



(19) **United States**

(12) **Patent Application Publication**  
**Bradford et al.**

(10) **Pub. No.: US 2009/0169202 A1**

(43) **Pub. Date: Jul. 2, 2009**

(54) **LINK DISCOVERY AND VERIFICATION USING LOSS OF LIGHT**

**Publication Classification**

(76) Inventors: **Richard Bradford**, Westford, MA (US); **Daniel C. Tappan**, Boxborough, MA (US)

(51) **Int. Cl.**  
**H04B 10/08** (2006.01)  
**H04B 17/00** (2006.01)  
(52) **U.S. Cl.** ..... **398/16**; 398/34; 398/19

Correspondence Address:  
**EDELL, SHAPIRO & FINNAN, LLC**  
**1901 RESEARCH BOULEVARD, SUITE 400**  
**ROCKVILLE, MD 20850 (US)**

(57) **ABSTRACT**

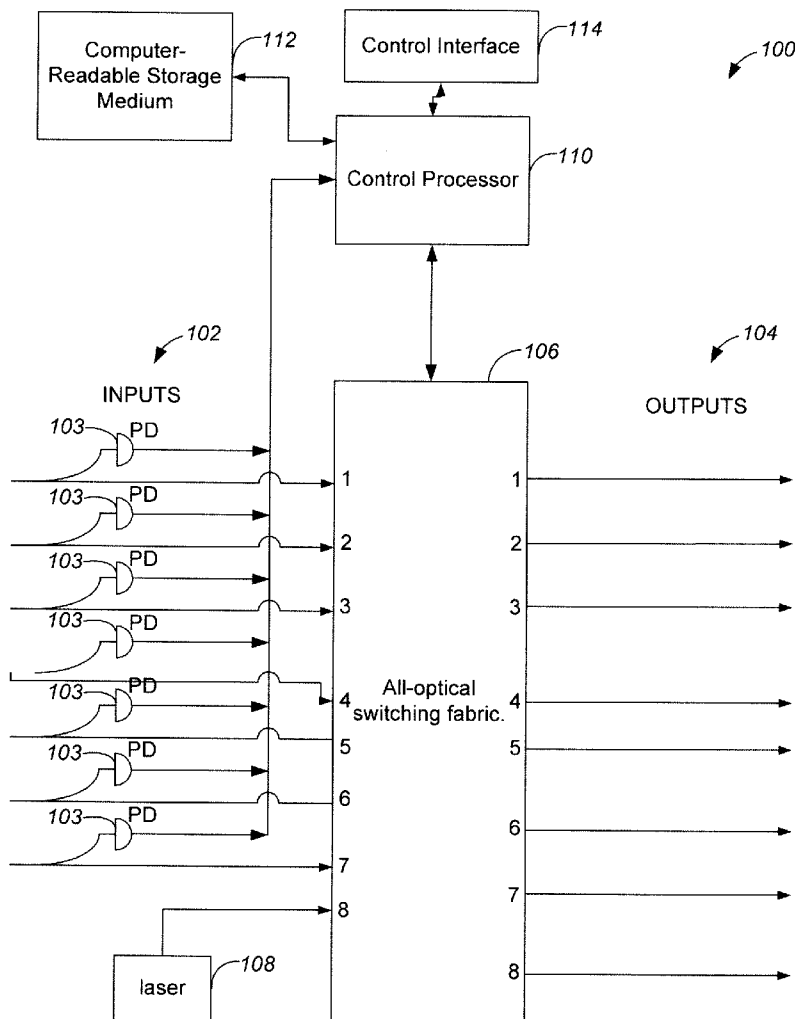
Systems and methods for link discovery and verification that minimize the need for line termination resources to generate and interpret packets. According to one aspect of the present invention, a method for operating a first node in a data communication network to verify connectivity to a second node includes sending a request for verification of connectivity to the second node that identifies an IP address of the first node, a port of the first node, and an IP address of the second node. The method also includes toggling a signal emitted by the port, and notifying the second node of a toggling mode of the port. Finally, the method includes receiving a first message from the second node indicating whether the second node detected the toggling. The request for verification of connectivity is sent on a control channel via a control message separate from the signal and the first message.

(21) Appl. No.: **12/403,649**

(22) Filed: **Mar. 13, 2009**

**Related U.S. Application Data**

(63) Continuation of application No. 11/276,158, filed on Feb. 16, 2006, now abandoned, which is a continuation of application No. 09/968,096, filed on Sep. 28, 2001, now Pat. No. 7,046,928.



