

HOST INITIATED LINK RESET IN A STORAGE AREA NETWORK

BACKGROUND

[0001] The present invention generally relates to storage area networks (SANs), and more specifically, to host initiated link reset in a SAN.

5 [0002] During normal operation of a SAN, the fibre channel frames, including acknowledgments may become corrupted in transport. Such corruption can be caused by optics failing, bad cables, loose connections, optical budgets not within tolerances, intermittent hardware malfunctions, etc.

[0003] When a receiving side of a fibre channel connection cannot recognize a start of
10 frame (SOF) in the header of the incoming frame, the receiver does not respond with the appropriate acknowledgment. In one example, a sender decrements an available buffer credit by one upon sending the frame but does not receive the corresponding acknowledgment. As a result, the synchronization between the sender and receiver in terms of buffer credits becomes skewed. When this condition occurs, it will continue and
15 cause the number of buffer credits perceived to be available to gradually decrease to the point where it impacts performance, or even lead to link transmission stopping altogether.

[0004] Currently, buffer credit recovery mechanisms provided in fibre channel standards attempt to prevent errors in the synchronization of buffer credits between a sender and receiver from occurring. The standards support two modes of buffer credit
20 recovery: 1) a link reset and 2) a calculation that simply results in replenishing the credits. However, both of the modes of buffer credit recovery are performed exclusively by network devices, such as a switch, and only for the switch ports.

SUMMARY

[0005] Embodiments of the present invention are directed to a computer-implemented
25 method for performing a host-initiated link reset in a storage area network (SAN). A

non-limiting example of the computer-implemented method includes identifying, by a host in communication with the SAN, each link in the SAN, wherein each link is defined by a pair of ports. The method also includes obtaining, by the host, a buffer credit balance for each port in the SAN and calculating, by the host, a buffer credit imbalance
5 for each link in the SAN. The method further includes causing a reset of the link based on a determination that the buffer credit imbalance for a link exceeds a threshold value.

[0006] Embodiments of the present invention are directed to a storage area network (SAN). A non-limiting example of the SAN includes one or more network devices and a plurality of end devices each coupled to at least one of the one or more network devices.
10 One of the plurality of end devices is a host device that is configured to identify each link in the SAN, wherein each link is defined by a pair of ports. The host device is also configured to obtain a buffer credit balance for each port in the SAN and calculate a buffer credit imbalance for each link in the SAN. The host device is further configured to cause a reset of the link based on a determination that the buffer credit imbalance for a
15 link exceeds a threshold value.

[0007] Embodiments of the invention are directed to a computer program product for performing a host-initiated link reset in a storage area network (SAN), the computer program product comprising a computer-readable storage medium having program instructions embodied therewith. The program instructions are executable by a processor
20 to cause the processor to perform a method. A non-limiting example of the method includes identifying, by a host in communication with the SAN, each link in the SAN, wherein each link is defined by a pair of ports. The method also includes obtaining, by the host, a buffer credit balance for each port in the SAN and calculating, by the host, a buffer credit imbalance for each link in the SAN. The method further includes causing a
25 reset of the link based on a determination that the buffer credit imbalance for a link exceeds a threshold value.

[0008] Additional technical features and benefits are realized through the techniques of the present invention. Embodiments and aspects of the invention are described in detail herein and are considered a part of the claimed subject matter. For a better understanding, refer to the detailed description and to the drawings.

5

BRIEF DESCRIPTION OF THE DRAWINGS

[0009] The specifics of the exclusive rights described herein are particularly pointed out and distinctly claimed in the claims at the conclusion of the specification. The foregoing and other features and advantages of the embodiments of the invention are apparent from the following detailed description taken in conjunction with the
10 accompanying drawings in which:

[0010] FIG. 1 is a schematic diagram of a computer system, in accordance with an embodiment of the invention;

[0011] FIG. 2 is a schematic diagram of a storage area network, in accordance with an embodiment of the invention;

15 [0012] FIG. 3 depicts a flow diagram of a method for performing a host-initiated link reset in a storage area network, in accordance with an embodiment of the invention, and

[0013] FIG. 4 depicts a flow diagram of a method for performing a host-initiated link reset in a storage area network, in accordance with another embodiment of the invention.

