US009270482B2

(12) **United States Patent**
Balbierer et al.

(10) **Patent No.: US 9,270,482 B2**
(45) **Date of Patent: Feb. 23, 2016**

(54) **METHOD FOR ACTIVATING A NETWORK COMPONENT OF A VEHICLE NETWORK SYSTEM**

(75) Inventors: **Norbert Balbierer**, Kelheim (DE); **Josef Nöbauer**, Neukirchen-Balbini (DE)

(73) Assignee: **Continental Automotive GmbH**, Hannover (DE)

(*) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 717 days.

(21) Appl. No.: **13/580,659**

(22) PCT Filed: **Feb. 18, 2011**

(86) PCT No.: **PCT/EP2011/052456**

§ 371 (c)(1), (2), (4) Date: **Aug. 22, 2012**

(87) PCT Pub. No.: **WO2011/101446**

PCT Pub. Date: **Aug. 25, 2011**

(65) **Prior Publication Data**

US 2012/0320792 A1 Dec. 20, 2012

(30) **Foreign Application Priority Data**

Feb. 22, 2010 (DE) 10 2010 008 818

(51) **Int. Cl.**
H04L 12/40 (2006.01)
H04L 12/12 (2006.01)
H04L 12/46 (2006.01)

(52) **U.S. Cl.**
CPC ***H04L 12/40039*** (2013.01); ***H04L 12/12*** (2013.01); ***H04L 12/462*** (2013.01); *H04L 12/40045* (2013.01); *H04L 2012/40273* (2013.01); *Y02B 60/32* (2013.01); *Y02B 60/34* (2013.01); *Y02B 60/35* (2013.01)

(58) **Field of Classification Search**
CPC H04L 12/4625; H04L 12/40006; H04L 43/0817; H04L 12/12; H04L 12/462; H04L 12/40045; H04L 12/40039; H04L 2012/40215; H04L 2012/40267; H04L 2012/40273; Y02B 60/34; Y02B 60/35; B60R 16/0315
See application file for complete search history.

(56) **References Cited**

U.S. PATENT DOCUMENTS

7,573,891 B1 * 8/2009 Chow et al. 370/401
7,649,912 B2 * 1/2010 Balasubramanian et al. 370/509
7,778,739 B2 * 8/2010 Preston et al. 701/1
(Continued)

FOREIGN PATENT DOCUMENTS

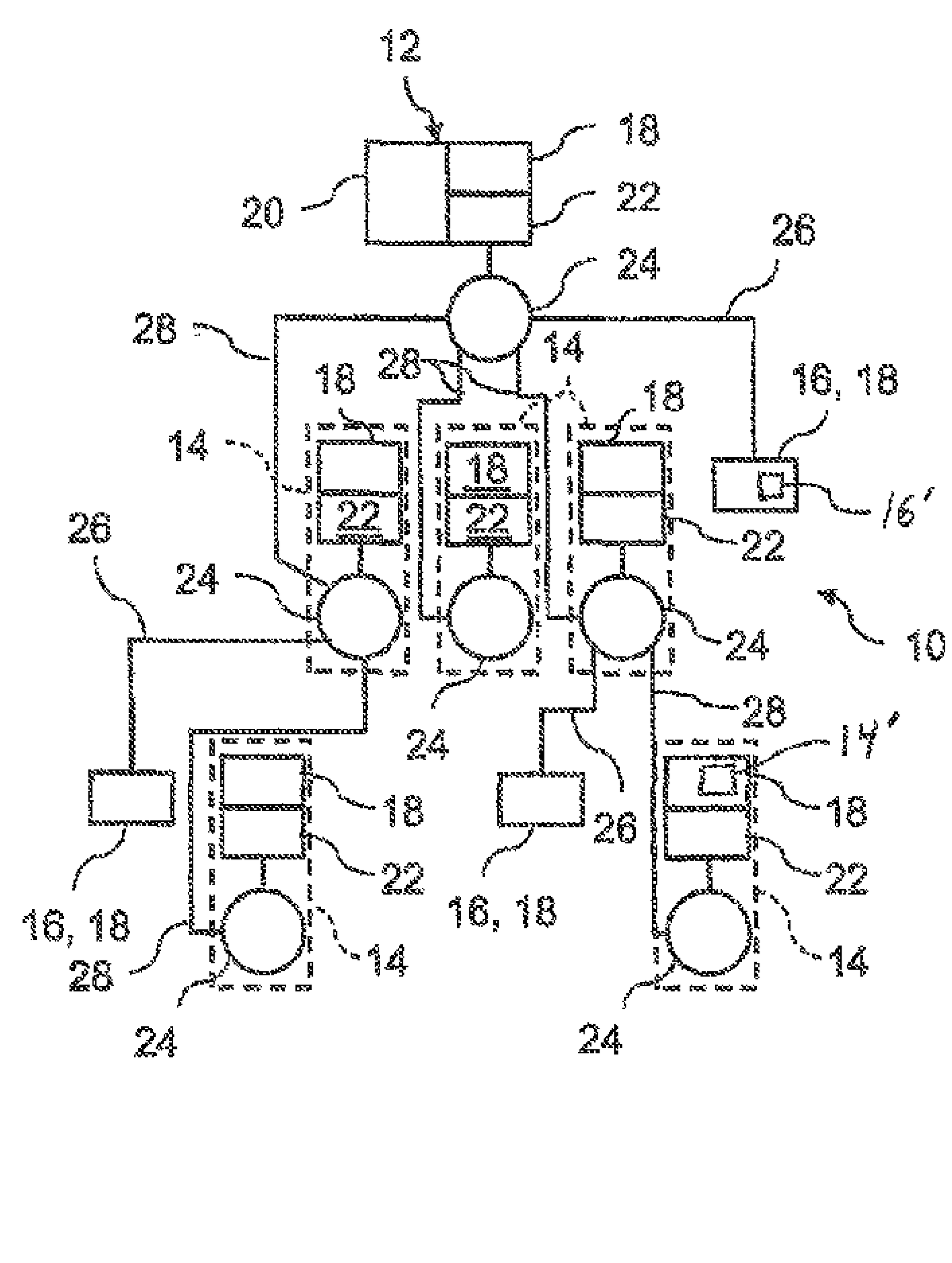
DE 100 31 891 A1 1/2002
DE 20 2005 021 265 U1 7/2007
(Continued)