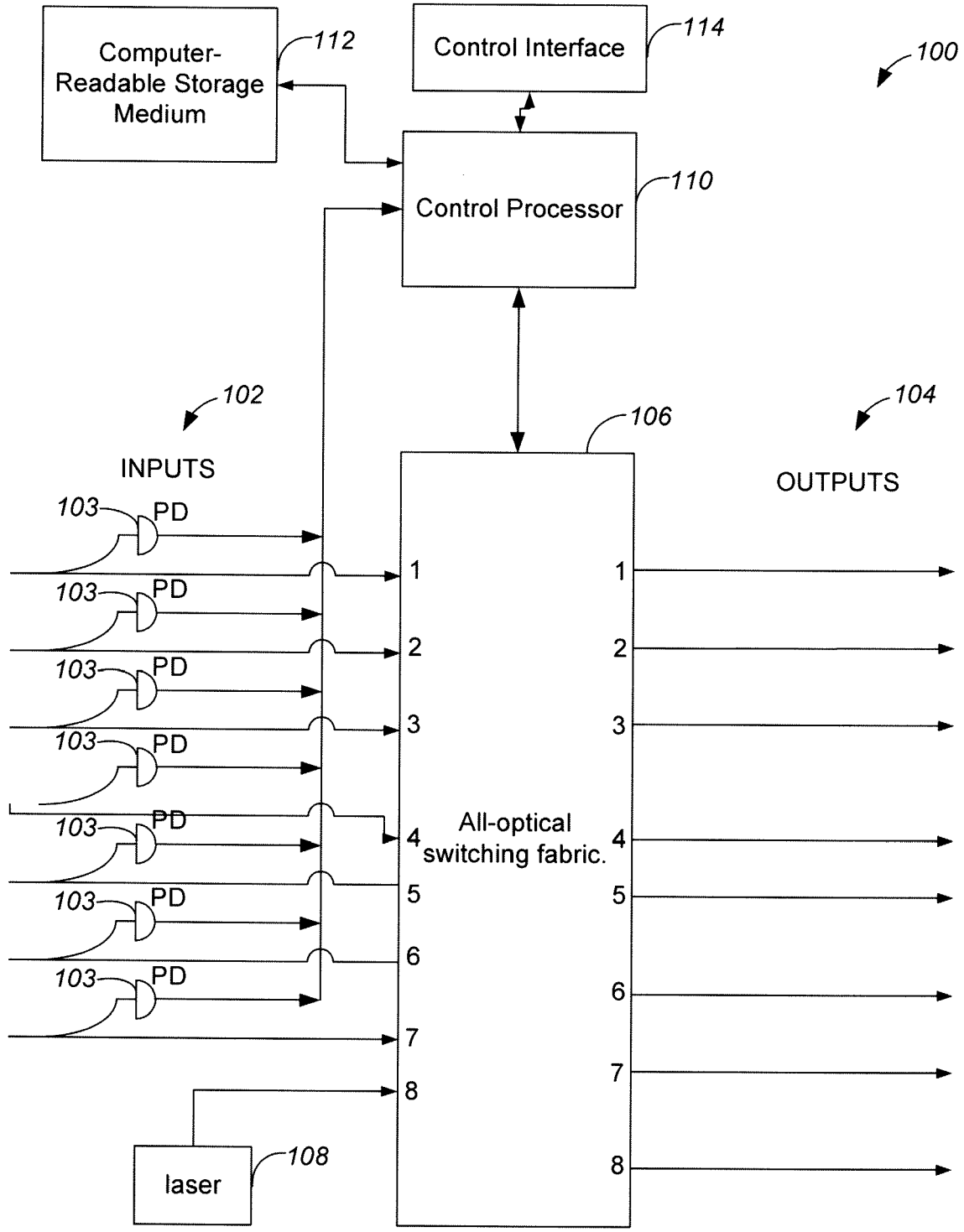


Fig. 1

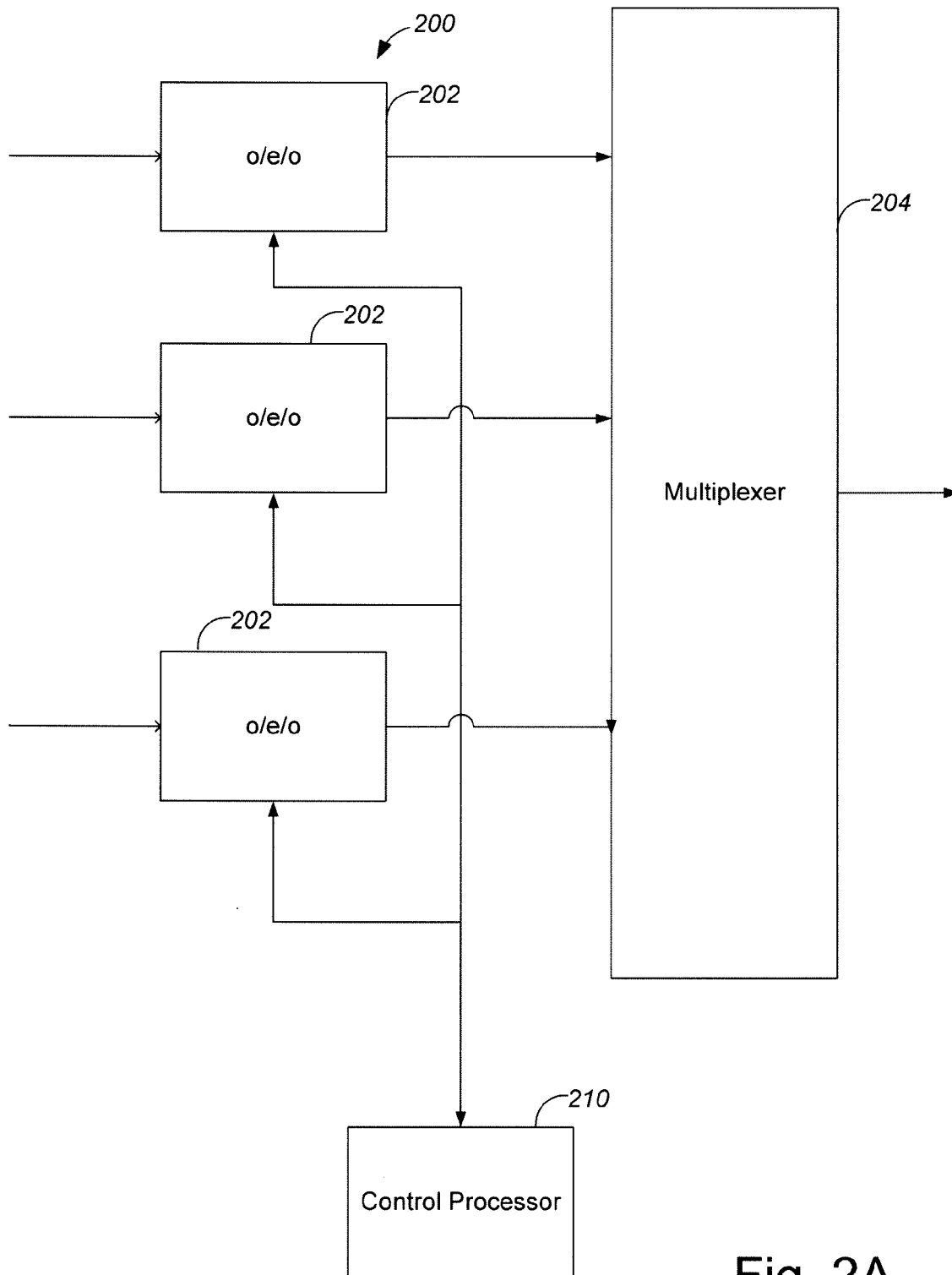


Fig. 2A

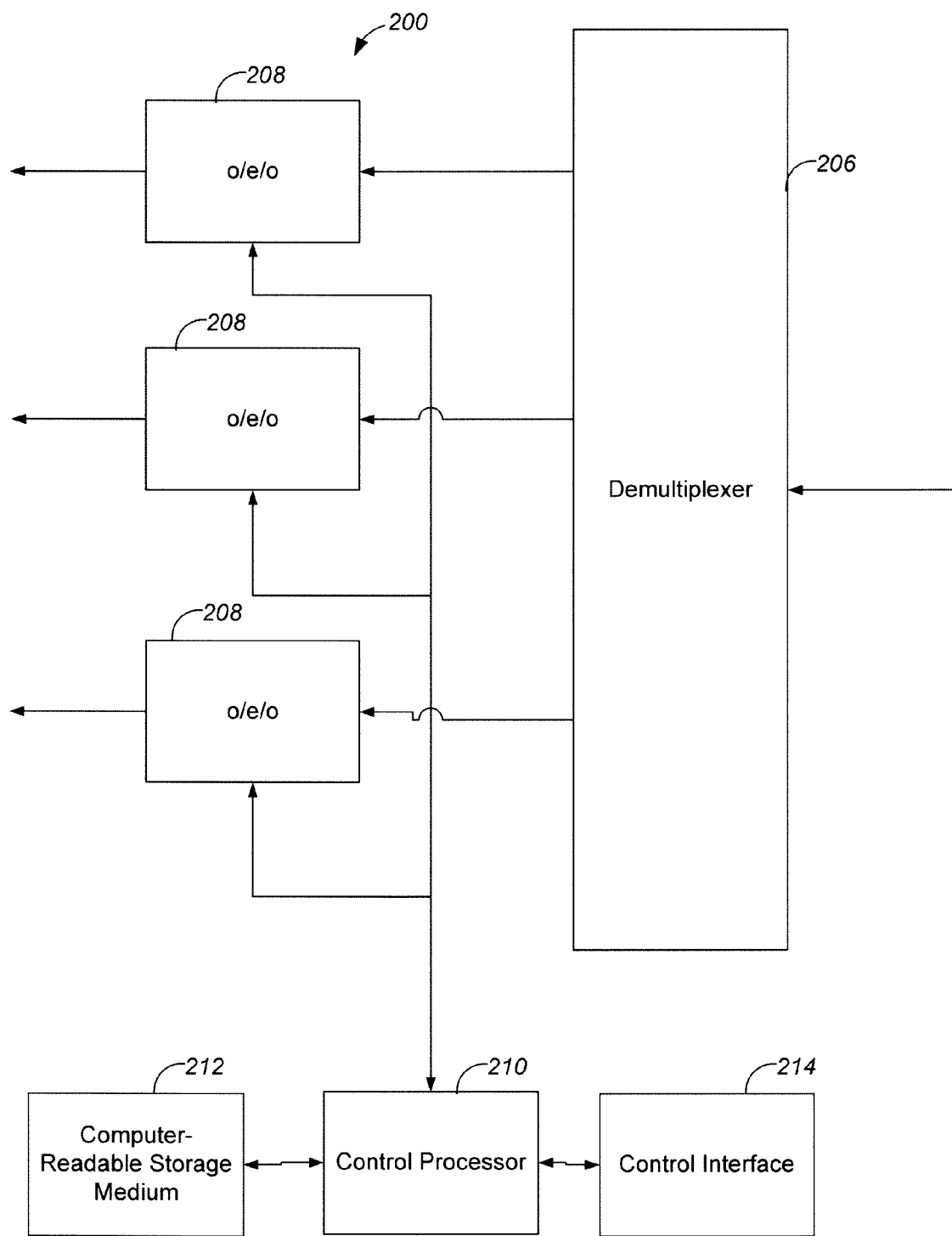


Fig. 2B

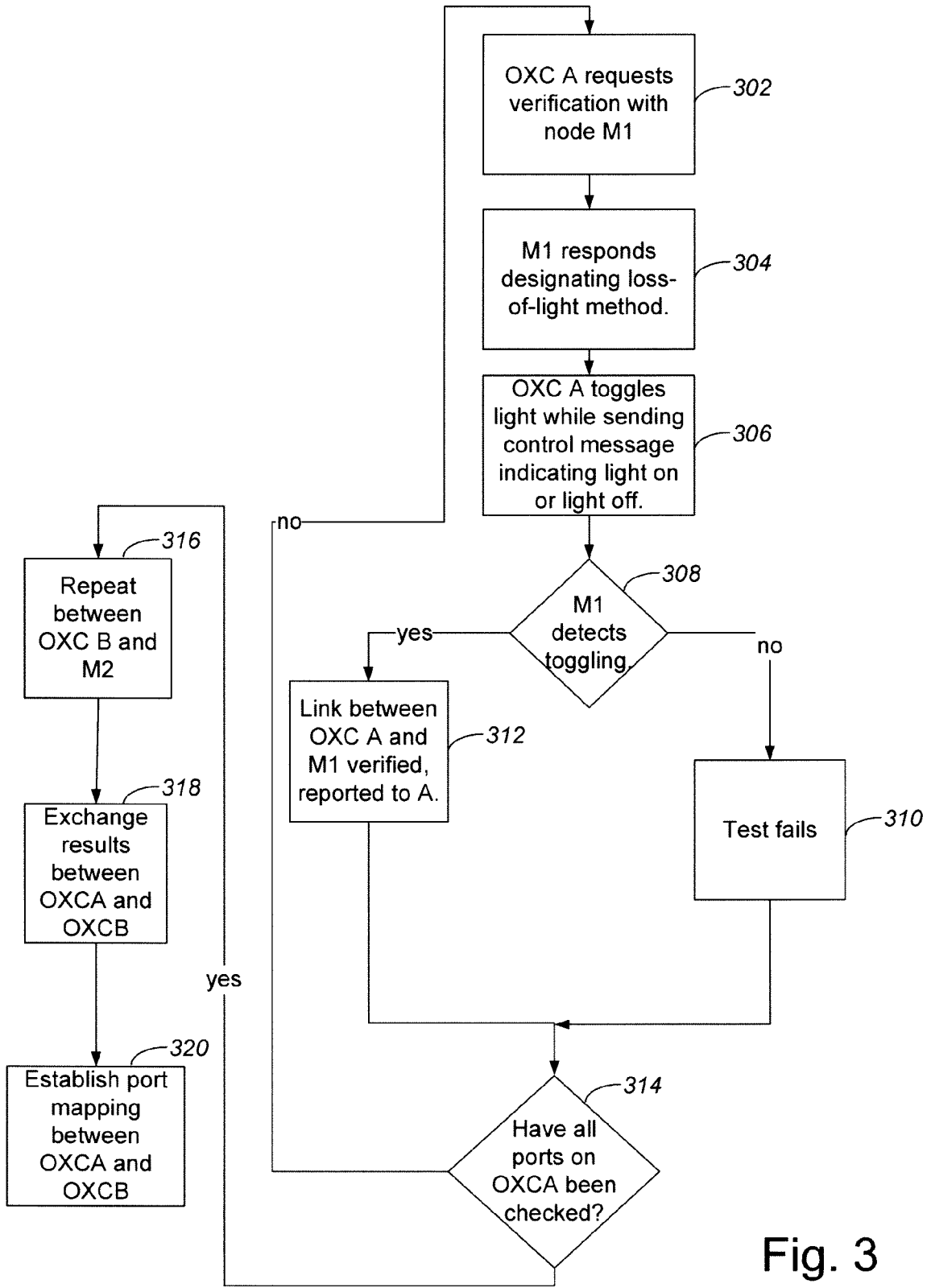


Fig. 3

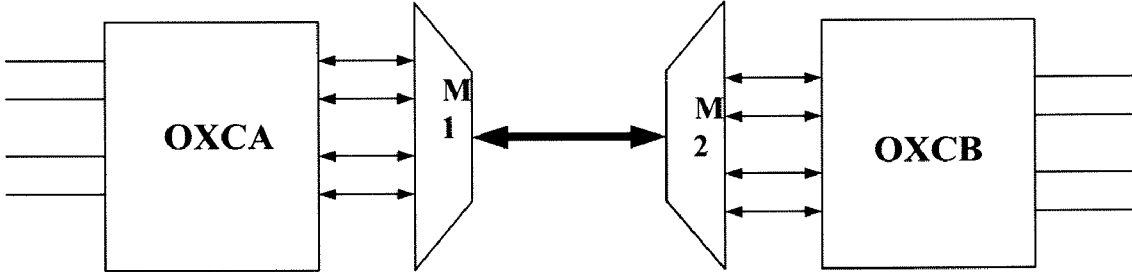


Fig. 4

## LINK DISCOVERY AND VERIFICATION USING LOSS OF LIGHT

### CROSS REFERENCE TO RELATED APPLICATION

**[0001]** This U.S. patent application is a continuation of co-pending U.S. patent application Ser. No. 09/968,096, filed Sep. 28, 2001, which is incorporated herein by reference in its entirety.

### BACKGROUND OF THE INVENTION

**[0002]** The present invention relates to data communication networks, and more particularly to systems and methods for discovering and verifying links between nodes.