[0014] The diagrams depicted herein are illustrative. There can be many variations to
20 the diagram or the operations described therein without departing from the spirit of the invention. For instance, the actions can be performed in a differing order or actions can be added, deleted or modified. Also, the term “coupled” and variations thereof describes having a communications path between two elements and does not imply a direct connection between the elements with no intervening elements/connections between
25 them. All of these variations are considered a part of the specification.

DETAILED DESCRIPTION

- [0015] Exemplary embodiments include methods, systems and computer program product for performing a host-initiated link reset in a storage area network (SAN). In exemplary embodiments, a host device that is connected to the SAN is configured to
- 5 monitor the buffer credits reported as available on all ports in the SAN, including channel ports and switch ports. The host device is further configured to cause a link reset in the SAN upon determining that a buffer credit imbalance between two ports on either end of a link exceeds a threshold value. The link reset forces the buffer credit counters for both ports of the link to return to their full default value.
- 10 [0016] Turning now to FIG 1, a computer system 100 is generally shown in accordance with an embodiment. The computer system 100 can be an electronic, computer framework comprising and/or employing any number and combination of computing devices and networks utilizing various communication technologies, as described herein. The computer system 100 can be easily scalable, extensible, and
- 15 modular, with the ability to change to different services or reconfigure some features independently of others. The computer system 100 may be, for example, a server, desktop computer, laptop computer, tablet computer, or smartphone. In some examples, computer system 100 may be a cloud computing node. Computer system 100 may be described in the general context of computer system executable instructions, such as
- 20 program modules, being executed by a computer system. Generally, program modules may include routines, programs, objects, components, logic, data structures, and so on that perform particular tasks or implement particular abstract data types. Computer system 100 may be practiced in distributed cloud computing environments where tasks are performed by remote processing devices that are linked through a communications
- 25 network. In a distributed cloud computing environment, program modules may be located in both local and remote computer system storage media including memory storage devices.

[0017] As shown in FIG 1, the computer system 100 has one or more central processing units (CPU(s)) 101a, 101b, 101c, etc. (collectively or generically referred to as processor(s) 101). The processors 101 can be a single-core processor, multi-core processor, computing cluster, or any number of other configurations. The processors 101, also referred to as processing circuits, are coupled via a system bus 102 to a system memory 103 and various other components. The system memory 103 can include a read only memory (ROM) 104 and a random access memory (RAM) 105. The ROM 104 is coupled to the system bus 102 and may include a basic input/output system (BIOS), which controls certain basic functions of the computer system 100. The RAM is read-write memory coupled to the system bus 102 for use by the processors 101. The system memory 103 provides temporary memory space for operations of said instructions during operation. The system memory 103 can include random access memory (RAM), read only memory, flash memory, or any other suitable memory systems.

[0018] The computer system 100 comprises an input/output (I/O) adapter 106 and a communications adapter 107 coupled to the system bus 102. The I/O adapter 106 may be a small computer system interface (SCSI) adapter that communicates with a hard disk 108 and/or any other similar component. The I/O adapter 106 and the hard disk 108 are collectively referred to herein as a mass storage 110.

[0019] Software 111 for execution on the computer system 100 may be stored in the mass storage 110. The mass storage 110 is an example of a tangible storage medium readable by the processors 101, where the software 111 is stored as instructions for execution by the processors 101 to cause the computer system 100 to operate, such as is described herein below with respect to the various Figures. Examples of computer program product and the execution of such instruction is discussed herein in more detail. The communications adapter 107 interconnects the system bus 102 with a network 112, which may be an outside network, enabling the computer system 100 to communicate with other such systems. In one embodiment, a portion of the system memory 103 and the mass storage 110 collectively store an operating system, which may be any

appropriate operating system, such as the z/OS or AIX operating system from IBM Corporation, to coordinate the functions of the various components shown in FIG 1.