Primary Examiner — Brian O'Connor
(74) *Attorney, Agent, or Firm* — Cozen O'Connor

(57) **ABSTRACT**

A method and system for activating at least one temporarily inactive network component of a network system for a vehicle, in particular a motor vehicle. A central network device of the network system is connected via signals to the network components by a path inside the network system, the path extending at least partially across a network segment of the network system. The network segment connects via signals the network component and a first activation device associated therewith to a switch device arranged in the path and to a second activation device associated therewith in an unbranched manner. The central network device responds to the first activation device by the switch device by sending a network function control signal.

10 Claims, 4 Drawing Sheets



(56) **References Cited**

U.S. PATENT DOCUMENTS

8,310,352	B2 *	11/2012	Miyashita	340/425.2
8,467,324	B2 *	6/2013	Yousefi et al.	370/310.2
2002/0006139	A1	1/2002	Kikkawa et al.	
2003/0046327	A1 *	3/2003	Reinold et al.	709/200
2003/0120396	A1	6/2003	Vollmer et al.	
2005/0256616	A1 *	11/2005	Rhoads	701/1
2006/0245454	A1 *	11/2006	Balasubramanian et al.	370/509
2007/0104124	A1	5/2007	Lee et al.	
2010/0098095	A1 *	4/2010	Kato et al.	370/401
2010/0188972	A1 *	7/2010	Knapp	370/226
2010/0215043	A1	8/2010	Hisada	
2010/0217965	A1	8/2010	Wolff	
2011/0035624	A1 *	2/2011	Miller	714/15
2011/0046844	A1	2/2011	Honner et al.	
2011/0145433	A1 *	6/2011	Noel et al.	709/232
2012/0320793	A1 *	12/2012	Balbierer et al.	370/254
2013/0151082	A1 *	6/2013	Preston	701/45