**[0003]** Internet traffic is typically forwarded from a source to a destination through a series of nodes connected by links. The nodes represent network devices such as routers, switches, etc. The links are physical media such as optical fiber, twisted pair, etc. and often connect a pair of nodes. Various protocols that control the forwarding of packets through a network depend on each node having a correct understanding of the links available to its immediate neighbors.

**[0004]** Typically, discovery and/or verification of a link between two nodes is accomplished by an exchange of packets over the link. As will now be explained, however, this exchange of packets is problematic for an important new class of network devices.

**[0005]** To accommodate increasing volumes of network traffic, the Internet is relying more and more on the vast bandwidth of optical fiber media. Routing and switching operations, however, have largely remained in the electrical domain. The need to convert optical signals to electrical form and then perform switching and/or routing computations on the electrical signals has become a bottleneck for optical networks. In order to remove this bottleneck, all-optical cross-connects (OXC's) have emerged as an important building block for optical networks. In an OXC, optical inputs and outputs are coupled to one another through a switching matrix without intermediate conversion to an electrical signal. Since the optical signals pass through untouched, OXC's do not incorporate expensive hardware either for conversion to electrical form and reconversion to optical form, or for processing packets in accordance with a protocol. This results in enormous savings in cost and extremely high throughput. Another advantage of the OXC is that it requires no special adaptation to the protocol or data rate of the data carried by the switched optical signals.