[0020] Additional input/output devices are shown as connected to the system bus 102 via a display adapter 115 and an interface adapter 116 and. In one embodiment, the
5 adapters 106, 107, 115, and 116 may be connected to one or more I/O buses that are connected to the system bus 102 via an intermediate bus bridge (not shown). A display 119 (e.g., a screen or a display monitor) is connected to the system bus 102 by a display adapter 115, which may include a graphics controller to improve the performance of graphics intensive applications and a video controller. A keyboard 121, a mouse 122, a
10 speaker 123, etc. can be interconnected to the system bus 102 via the interface adapter 116, which may include, for example, a Super I/O chip integrating multiple device adapters into a single integrated circuit. Suitable I/O buses for connecting peripheral devices such as hard disk controllers, network adapters, and graphics adapters typically include common protocols, such as the Peripheral Component Interconnect
15 (PCI). Thus, as configured in FIG 1, the computer system 100 includes processing capability in the form of the processors 101, and, storage capability including the system memory 103 and the mass storage 110, input means such as the keyboard 121 and the mouse 122, and output capability including the speaker 123 and the display 119.

[0021] In some embodiments, the communications adapter 107 can transmit data using
20 any suitable interface or protocol, such as the internet small computer system interface, among others. The network 112 may be a cellular network, a radio network, a wide area network (WAN), a local area network (LAN), or the Internet, among others. An external computing device may connect to the computer system 100 through the network 112. In some examples, an external computing device may be an external webserver or a cloud
25 computing node.

[0022] It is to be understood that the block diagram of FIG 1 is not intended to indicate that the computer system 100 is to include all of the components shown in FIG 1. Rather, the computer system 100 can include any appropriate fewer or additional components not

illustrated in FIG 1 (e.g., additional memory components, embedded controllers, modules, additional network interfaces, etc.). Further, the embodiments described herein with respect to computer system 100 may be implemented with any appropriate logic, wherein the logic, as referred to herein, can include any suitable hardware (e.g., a processor, an
5 embedded controller, or an application specific integrated circuit, among others), software (e.g., an application, among others), firmware, or any suitable combination of hardware, software, and firmware, in various embodiments.

[0023] FIG. 2 is a diagram of a storage area network (SAN) 200, in accordance with an embodiment. SAN 200 includes a plurality of end devices 210, 220 that are each
10 connected to one or more network devices 230, such as switches. Network devices 230 include memory circuitry and processing circuitry (e.g. including monitoring circuitry and limiting circuitry). The network topology of SAN 200 may be a switched fabric, where network devices 230 are switches. In some embodiments, a particular switch 230 may be set to switch mode or N-port ID virtualization (NPIV) mode. In exemplary
15 embodiments, at least one of the end devices 210, 220 in the SAN device 210 is configured as a host device 210. In one embodiment, the host device 210 is embodied in a computer system, such as the one shown in FIG. 1.

[0024] In exemplary embodiments, each of the network devices 230 includes a plurality of ports that connect the network devices 230 to various end devices 210, 220
20 via links 225. In addition, each end device 210, 220 includes one or more ports that connect the end devices to one or more network devices 230 or to other end devices 220. In exemplary embodiments, each link 225 is defined by the two ports that are connected to the link 255. The ports of end devices 220, 210 are referred to as channel ports while the ports of the network devices 230 are referred to as switch ports.

25 **[0025]** In exemplary embodiments, the network devices 230 support the Fibre Channel (FC) and/or Fibre channel over Ethernet (FCoE) Protocols. For example, a particular fixed port network device may support the FC protocol and/or the FCoE protocol. As another example, if a particular network device includes a plurality of line

cards, at least some of which may support the FC protocol and/or at least some of which may support the FCoE protocol. It is noted that a particular port on a particular network device 230 may support the FC protocol or the FCoE protocol by default or as configured (e.g. if the particular port is a universal port). Optionally, network devices 230 support
5 one or more other protocols such as simple network management protocol (SNMP) for collecting information, e.g. for output on management device 210.