FOREIGN PATENT DOCUMENTS

EP	1 783 951	A2	5/2007
WO	WO 2009/044915	A1	4/2009
WO	WO 2009/054769	A1	4/2009
WO	WO 2010/097128	A1	9/2010

* cited by examiner

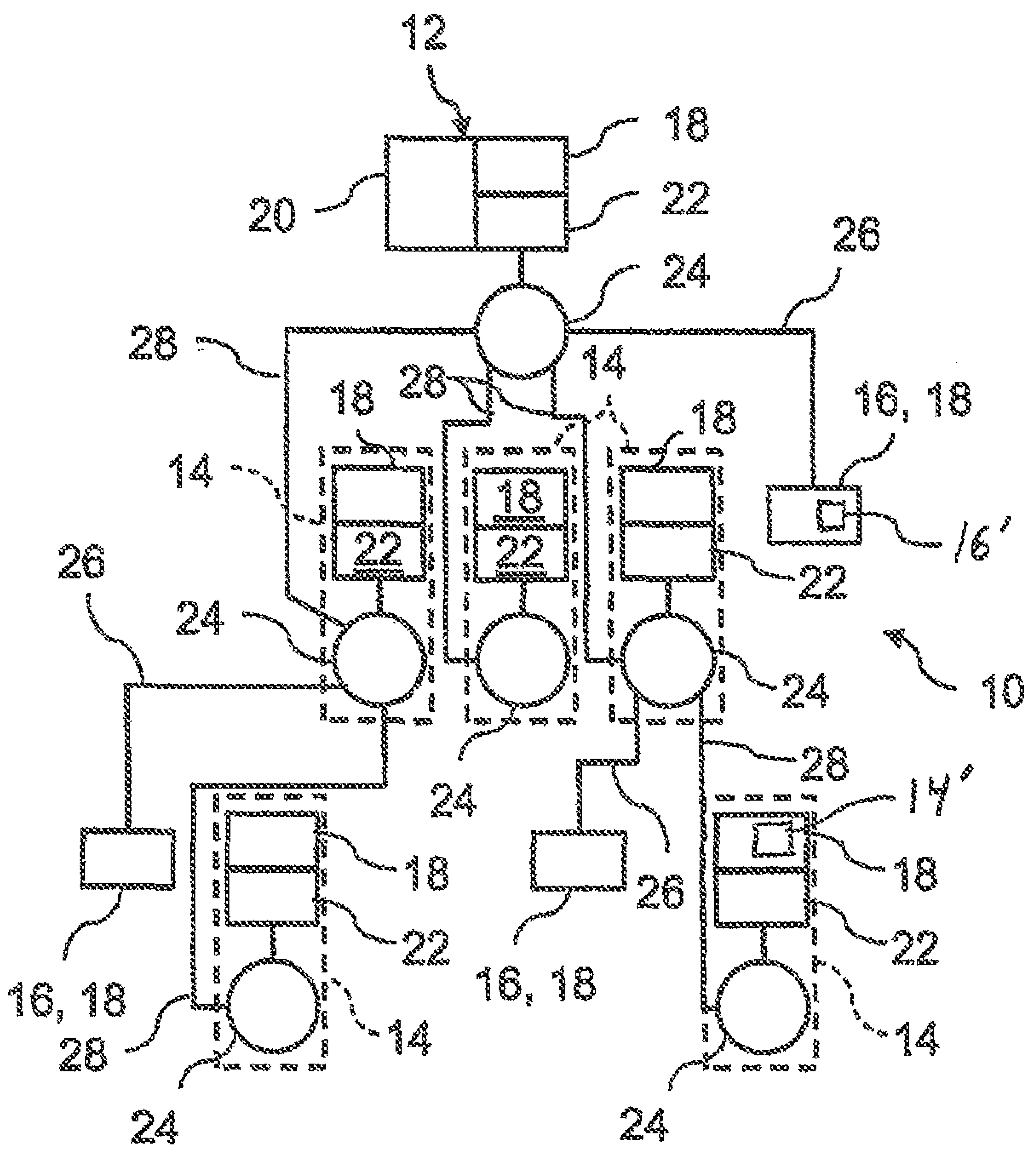

12
18
20
22
26
24
28
14
28
18
14
18
16, 18
18
22
16'
22
26
24
22
24
10
24
28
14'
18
26
18
16, 18
22
22
16, 18
28
14
14
24
24