**[0006]** To best integrate OXC's into the Internet, it is desirable to discover and verify links to and from them. However, if an OXC is to exchange packets with its neighbors for the purpose of link discovery and verification according to conventional techniques, it will have to incorporate line termination capability, i.e., the ability to generate and interpret packets in accordance with a protocol used by a neighbor. Supporting this capability will require that a line termination unit (LTU) capable of generating and interpreting packets according to the relevant protocol be incorporated within the OXC. To maximize the number of protocols which use an OXC, it will be desirable to incorporate a separate line termination unit (LTU) for each anticipated protocol that might be terminated by a neighboring node. Each LTU would incorporate the electrical to optical and the optical to electrical con-

version circuitry as well as high speed packet processing that would otherwise be made unnecessary by use of the OXC.

**[0007]** Furthermore, when a particular OXC switch port is having its neighbor connectivity verified, that port must be switched to another port connected to the correct LTU so that the appropriate packets may be exchanged. Thus, each LTU that is needed consumes a port and also a significant percentage of the available switching resources. It becomes clear then that current link discovery and verification techniques are expensive to apply to optical switching equipment that is not otherwise capable of originating and/or interpreting packets via optical links.

**[0008]** Another type of device that is finding increasing application on the Internet is the dense wave division multiplexer (DWDM) that multiplexes signals from several incoming fibers onto multiple wavelengths on the same fiber and demultiplexes multiple wavelength components carried by the same fiber onto separate fibers. By allowing the combining of numerous signals on the same fiber, DWDMs greatly expand the transmission capacity without increasing the data rate of individual optical signals, thus simplifying the electronics needed to generate and interpret the payload data streams modulated onto the optical signals. DWDMs are also preferably insensitive to protocol and data rate.

**[0009]** DWDMs, depending on the application, may or may not incorporate hardware to convert optical signals to electrical signals and vice versa. However, DWDM equipment does not typically include the capability to terminate packets that are being carried by optical signals. As with OXC's, this represents an obstacle to current link discovery and verification techniques.

**[0010]** What is needed are systems and methods for link discovery and verification that minimize the needed line termination resources and are thus suitable for implementation on devices that do not otherwise provide extensive line termination capabilities such as OXC's and DWDMs.

### BRIEF DESCRIPTION OF THE DRAWINGS

**[0011]** FIG. 1 depicts an optical cross-connect (OXC) suitable for implementing one embodiment of the present invention.

**[0012]** FIGS. 2A-2B depict a dense wave division multiplexer (DWDM) suitable for implementing one embodiment of the present invention.

**[0013]** FIG. 3 is a flowchart describing steps of link verification using loss of light according to one embodiment of the present invention.

**[0014]** FIG. 4 depicts a link verification situation where loss of light techniques may be used according to one embodiment of the present invention.

### DETAILED DESCRIPTION

**[0015]** The present invention finds application in networks where it is desirable for nodes to discover and verify links among themselves. Wherever the terms "verify" or "verification" are used, it will be understood that the present invention also applies to the discovery of links. One particular application of the present invention is to optical network devices such as optical cross-connects (OXC's) and dense wave division multiplexers (DWDMs) that require link verification yet are often unable to terminate packets.

**[0016]** Before describing details of link verification operation according to the present invention, the structures of rep-

representative nodes that may participate in link verification schemes will be described. FIG. 1 depicts an all-optical cross-connect (OXC) 100 suitable for implementing one embodiment of the present invention. An OXC 100 provides connectivity between a plurality of inputs 102 and a plurality of outputs 104. These are understood to be optical inputs and optical outputs. A photodetector 103 is coupled to each of inputs 102. Any one of inputs 102 may be selectively coupled to any one of outputs 104 through an all-optical switching fabric 106. Switching fabric 106 has eight ports each including one input and one output. The input to a port 8 is connected to a laser 108.

[0017] Switching fabric 106 is implemented by e.g., a MEMS integrated circuit that uses adjustable mirrors to guide light from a desired input to the desired output. A representative switching fabric is the PDS-002 model available from OMM, Inc. of San Diego, Calif. Any suitable technology may be used to implement switching fabric 106. A control processor 110 selects a mapping of inputs to outputs. Control processor 110 may be implemented in any suitable manner such as e.g., a general programmable processor, custom logic, multiprocessor system, or any combination thereof. Processor 110 may execute software instructions in any suitable machine level or high level programming language. Storage medium 112 may store instructions for control processor 110. Storage medium 112 may represent a memory device such as a random access memory device, magnetic storage medium, an optical storage medium, etc. Instructions on storage medium 112 may be loaded from another storage medium such as e.g., a compact disc (CD), a digital video disc (DVD), a floppy disc, etc. Another example of loading instructions from another storage medium is receiving code via a network.

[0018] Control packets used in implementing one embodiment of the present invention may be transmitted and received via a control interface 114. Control interface 114 may be e.g., an Ethernet interface, a SONET Interface, etc. Control interface 114 may also receive control packets to direct the mapping between inputs 102 and outputs 104.

[0019] Referring now to FIGS. 2A-2B, a dense wave division multiplexer (DWDM) 200 suitable for implementing the present invention is depicted. FIG. 2A depicts a portion of DWDM 200 where multiple optical signals are combined onto the same fiber. FIG. 2B depicts a portion of DWDM 200 that operates in the reverse direction, extracting multiple wavelength components from a single fiber and separating them onto multiple fibers each carrying a single wavelength. For ease of illustration, FIG. 2A-2B assume a system that combines 3 wavelengths on a single fiber but it will of course be appreciated that DWDM techniques apply to any number of wavelengths.