[0026] Referring now to FIG. 3, a flow diagram of a method 300 for performing a host-initiated link reset in a storage area network, in accordance with an embodiment of the invention is shown. As shown at block 302, the method 300 includes identifying, by
10 a host in communication with the SAN, each link in the SAN. In exemplary embodiments, each link in the SAN is defined by a pair of ports, which are one of a channel port and a switch port or in the case of inter-switch links it can be two switch ports. The method 300 also includes obtaining, by the host, a buffer credit balance for each port in the SAN, as shown at block 304. Next, as shown at block 306, the
15 method 300 includes calculating, by the host, a buffer credit imbalance for each link in the SAN. In one embodiment, the buffer credit imbalance is calculated by determining a difference between a buffer credit balance of the two ports of the link. In another embodiment, the buffer credit imbalance is calculated by determining a percentage difference between a buffer credit balance of the two ports of the link.

20 [0027] The method 300 also includes determining if the buffer credit imbalance for a link exceeds a threshold value, as shown at decision block 308. If the buffer credit imbalance for a link does exceed a threshold value, the method 300 proceeds to block 310 and causes a reset of the link. Otherwise, the method 300 returns to block 304. In exemplary embodiments, the threshold value is set by a user of the host device. In
25 exemplary embodiments, obtaining the buffer credit balance for each port in the SAN is performed on a periodic basis. For example, in one embodiment, the host device is configured to obtain the buffer credit balance for each port in the SAN every fifteen minutes, or at another predetermined interval that is set by a user of the host device.

[0028] In one embodiment, the determination that the buffer credit imbalance for the link exceeds the threshold value includes determining that a difference between the buffer credit balance for each port of the pair of ports is greater than a predetermined number. In one example, a link that is defined by a first port and a second port and the buffer credit balance for the first port is eight and the buffer credit balance for the first port is five. If the threshold is two, then the link will be reset. However, if the threshold is three the link will not be reset at this time.

[0029] In another embodiment, the threshold value is a predetermined percentage and the buffer credit imbalance is calculated by dividing a larger buffer credit balance of the pair of ports of the link by a smaller buffer credit balance of the pair of ports of the link and subtracting one. In one example, a link that is defined by a first port and a second port and the buffer credit balance for the first port is eight, the buffer credit balance for the first port is five, and the threshold value is fifty percent or .5. In this example, the buffer credit imbalance for the link is calculated as $(8/5) - 1 = .6$ and therefore the link will be reset.

[0030] In exemplary embodiments, once it is determined that a link needs to be reset the action that is taken to reset the link depends upon a type of the ports that define the link. In one embodiment, based on the determination that at least one of the pair of ports for the link is a channel port, the host device causes the link to be reset by issuing a command to the channel port to reset the link. In another embodiment, based on the determination that neither of the pair of ports is a channel port, the host device causes the link to be reset by instructing a network device in the SAN associated with the link to perform a reset of the link. In one embodiment, the instruction is sent to the network device using the Control Unit Protocol (CUP).

[0031] In exemplary embodiments, the SAN utilizes fibre channels for communication among the nodes. Fibre channels are configured to send packets of data referred to as frames, which include frame delimiters that demarcate the start of frame (SOF) or the end of frame (EOF). Primitive signals-represent events on the sending port include an

- acknowledgment signal (R_RDY) and buffer-to-buffer state change (BB_SC) signals. A BB_RDY_N variable is commonly used to maintain a count of R_RDY primitives received and BB_FRM_N maintains a count of frames received. The Buffer to Buffer State Change Number (BB_SC_N) reflects the value specifying the number of frames
- 5 between BB_SCs primitive signals and the number of R_RDYs between BB_SCr primitive signals. The Buffer to Buffer State Change-R_RDY (BB_SCr) signal is transmitted periodically to enable the attached port to determine if any R_RDYs have been lost. A Buffer to Buffer State Change-SOF (BB_SCs) signal is transmitted to enable an attached port to determine if any frames have been lost.
- 10 **[0032]** In exemplary embodiments, the BB_SC_N (Buffer to Buffer state change number) is used to track an agreed-upon buffer credit maximum for a link between two ports. For example, when two ports are connected, during login (FLOGI or PLOGI) the ports perform an exchange link parameters (ELP) and send each other an internal link services (ILS) frame carrying their value. If the two ports have different values, then the
- 15 larger of the two values is used by both ports. Typically, BB_SC_N ranges from 1 to 15. In exemplary embodiments, a variable F is used to specify a number of frames to be sent between BB_SCs and the number of R_RDYs sent between BB_SCr primitive signals. In one embodiment F is defined as $2^{BB_SC_N}$. In other words, to establish checkpoints a BB_SCs signal is sent after every F frames and a BB_SCr is sent after
- 20 every F R_RDYs. In one example, BB_SC_N=3, $\Rightarrow$ F=8. In this example, each port will send BB_SCs after every 8 frames that are subject to buffer to buffer flow control and each port will send BB_SCr after every 8 R_RDYs.