Fig. 1

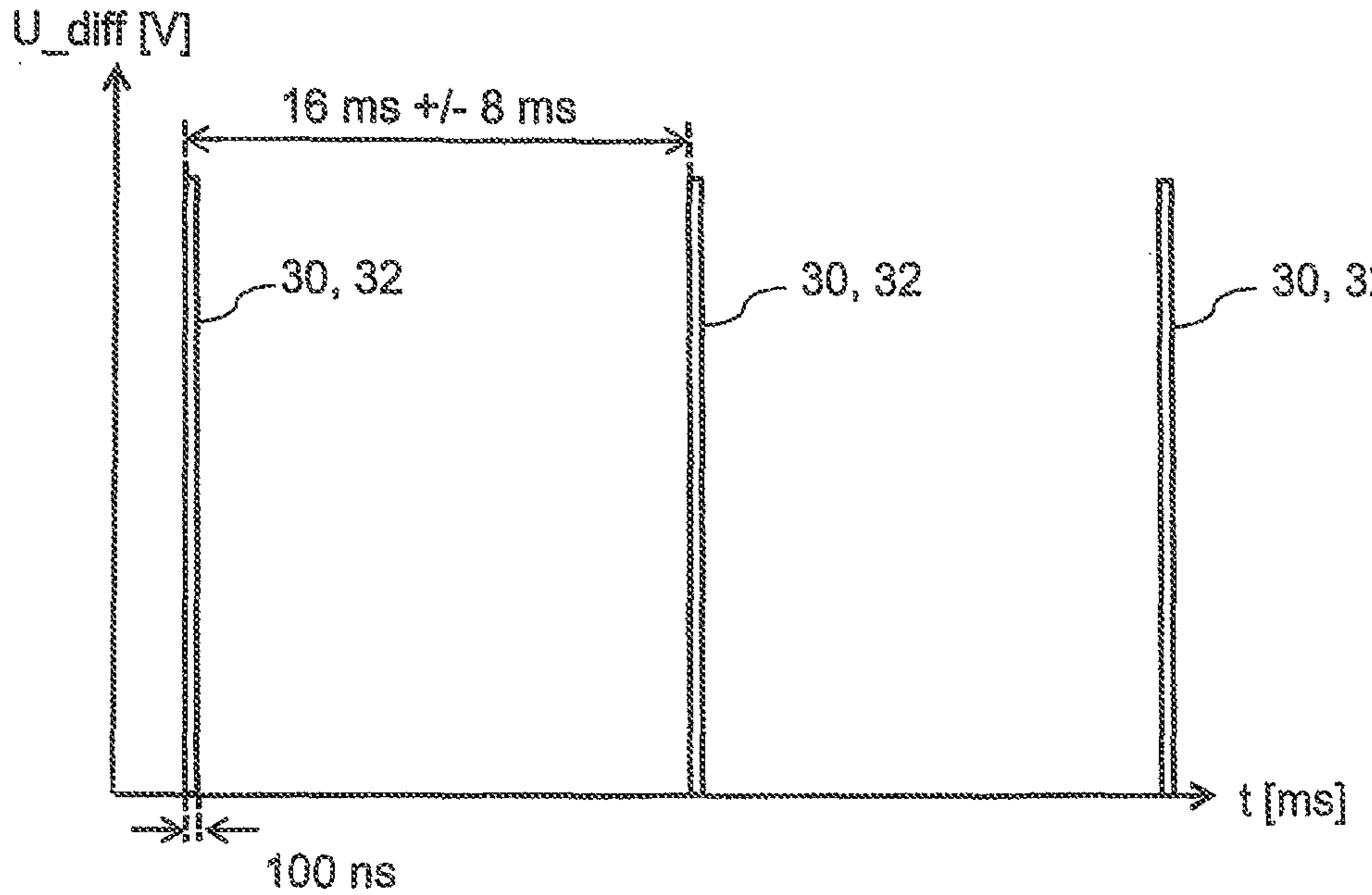

U_diff [V]
16 ms +/- 8 ms
30, 32
30, 32
30, 32
t [ms]
100 ns

Fig. 2