[0020] In FIG. 2A, each incoming optical signal is input to an optical-to-electrical conversion (o/e/o) block 202. Each o/e/o block 202 incorporates a photodetector to convert the optical signal to electrical form and a laser to regenerate the optical signal. The o/e/o block 202 may also position its output in the frequency domain in accordance with an operative wavelength division multiplexing scheme. To support link verification in accordance with the present invention, o/e/o block 202 is capable of sensing the presence or absence of light on its input. The various outputs of o/e/o blocks 202 are combined onto the same fiber by a multiplexer 204.

[0021] In FIG. 2B, a demultiplexer 206 receives a wavelength division multiplexed signal including multiple wavelength components and outputs the components on separate

fibers. A set of o/e/o blocks 208 convert the demultiplexed optical signals to electrical form and then regenerate them optically onto wavelengths appropriate for reception by other hardware. The o/e/o blocks 208 incorporate photodetectors to convert from optical to electrical and lasers to regenerate the optical signals.

[0022] A control processor 210 interacts with o/e/o blocks 202 and o/e/o blocks 208 to monitor and control activity. According to one embodiment of the present invention, control processor 210 may direct any of o/e/o blocks 208 to turn light output on or off to support link verification. Furthermore, control processor 210 is capable of monitoring whether light energy is present at the inputs of o/e/o blocks 202. Control processor 210 may be implemented, e.g., in any of the ways discussed in reference to control processor 110. Also, as with control processor 110, instructions for control processor 210 may be stored on a computer-readable storage medium 212 or loaded from another storage medium. Control interface 214 may, similarly to control interface 114, exchange control packets to support link verification, or also receive instructions concerning the overall operation of DWDM 200.

[0023] Now that the structures of representative participating nodes have been described, link verification according to one embodiment of the present invention will be described with reference to an example situation depicted in FIG. 4. In FIG. 4, OXCA is to verify links with OXCB. OXCA and OXCB are depicted as exchanging four optical signals that are wavelength multiplexed onto the same fiber by the operation of DWDM's M1 and M2. Thus, four ports (input and output) of OXCA are coupled to four ports of M1 (input and output) while four ports of OXCB (input and output) are coupled to four ports of M2 (input and output).

[0024] FIG. 3 is a flowchart describing steps of link verification between OXCA and OXCB according to one embodiment of the present invention. Details of link verification operation will be described in terms of an extension of the Link Management Protocol (LMP) which is described in Lang, et al. Link Management Protocol (LMP) Internet Draft, Internet Engineering Task Force, July 2001, the contents of which are herein incorporated in their entirety by reference for all purposes. The present invention may also be implemented without employing LMP.

[0025] Since OXCA is not directly coupled to OXCB but rather coupled to OXCB through M1 and M2, link verification between OXCA and OXCB is accomplished by first verifying between OXCA and M1 and between OXCB and M2 and then establishing a mapping based on the wavelength assignments in place between M1 and M2. At step 302, OXCA requests verification with M1. This request is sent through the control channel rather than through any of the optical connections between OXCA and M1 and identifies the IP addresses of OXCA and M1 as well as which port OXCA will be verifying connectivity through. M1 responds at step 304 by sending an LMP message, VerifyTransportMechanism, through the control channel, that is extended according to the present invention to designate loss-of-light as the verification mechanism. Instead of sending test packets via the port being verified as would be done in accordance with LMP, OXCA, at step 306, toggles the light output by the port under test. Simultaneously, OXCA sends OXCB control messages via the control channel indicating whether the light is currently on or off. The frequency of toggling should preferably be sufficiently low for the control messages for the synchronized control messages to be communicated and processed.



**[0026]** In order to perform the toggling function, switching fabric **106** may be used to alternately connect and disconnect the output of the port under test to port **8** which is hardwired to laser **108**. Alternatively, laser **108** may be switched on and off to achieve the desired toggling.

**[0027]** At step **308**, **M1** monitors its input ports to look for the toggling effect caused by **OXCA**. **M1** will monitor the light on or light off condition of each port to identify a particular port that it is indeed synchronized with the control messages generated by **OXCA**. Alternatively, it is also possible to communicate an aperiodic toggling pattern or a pseudorandomly generated toggling pattern to **M1** instead of communicating control messages synchronized with the toggling. **M1** would then seek to match any detected toggling to the pattern. If no toggling has been detected on any of **M1**'s ports, processing proceeds to a step **310** and the test fails. **M1** sends an LMP TestStatusFailure message indicating the failure to **OXCA** via the control channel.

**[0028]** If, however, **M1** has detected toggling through a particular one of its ports at step **308**, then a link between **OXCA** and **M1** has been verified. Then, at step **312**, **M1** sends **OXCA**, via the control channel, an LMP TestStatusSuccess message identifying the ports that have been verified as linked. The TestStatusSuccess message may also identify the wavelength to which **M1** assigns the verified port.

**[0029]** After either step **312** or step **310**, link verification proceeds to a step **314** which tests whether **OXCA** has verified connectivity to **M1** through all of its available ports already. If **OXCA** has not, then processing returns to step **302** for verification of the mapping of the next available port. If however, all of the available ports on **OXCA** have been mapped to **M1**, execution proceeds to a step **316**. Although step **316** is depicted as following the previously discussed steps, it will be appreciated that it may also occur in parallel.

**[0030]** At step **316**, **OXCB** verifies its connectivity to **M2**, establishing a similar mapping of its ports to the ports of **M2** using the same procedure that has been described with reference to **OXCA** and **M1**.