- [0033]** In exemplary embodiments, each port maintains a count of the number of frames, for example BB_FRM_N, received between the completion of login and receipt
- 25 of the first BB_SCs. In addition, each port also maintains a count between successive occurrences of BB_SCs. When the count of the number of frames reaches F, it wraps back to zero and starts over. However, if the counter is not at zero when the BB_SCs arrives, it means that one or more of the frames were lost.

[0034] In exemplary embodiments, each port maintains a count of the number of R_RDYs, for example BB_RDY_N, received between completion of login and receipt of the first BB_SCr. Each port also maintains a count between successive occurrences of BB_SCr. When the count of the number of R_RDYs reaches the computed value F, it wraps back to zero and starts over. However, if the count of the number of R_RDYs received is not zero when BB_SCr is received, one or more R_RDYs has been lost.

[0035] Referring now to FIG. 4 depicts a flow diagram of a method 400 for performing a host-initiated link reset in a storage area network in accordance with another embodiment of the invention. As shown at block 402, a BB_SCr or BB_SCs arrives at port. Next, as shown at decision block 404, it is determined if BB_RDY_N or BB_FRM_N are equal to zero. If BB_RDY_N or BB_FRM_N are equal to zero, the method 400 proceeds to block 406 and continues normal operations of the SAN. However, if one of BB_RDY_N and BB_FRM_N are not equal to zero, it is determined that one or more frames or R_RDYs were lost and the method 400 proceed to decision block 408.

[0036] As will be understood by those of ordinary skill in the art, BB_RDY_N and BB_FRM_N are different embodiments of the counting/tracking implementation allowed in the fibre channel standards. In exemplary embodiments, it is normal for these counters to have a zero value and no action required. However, when these counters have a non-zero values, it indicates R_RDY primitives were lost/unaccounted for, and hence the number of buffer credits available on the receiving half of the link will no longer be their full/expected configured value. In exemplary embodiments, a threshold value is used to trigger to determine when to execute the buffer credit recovery. In different embodiments, the threshold trigger may be a user-configurable value based on a percentage of the port's login BB_Credit value, 0 buffer credits remaining, buffer credits below a specified value (for example, less than 5) or a number of instances of frame pacing delay being reported via the RMF (Resource Measurement Facility) 74-7 (FICON Director Activity report) during the configured RMF interval.

[0037] Continuing with reference to FIG. 4, at decision block 408 it is determined if the threshold for buffer credit loss triggered. If so, the method 400 proceeds to decision block 410. Otherwise, the method 400 proceeds to block 406 and continues normal operations of the SAN. At decision block 410, it is determined if one or more of the ports
5 of the link are a channel port, or a directly attached CU port. If one or more of the ports of the link are a channel port, or a directly attached CU port, the method proceeds to block 412 and the host device issues command to channel port to perform link reset (LR). Otherwise, the method proceeds to block 414 and the host device issues a command (via FICON CUP or SMI-S) to perform a LR. The method 400 concludes at block 416 when
10 the buffer credits on the link being restored. In exemplary embodiments, during a LR, frames are not discarded, they are held in buffers until the LR concludes and the LR resets all involved counters to initial values, in effect, "replenishing" "lost" Buffer Credits.