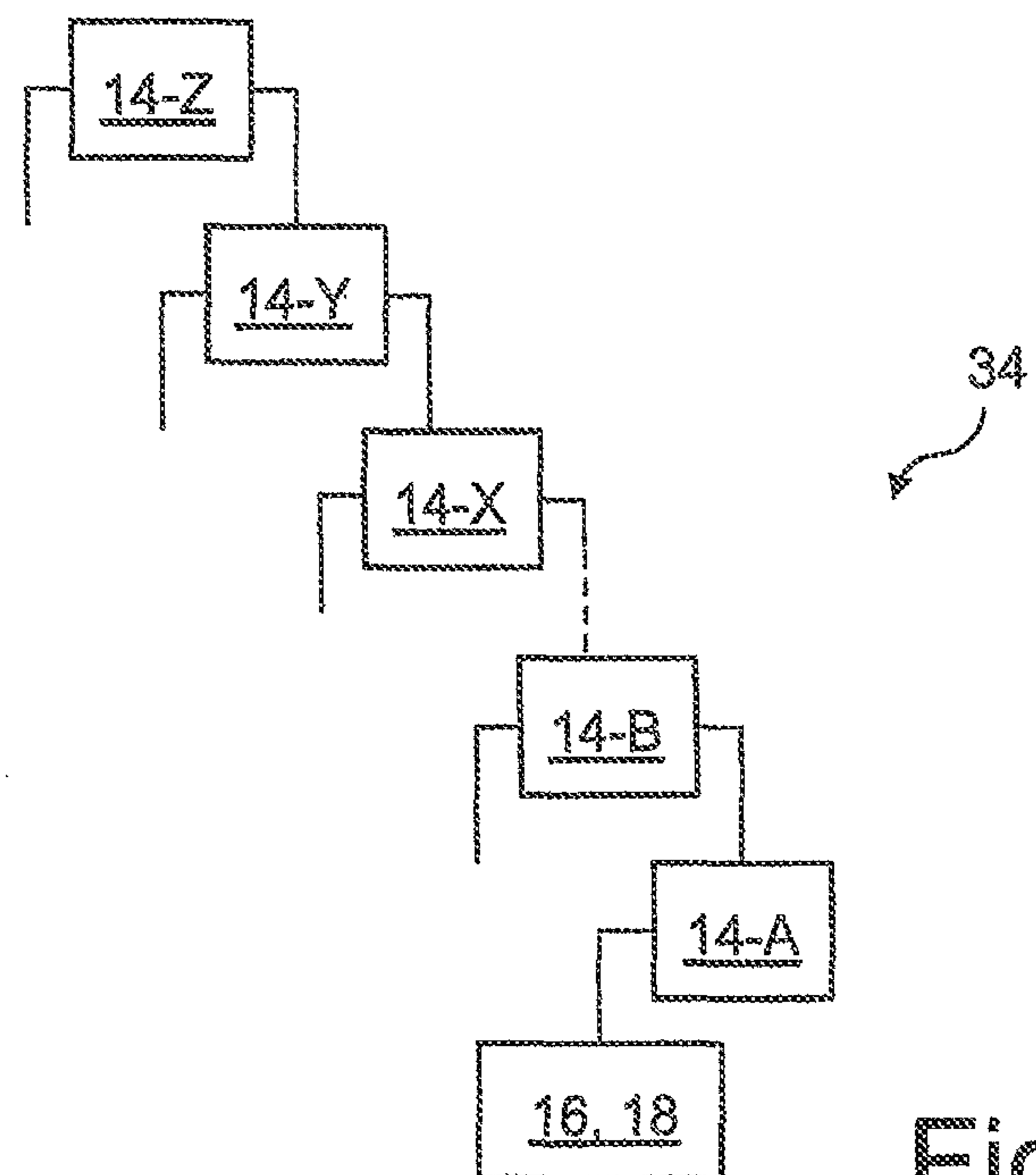

14-Z
14-Y
14-X
14-B
14-A
16, 18
34

Fig. 3

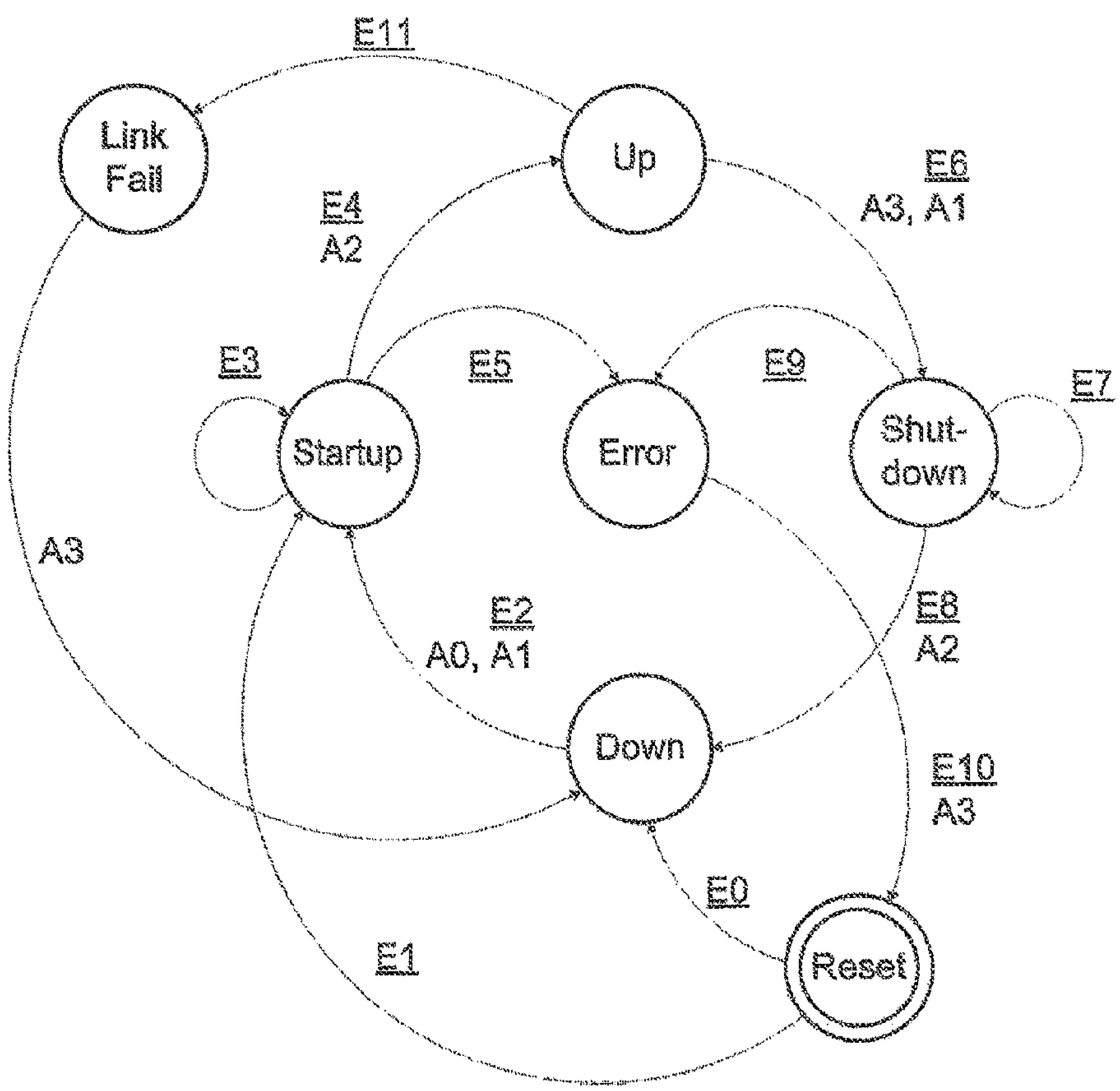

E11
Link Fail
Up
E6
A3, A1
E4
A2
E3
Startup
E5
Error
E9
Shut-down
E7
A3
E2
A0, A1
E8
A2
Down
E10
A3
E0
Reset
E1