**[0031]** At the completion of step **316**, **OXCA** is aware of the mapping of its ports to the ports of **M1**, while **OXCB** is aware of the mapping of its ports to **M2**. This also includes an awareness of the particular wavelengths associated with each relevant port of **M1** such that each port of **OXCA** has thereby been mapped to a wavelength on the line connecting **M1** and **M2**. Similarly, each port on **OXCB** has also now been associated with a wavelength on the line coupling **M1** and **M2**. It is therefore now possible by analyzing this wavelength mapping, to establish the port mapping between **OXCA** and **OXCB**.

**[0032]** In one embodiment, LMP LinkSummary messages are used to establish the port mapping. For example, at step **318**, **OXCA** sends **OXCB** a series of LinkSummary messages via the control channel indicating the mapping between the ports of **OXCA** and the various wavelengths. Similarly, also at step **318**, **OXCB** sends **OXCA** its own mapping between the ports of **OXCB** and the various associated wavelengths. Based on this exchange of information, either **OXCA** or **OXCB** determines the mapping between ports of **OXCA** and ports of **OXCB** and communicates this mapping to the other **OXC** at step **320**, using a LinkSummary message.

**[0033]** It should be noted that although the above process has been described with reference to **OXCs** performing toggling and **DWDMs** detecting this toggling, the present invention also contemplates that toggling may be detected by the

**OXCs** through, e.g., one of photodetectors **103** coupled to inputs **102** in FIG. 1. Also, the present invention is not restricted to optical devices and toggling light but will find application, e.g., wherever a signal may be turned on or off.

**[0034]** An advantageous modification to the link verification technique described above is for a single node to simultaneously toggle light through multiple ports. This greatly increases the speed of link verification. Toggling will then preferably be coordinated so that the monitoring node can distinguish among the multiple toggled light signals. For example, instead of sending control messages indicating the "light on" or "light off" state of each port, a single control message may include a bitmap that gives the current "light on" or "light off" state of each toggling port. By performing verification through multiple ports in parallel and aggregating current light state information into a single control message, one can greatly reduce the message exchanges needed to map ports.

**[0035]** Consider a situation where the toggling node has 8 ports. The sender first disables light on ports **0-3**, enable light on ports **4-7** and send the value **0xF0** (hex) via the control channel as a bitmap where "0" indicates the absence of light and "1" the presence of light. Once the light status had been recorded on the monitoring node, the toggling node changes to the next pattern, for example, disabling light on ports **0,1,4**, and **5** and enabling light on ports **2,3,6** and **7** resulting in a bitmap of **0xCC** (hex). Each change in pattern is signaled by a control message including the currently operative bitmap. By activating and deactivating light at the various ports in accordance with the bitmaps in the series, **0xF0**, **0xCC**, **0xAA**, and **0x55**, the monitoring node maps all eight ports with a sequence of only 4 patterns. Using this technique, 16 ports can be distinguished using a sequence of only 5 patterns, 32 ports using 6 patterns, etc.

**[0036]** With the above-described modification, the time necessary to verify links from a node having  $p$  ports will vary roughly with  $\ln(p)$ . By contrast, with the technique specified by LMP, link verification time varies roughly with  $p^2$ . The present invention thus provides enormous advantages in applications where nodes have a large number of ports, e.g., 512 ports.

**[0037]** In one embodiment, where line termination capability is available, loss of light is used as an extension to LMP verification techniques that rely on exchanging data packets via the tested ports rather than through a control channel as described above. Considering the procedure described above for link verification between **OXCA** and **M1**, one could use the loss of light technique that has been described to eliminate the ports on **M1** that do not detect toggling and then use the verification techniques provided by LMP to further verify the link to the port that does detect toggling. This can greatly increase the speed of verification, especially where there are large numbers of ports to be tested.

**[0038]** The present invention thus provides rapid link verification without the need to terminate packets in accordance with any particular protocol. Alternatively, existing link verification techniques may be accelerated.

**[0039]** It is understood that the examples and embodiments that are described herein are for illustrative purposes only and various modifications are changes in light there of will be suggested to persons skilled in the art and are to be included within the spirit and purview of this application and scope of the appended claims and their full scope of equivalents. For example, although optical cross-connects and **DWDMs** have

been described as representative nodes that may implement the link verification techniques described herein, the present invention may be applied to any suitable network device including, e.g., IP switches, ATM switches, SONET TDM add/drop multiplexers, optical packets switches, optical packet routers, etc. Also, the term “port” may refer to a logical port as well as a physical port.

1. A method for operating a first node in a data communication network to verify connectivity to a second node in said data communication network, said method comprising:

sending a request for verification of connectivity to said second node, said request for verification of connectivity identifying an IP address of said first node and an IP address of said second node, said request for verification of connectivity further identifying a port of said first node;

toggling a signal emitted by said port of said first node; notifying said second node of a toggling mode of said port to facilitate connectivity verification; and

receiving a first message from said second node indicating whether said second node detected said toggling, said request for verification of connectivity identifying said first node being sent on a control channel via a control message separate from said signal and said first message being sent on said control channel.

2. The method of claim 1 wherein if said first message indicates that said second node detected said toggling, said first message identifies a property which the second node assigns said port of said first node.

3. The method of claim 1 further comprising: simultaneously with toggling of said signal emitted by said port of said first node, toggling another signal emitted by another port of said first node.

4. The method of claim 1 further comprising: sending notifications from said first node to said second node as said signal toggles on and off to facilitate connectivity verification.

5. The method of claim 1 wherein said signal comprises a selected wavelength component of a WDM signal.

6. The method of claim 1 wherein toggling comprises: turning said signal on and off aperiodically.

7. The method of claim 1 wherein toggling comprises: turning said signal on and off in accordance with a pseudorandomly selected sequence of intervals.