[0038] Various embodiments of the invention are described herein with reference to
15 the related drawings. Alternative embodiments of the invention can be devised without departing from the scope of this invention. Various connections and positional relationships (e.g., over, below, adjacent, etc.) are set forth between elements in the following description and in the drawings. These connections and/or positional relationships, unless specified otherwise, can be direct or indirect, and the present
20 invention is not intended to be limiting in this respect. Accordingly, a coupling of entities can refer to either a direct or an indirect coupling, and a positional relationship between entities can be a direct or indirect positional relationship. Moreover, the various tasks and process steps described herein can be incorporated into a more comprehensive procedure or process having additional steps or functionality not described in detail
25 herein.

[0039] The following definitions and abbreviations are to be used for the interpretation of the claims and the specification. As used herein, the terms "comprises," "comprising," "includes," "including," "has," "having," "contains" or "containing," or any other

variation thereof, are intended to cover a non-exclusive inclusion. For example, a composition, a mixture, process, method, article, or apparatus that comprises a list of elements is not necessarily limited to only those elements but can include other elements not expressly listed or inherent to such composition, mixture, process, method, article, or apparatus.

[0040] Additionally, the term "exemplary" is used herein to mean "serving as an example, instance or illustration." Any embodiment or design described herein as "exemplary" is not necessarily to be construed as preferred or advantageous over other embodiments or designs. The terms "at least one" and "one or more" may be understood to include any integer number greater than or equal to one, i.e. one, two, three, four, etc. The terms "a plurality" may be understood to include any integer number greater than or equal to two, i.e. two, three, four, five, etc. The term "connection" may include both an indirect "connection" and a direct "connection."

[0041] The terms "about," "substantially," "approximately," and variations thereof, are intended to include the degree of error associated with measurement of the particular quantity based upon the equipment available at the time of filing the application. For example, "about" can include a range of $\pm 8\%$ or 5% , or 2% of a given value.

[0042] For the sake of brevity, conventional techniques related to making and using aspects of the invention may or may not be described in detail herein. In particular, various aspects of computing systems and specific computer programs to implement the various technical features described herein are well known. Accordingly, in the interest of brevity, many conventional implementation details are only mentioned briefly herein or are omitted entirely without providing the well-known system and/or process details.

[0043] The present invention may be a system, a method, and/or a computer program product at any possible technical detail level of integration. The computer program product may include a computer readable storage medium (or media) having computer

readable program instructions thereon for causing a processor to carry out aspects of the present invention.

- [0044]** The computer readable storage medium can be a tangible device that can retain and store instructions for use by an instruction execution device. The computer readable storage medium may be, for example, but is not limited to, an electronic storage device, a magnetic storage device, an optical storage device, an electromagnetic storage device, a semiconductor storage device, or any suitable combination of the foregoing. A non-exhaustive list of more specific examples of the computer readable storage medium includes the following: a portable computer diskette, a hard disk, a random access memory (RAM), a read-only memory (ROM), an erasable programmable read-only memory (EPROM or Flash memory), a static random access memory (SRAM), a portable compact disc read-only memory (CD-ROM), a digital versatile disk (DVD), a memory stick, a floppy disk, a mechanically encoded device such as punch-cards or raised structures in a groove having instructions recorded thereon, and any suitable combination of the foregoing. A computer readable storage medium, as used herein, is not to be construed as being transitory signals per se, such as radio waves or other freely propagating electromagnetic waves, electromagnetic waves propagating through a waveguide or other transmission media (e.g., light pulses passing through a fiber-optic cable), or electrical signals transmitted through a wire.
- [0045]** Computer readable program instructions described herein can be downloaded to respective computing/processing devices from a computer readable storage medium or to an external computer or external storage device via a network, for example, the Internet, a local area network, a wide area network and/or a wireless network. The network may comprise copper transmission cables, optical transmission fibers, wireless transmission, routers, firewalls, switches, gateway computers and/or edge servers. A network adapter card or network interface in each computing/processing device receives computer readable program instructions from the network and forwards the computer readable

program instructions for storage in a computer readable storage medium within the respective computing/processing device.