Fig. 4

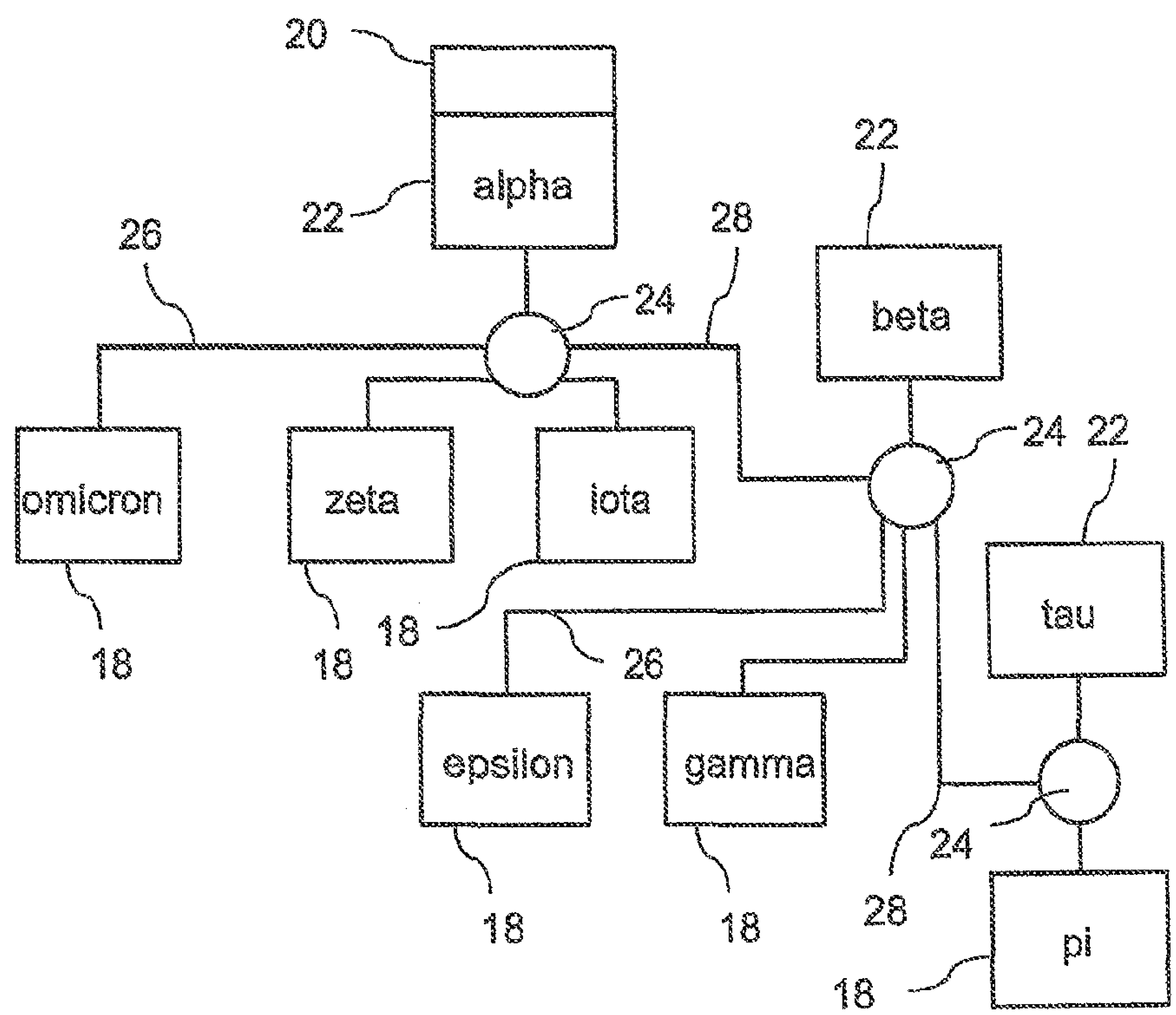

20
22
alpha
26
28
22
beta
24
24
22
omicron
zeta
iota
tau
18
18
18
26
epsilon
gamma
24
28
pi
18
18
18

Fig. 5

METHOD FOR ACTIVATING A NETWORK COMPONENT OF A VEHICLE NETWORK SYSTEM

CROSS REFERENCE TO RELATED APPLICATIONS

This is a U.S. national stage of application No. PCT/EP2011/052456, filed on 18 Feb. 2011. Priority is claimed on German, Application No.: 10 2010 008 818.8, filed 22 Feb. 2010, the content of which is incorporated here by reference.

BACKGROUND OF THE INVENTION

1. Field of the Invention

The invention relates to a method for activating at least one temporarily inactive network component of a network system for a vehicle, especially for a motor vehicle.

2. Prior Art

Known network systems for vehicles, particularly motor vehicles, are based in most cases on serial bus systems. An example of such a bus system for networking various control devices for implementing system-wide functions of a vehicle is an asynchronous, serial bus system based on a CAN bus (Controller Area Network). Another example is a LIN: Local Interconnect Network Bus System. Since electrical power is supplied in many vehicles by an energy store of limited storage capacity, it is desirable that non-active parts of the network do not absorb any power or absorb as little power as possible.

In the bus systems used in the field of application for vehicles, an energy detection concept is used. The entire bus system is initially inactive. An energy pulse on the bus line of the bus system that leads to the controller “waking up” and activating the entire system as a consequence. In this context, the energy pulse can be a data frame or also a single voltage pulse. In this system, the demand for quiescent current is extremely low but all components connected to the bus system are activated and “wake up”.

For stationary networks, the “Wake on LAN” standard (WOL) has been established for some time (LAN: Local Area Network). It enables inactive hosts in the network to be selectively woken by the so-called magic packet, an Ethernet frame that contains the MAC address of the host to be woken and which is recognized by the corresponding host Ethernet controller.

However, this technology is unsuitable for use in the automotive or motor vehicle field since the network controllers themselves must be active or at least partially active to recognize such a packet. As a result, the demand for quiescent current is distinctly too great for an automotive environment.

SUMMARY OF THE INVENTION

It is an object of one embodiment of the present invention is to provide a method for activating at least one temporarily inactive network component of a network for a vehicle in which individual network components can be activated selectively.

In the method according to one embodiment of the invention, a central network device of the network system is connected via signals to the network component via a path inside the network system. The path leads at least partially across a network segment of the network system, the network segment connecting via signals the network component and an associated first activation device unbranched to a switch device arranged in the path and to an associated second activation device, and the central network device addressing the activation device by the switch device by sending a network function control signal. The central network device of the network system has, in particular, a network manager module.