8. A method for operating a first node in a data communication network to verify connectivity to a second node in said data communication network, said method comprising:

receiving a request for verification of connectivity from said second node on a control channel, said request for verification of connectivity identifying an IP address of said second node and an IP address of said first node, said request for verification of connectivity further identifying a port at said second node;

detecting toggling of a signal received at a port of said first node;

verifying a link to said second node by matching said toggling to a known toggling mode of said port at said second node; and

sending a message to said second node indicating whether said first node matched said toggling to said known toggling mode, said message being sent on said control channel.

9. The method of claim 8 wherein if said message indicates that said first node matched said toggling to said known

toggling mode, said message identifies a property which said first node assigns said port at said second node

10. The method of claim 8 further comprising:

simultaneously with detecting toggling of said signal received at said port of said first node, detecting toggling of another signal received at another port of said first node; and

verifying an additional link to said second node by matching said toggling of another signal to a known toggling mode of an additional port at said second node.

11. The method of claim 8 wherein verifying comprises: receiving messages from said second node indicating when a toggled signal transmitted by said second node has been turned on and turned off; and

matching said messages to on and off states of said signal received at said port of said first node.

12. The method of claim 10 wherein receiving messages comprises:

receiving messages through a control channel.

13. An apparatus for operating a first node in a data communication network to verify connectivity to a second node in said data communication network, said apparatus comprising:

a processor that sends a request for verification of connectivity to said second node, said processor being arranged to toggle a signal emitted by a port of said first node, said request for verification of connectivity being arranged to identify an IP address of said first node, an IP address of said second node, and said port; and

a control interface that notifies said second node of a toggling mode of said port to facilitate connectivity verification, wherein said request for verification is sent on a control channel via a control message separate from said signal.

14. The apparatus of claim 13 wherein said control interface sends notifications from said first node to said second node as said signal toggles on and off to facilitate connectivity verification.

15. The apparatus of claim 13 wherein said signal comprises a selected wavelength component of a WDM signal.

16. The apparatus of claim 13 wherein said processor toggles by turning said signal on and off aperiodically.

17. The apparatus of claim 13 wherein said processor toggles by turning said signal on and off in accordance with a pseudorandomly selected sequence of intervals.

18. An apparatus for operating a first node in a data communication network to verify connectivity to a second node in said data communication network, said apparatus comprising:

a photodetector that detects toggling of a signal received at a port of said first node; and

a processor that receives a request for verification of connectivity at said second node and verifies a link to said second node by matching said toggling to a known toggling mode of a port at said second node, wherein said request for verification comprises identification of an IP address of said first node, an IP address of said second node, and a port over which connectivity verification is to be performed and wherein said identification of said IP address of said first node, said IP address of said second node, and said port is received on a control channel via a control message separate from said signal.

19. The apparatus of claim 18 wherein said processor receives messages from said second node indicating when a

toggled signal transmitted by said second node has been turned on and turned off and matches said messages to on and off states of said signal received at said port of said first node.

20. The apparatus of claim 18 wherein said messages are received through a control channel.

21. An apparatus for operating a first node in a data communication network to verify connectivity to a second node in said data communication network, said apparatus comprising:

means for sending a request or verification of connectivity to said second node, said request for verification of connectivity identifying an IP address of said first node and an IP address of said second node, said request for verification of connectivity further identifying a port of said first node;

means for toggling a signal emitted by said port of said first node;

means for notifying said second node of a toggling mode of said port to facilitate connectivity verification; and

means for receiving a first message from said second node indicating whether said second node detected said toggling, said request for verification of connectivity identifying said first node being sent on a control channel via a control message separate from said signal and said first message being sent on said control channel.

22. The apparatus of claim 21 wherein if said first message indicates that said second node detected said toggling, said first message identifies a property which the second node assigns said port of said first node.

23. The apparatus of claim 22 further comprising:

means for sending notifications from said first node to said second node as said signal toggles on and off to facilitate connectivity verification.

24. An apparatus for operating a first node in a data communication network to verify connectivity to a second node in said data communication network, said apparatus comprising:

means for receiving a request for verification of connectivity from said second node on a control channel, said request for verification of connectivity identifying an IP address of said second node and an IP address of said first node;

means for detecting toggling of an optical signal received at a port of said first node; and

means for verifying a link to said second node by matching said toggling to a known toggling mode of said port at said second node, wherein said request for verification comprises identification of an IP address of said first

node, an IP address of said second node, and a port over which connectivity verification is to be performed and wherein said identification of said IP address of said first node, said IP address of said second node, and port information is received on a control channel via a control message separate from said signal.

25. A computer program product for operating a first node in a data communication network to verify connectivity to a second node in said data communication network, said computer program product comprising:

code that receives a request for verification of connectivity from said second node, said request for verification of connectivity identifying an IP address of said first node and an IP address of said second node;

code that detects toggling of a signal received at a port of said first node;

code that verifies a link to said second node by matching said toggling to a known toggling mode of a port at said second node, wherein said request for verification of connectivity identifying said IP address of said first node, said IP address of said second node, and said port of said second node is sent on a control channel via a control message separate from said signal; and

a computer-readable storage medium that stores the codes.

26. A computer program product for operating a first node in a data communication network to verify connectivity to a second node in said data communication network, said computer program product comprising:

code that is configured for sending a request for verification of connectivity to said second node on a control channel, said request for verification of connectivity identifying an IP address of said second node and an IP address of said first node;

code that toggles a signal emitted by a port of said first node;

code that notifies said second node of a toggling mode of said port to facilitate connectivity verification, wherein said request for verification comprises identification of said IP address of said first node, said IP address of said second node, and said port at said second node over which connectivity verification is to be performed and wherein said identification of said IP address of said first node, said IP address of said second node, and said port at said second node is received on a control channel via a control message separate from said signal; and

a computer-readable storage medium for storing the codes.

\* \* \* \* \*