.
15. The system (104) as claimed in claim 9, wherein the framework unit (202) transfers, tasks to be performed by the first node (102) to the second node (108), enables takeover of restarting procedure and managing network connectivity during failure.
16. The system (104) as claimed in claim 13, wherein the second node (108) is selected by the framework unit (202) simultaneously when the one or more API functions are invoked, thereby transitioning from the first node (102) to the second node (108).
17. A non-transitory computer-readable medium having stored thereon computer-readable instructions that, when executed by a processor (204), cause the processor (204) to:
 - receive, a failure trigger from a first node (102);
 - disconnect, a connection between the first node (102) and the one or more processors (204) on receiving the failure trigger;
 - initiate, one or more failover procedures; and

transfer, tasks to be performed by the first node (102) to a second node (108), thereby recovering the tasks and managing network connectivity during failure.

1/4

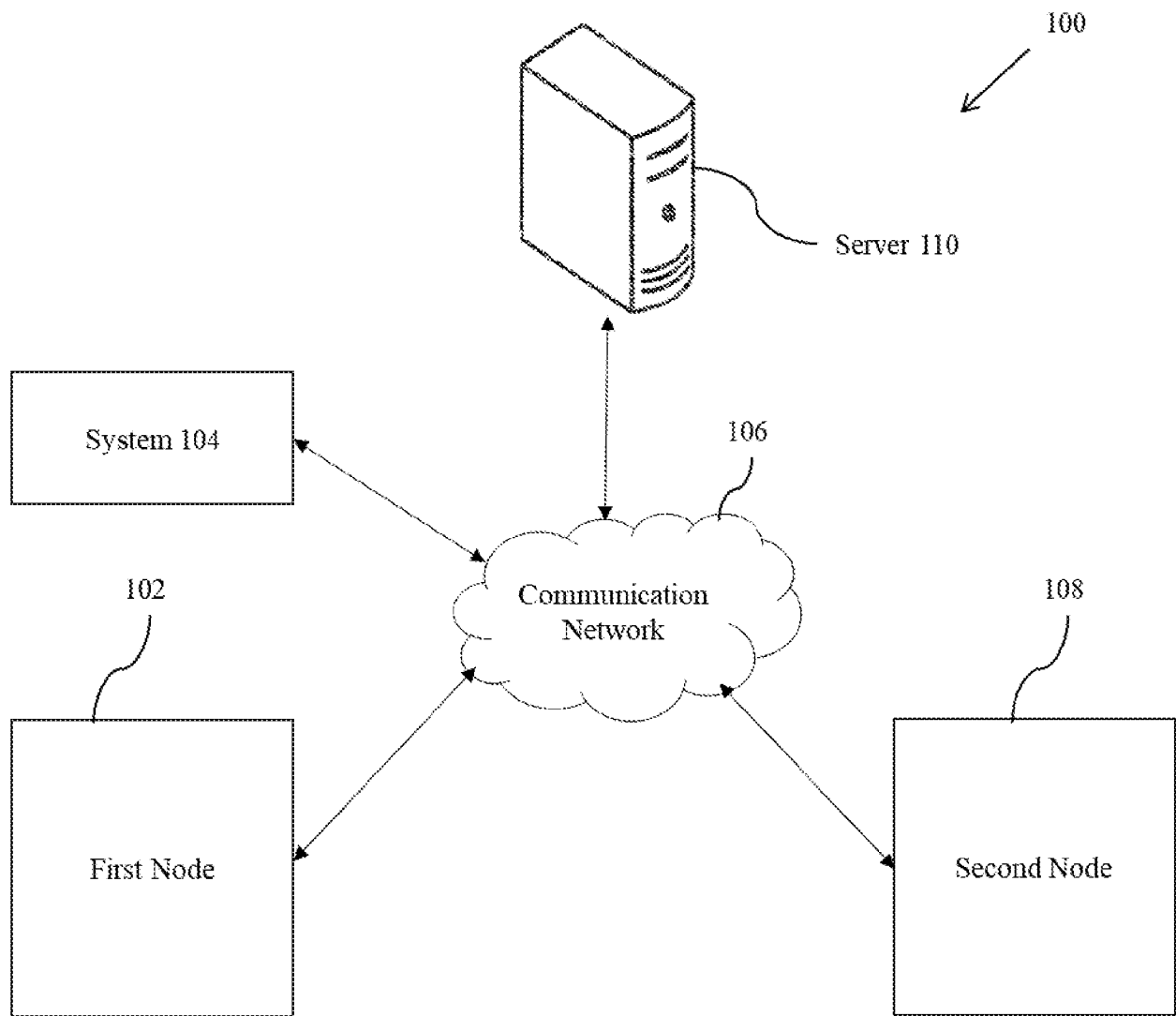


FIG. 1

2/4

System 104

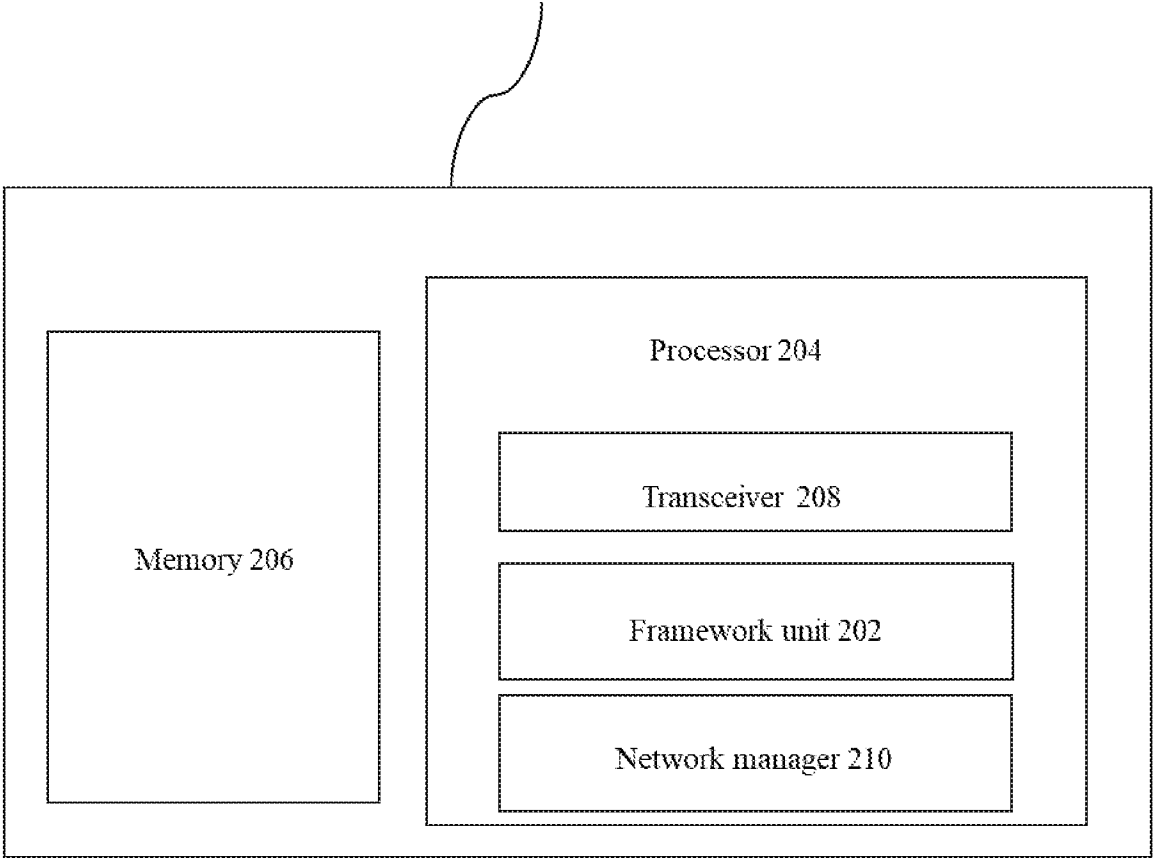


FIG. 2

3/4

300

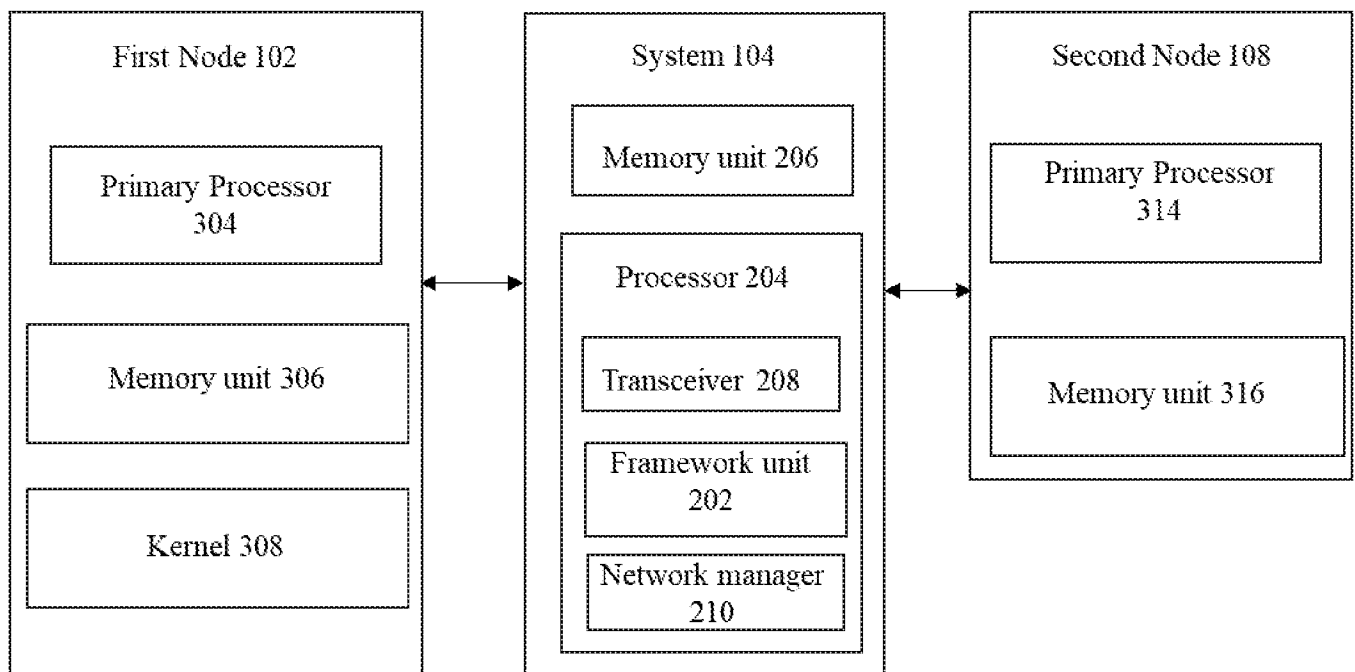


FIG. 3

4/4

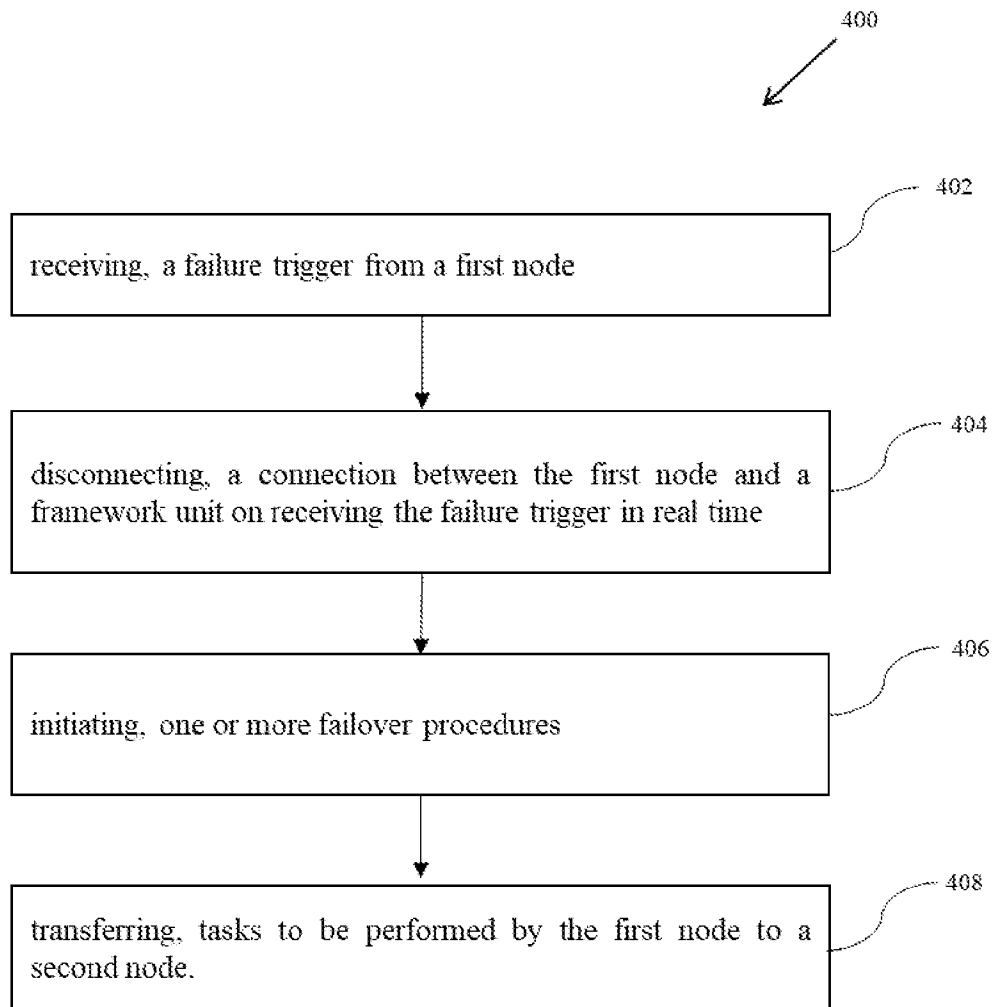


FIG. 4

INTERNATIONAL SEARCH REPORT

International application No.
PCT/IN2024/050957

A. CLASSIFICATION OF SUBJECT MATTER

H04L45/28, H04L43/0811, H04L69/40 Version=2024.01

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

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

H04L

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

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

PatSeer, IPO Internal Database

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
Y	US6490610B1 (ORACLE CORP) 03-DECEMBER-2002 (03-12-2002) lines 38-47 of Column 1, lines 42-51 of Column 2, lines 19-53 of Column 3, line 33 of Column 3 to line 19 of Column 4, line 20 - 32 of Column 4 -----	1-17
Y	WO2014116148A1 (ERICSSON TELEFON AB L M) 31-JULY-2014 (31-07-2014) lines 14-24 on page 11, lines 1-6 on page 7	1-17



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"D" document cited by the applicant in the international application

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"I" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search

28-10-2024

Date of mailing of the international search report

28-10-2024

Name and mailing address of the ISA/

Indian Patent Office

Plot No.32, Sector 14, Dwarka, New Delhi-110075

Facsimile No.

Authorized officer

Himanshu Sourabh

Telephone No. +91-1125300200

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.
PCT/IN2024/050957

Citation	Pub.Date	Family	Pub.Date
WO 2014116148 A1	31-07-2014	EP 2949104 A1	02-12-2015
		US 2015350043 A1	03-02-2015