[0046] Computer readable program instructions for carrying out operations of the present invention may be assembler instructions, instruction-set-architecture (ISA) instructions, machine instructions, machine dependent instructions, microcode, firmware instructions, state-setting data, configuration data for integrated circuitry, or either source code or object code written in any combination of one or more programming languages, including an object oriented programming language such as Smalltalk, C++, or the like, and procedural programming languages, such as the "C" programming language or similar programming languages. The computer readable program instructions may execute entirely on the user's computer, partly on the user's computer, as a stand-alone software package, partly on the user's computer and partly on a remote computer or entirely on the remote computer or server. In the latter scenario, the remote computer may be connected to the user's computer through any type of network, including a local area network (LAN) or a wide area network (WAN), or the connection may be made to an external computer (for example, through the Internet using an Internet Service Provider). In some embodiments, electronic circuitry including, for example, programmable logic circuitry, field-programmable gate arrays (FPGA), or programmable logic arrays (PLA) may execute the computer readable program instruction by utilizing state information of the computer readable program instructions to personalize the electronic circuitry, in order to perform aspects of the present invention.

[0047] Aspects of the present invention are described herein with reference to flowchart illustrations and/or block diagrams of methods, apparatus (systems), and computer program products according to embodiments of the invention. It will be understood that each block of the flowchart illustrations and/or block diagrams, and combinations of blocks in the flowchart illustrations and/or block diagrams, can be implemented by computer readable program instructions.

[0048] These computer readable program instructions may be provided to a processor of a general purpose computer, special purpose computer, or other programmable data processing apparatus to produce a machine, such that the instructions, which execute via the processor of the computer or other programmable data processing apparatus, create
5 means for implementing the functions/acts specified in the flowchart and/or block diagram block or blocks. These computer readable program instructions may also be stored in a computer readable storage medium that can direct a computer, a programmable data processing apparatus, and/or other devices to function in a particular manner, such that the computer readable storage medium having instructions stored
10 therein comprises an article of manufacture including instructions which implement aspects of the function/act specified in the flowchart and/or block diagram block or blocks.

[0049] The computer readable program instructions may also be loaded onto a computer, other programmable data processing apparatus, or other device to cause a
15 series of operational steps to be performed on the computer, other programmable apparatus or other device to produce a computer implemented process, such that the instructions which execute on the computer, other programmable apparatus, or other device implement the functions/acts specified in the flowchart and/or block diagram block or blocks.

20 [0050] The flowchart and block diagrams in the Figures illustrate the architecture, functionality, and operation of possible implementations of systems, methods, and computer program products according to various embodiments of the present invention. In this regard, each block in the flowchart or block diagrams may represent a module, segment, or portion of instructions, which comprises one or more executable instructions
25 for implementing the specified logical function(s). In some alternative implementations, the functions noted in the blocks may occur out of the order noted in the Figures. For example, two blocks shown in succession may, in fact, be executed substantially concurrently, or the blocks may sometimes be executed in the reverse order, depending

upon the functionality involved. It will also be noted that each block of the block diagrams and/or flowchart illustration, and combinations of blocks in the block diagrams and/or flowchart illustration, can be implemented by special purpose hardware-based systems that perform the specified functions or acts or carry out combinations of special purpose hardware and computer instructions.

[0051] The descriptions of the various embodiments of the present invention have been presented for purposes of illustration, but are not intended to be exhaustive or limited to the embodiments disclosed. Many modifications and variations will be apparent to those of ordinary skill in the art without departing from the scope and spirit of the described embodiments. The terminology used herein was chosen to best explain the principles of the embodiments, the practical application or technical improvement over technologies found in the marketplace, or to enable others of ordinary skill in the art to understand the embodiments described herein.

WHAT IS CLAIMED IS:

1. A computer-implemented method for performing a host-initiated link reset in a storage area network (SAN), the method comprising;

identifying, by a host in communication with the SAN, each link in the SAN, wherein each link is defined by a pair of ports;

obtaining, by the host, a buffer credit balance for each port in the SAN;

calculating, by the host, a buffer credit imbalance for each link in the SAN; and

based on a determination that the buffer credit imbalance for a link exceeds a threshold value, causing a reset of the link, wherein the threshold value is a predetermined percentage and the buffer credit imbalance is calculated by dividing a larger buffer credit balance of the pair of ports of the link by a smaller buffer credit balance of the pair of ports of the link and subtracting one.

2. The computer-implemented method of claim 1, wherein obtaining the buffer credit balance for each port in the SAN is performed on a periodic basis.

3. The computer-implemented method of claim 2, wherein a frequency of the periodic basis is configurable by a user of the host.

4. The computer-implemented method of claim 1, wherein the determination that the buffer credit imbalance for the link exceeds the threshold value comprises determining that a difference between the buffer credit balance for each port of the pair of ports is greater than a predetermined number.

5. The computer-implemented method of claim 1, wherein the threshold value is configurable by a user of the host.

6. The computer-implemented method of claim 1, further comprising:

based on the determination that the buffer credit imbalance for the link exceeds the threshold value and that at least one of the pair of ports for the link is a channel port, issuing, by the host, a command to the channel port to reset the link.

7. The computer-implemented method of claim 1, further comprising:

based on the determination that the buffer credit imbalance for the link exceeds the threshold value and that neither of the pair of ports is a channel port, instructing a network device in the SAN associated with the link to perform a reset of the link.

8. A storage area network (SAN) comprising:

- one or more network devices;

- a plurality of end devices each coupled to at least one of the one or more network devices, wherein one of the plurality of end devices is a host device that is configured to:

 - identify each link in the SAN, wherein each link is defined by a pair of ports;

 - obtain a buffer credit balance for each port in the SAN;

 - calculate a buffer credit imbalance for each link in the SAN; and

 - based on a determination that the buffer credit imbalance for a link exceeds a threshold value, cause a reset of the link, wherein the threshold value is a predetermined percentage and the buffer credit imbalance is calculated by dividing a larger buffer credit balance of the pair of ports of the link by a smaller buffer credit balance of the pair of ports of the link and subtracting one.

9. The SAN of claim 8, wherein obtaining the buffer credit balance for each port in the SAN is performed on a periodic basis.

10. The SAN of claim 9, wherein a frequency of the periodic basis is configurable by a user of the host.

11. The SAN of claim 8, wherein the determination that the buffer credit imbalance for the link exceeds the threshold value comprises determining that a difference between the buffer credit balance for each port of the pair of ports is greater than a predetermined number.

12. The SAN of claim 8, wherein the threshold value is configurable by a user of the host.

13. The SAN of claim 8, wherein the host device that is further configured to:

based on the determination that the buffer credit imbalance for the link exceeds the threshold value and that at least one of the pair of ports for the link is a channel port, issue, by the host, a command to the channel port to reset the link.

14. The SAN of claim 8, wherein the host device that is further configured to:
based on the determination that the buffer credit imbalance for the link exceeds the threshold value and that neither of the pair of ports is a channel port, instruct a network device in the SAN associated with the link to perform a reset of the link.

15. A computer program product for performing a host-initiated link reset in a storage area network (SAN), the computer program product comprising:

a computer-readable storage medium having program instructions embodied therewith,
the program instructions readable by a processor to cause the processor to:

identify, by a host in communication with the SAN, each link in the SAN, wherein each link is defined by a pair of ports;

obtain, by the host, a buffer credit balance for each port in the SAN;

calculate, by the host, a buffer credit imbalance for each link in the SAN; and

based on a determination that the buffer credit imbalance for a link exceeds a threshold value, causing a reset of the link, wherein the threshold value is a predetermined percentage and the buffer credit imbalance is calculated by dividing a larger buffer credit balance of the pair of ports of the link by a smaller buffer credit balance of the pair of ports of the link and subtracting one.

16. The computer program product of claim 15, wherein obtaining the buffer credit balance for each port in the SAN is performed on a periodic basis.

17. The computer program product of claim 16, wherein a frequency of the periodic basis is configurable by a user of the host.

18. The computer program product of claim 15, wherein the determination that the buffer credit imbalance for the link exceeds the threshold value comprises determining that a difference between the buffer credit balance for each port of the pair of ports is greater than a predetermined number.