The network component is an electrical device, particularly a control device of a vehicle component of the vehicle, preferably of the motor vehicle. To minimize the energy demand of the vehicle, the electrical device is temporarily inactivated when it is not needed.

In a network system of particularly simple structure, the central network device of the network system is connected via signals to the corresponding network component via, in each case, one path within the network system, the respective path leading completely via a corresponding network segment. In this network system, the central network device has the switch device itself or is connected via signals to it by another network segment.

Whilst the network segment is unambiguously allocated to the respective network component, the other network segment can be allocated to a plurality of network components.

The network function control signal is preferably at least one voltage pulse that is applied by one component (network device, switch device and/or network component) to the signal line of the corresponding network segment.

According to a preferred embodiment of the invention, the first activation device activates the network component after receipt of the network function control signal and subsequently in turn sends out a further network function control signal to the second activation device for confirming the activation.

According to a further preferred embodiment of the invention, it is provided that the second activation device brings the switch device into a transmitting/receiving state after receipt of the further network function control signal. With this step, the activation is completely finished and the network component can communicate bidirectionally with the switch device via the associated network segment.

According to a preferred embodiment of the invention, the network is an Ethernet network. In an Ethernet network, the network components and a central network device (e.g. as hosts), switch devices, and a corresponding network structure with network segments allocated to respective hosts are already known. The network function control signal is designed, for example, as NLP (NLP: Normal Link Pulses).

According to a preferred embodiment of the invention, the network system has a tree topology formed by the central network device, the at least one switch device, and the network components. This topology is particularly suitable for implementing the method according to one embodiment of the invention. As an alternative, the network system preferably has a mesh topology.

In particular, the network component and/or the network device is a control device of a vehicle component or at least part of such a control device.

The invention also relates to a network system of a vehicle, especially a motor vehicle, preferably for carrying out the aforementioned method and suitable for activating at least one temporarily inactive network component. The network system according to one embodiment of the invention has a central network device connected via signals to the network component via a path within the network system, the path leading at least partially across a network segment of the network system and the network segment connecting via signals the network component and an associated first activation device unbranched to a switch device arranged in the path and to an associated second activation device, and the central network device addressing the first activation device by the switch device by sending a network function control signal.

The network component is an electrical device, particularly a control device, of a vehicle component of the vehicle, preferably of the motor vehicle. To minimize the power requirement of the vehicle, the electrical device is temporarily inactivated when it is not needed.

According to a preferred embodiment of the invention, the network is an Ethernet network. In an Ethernet network, network components and a central network device (e.g. as hosts), switch devices, and a corresponding network structure with network segments, which are allocated to respective hosts, are already known.

According to a preferred embodiment of the invention, it is provided that the network system has a tree topology formed by the central network device, the at least one switch device, and the network components. This topology is particularly suitable for implementing the method according to the invention. As an alternative, the network system preferably has a mesh topology.

In particular, the network component and/or the network device is a control device of a vehicle component or at least part of such a control device.

Finally, the invention also relates to a motor vehicle comprising an aforementioned network system, particularly an Ethernet network.

BRIEF DESCRIPTION OF THE DRAWINGS

In the text which follows, one embodiment of the invention will be explained in an exemplary manner referring to the drawing. However, the invention is not restricted to the exemplary embodiment shown. In the drawing:

FIG. **1** is a block diagram of a network system according to one embodiment of the invention;

FIG. **2** is a network function control signal designed as so-called "normal link pulses";

FIG. **3** is a block diagram of a network system according to one embodiment of the invention;

FIG. **4** is a state diagram of a so-called "port state machine" of a network system; and

FIG. **5** is a block diagram of a network system according to the invention in accordance with a further embodiment.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

FIG. **1** shows a diagrammatic structure of a network system **10** according to one embodiment of the invention with a tree topology formed by a central network device **12**, a plurality of switch devices **14** and a plurality of network components **16**. In this arrangement, the network system **10** is designed as Ethernet network. The central network device **12** comprises a host **18**, a network manager **20** (network manager module), a switch manager **22** (switch manager module) and a switch **24**. The central network device **12** is connected via signals via a network segment **26** to a network component **18** and via three other network segments **28** to three switch devices **14** with switch **24**, switch manager (switch manager module), **22** and host **18**. The switch devices **14** are in turn connected via signals via network segments **26** to network components **16** and via other network segments **28** to other switch components **14**, producing the tree structure.

To each of the network components **16**, a first activation device **16'** designed as a so-called "energy detection module" is allocated and to each of the switch devices **14**, a second activation device **14'** designed as an "energy detection module" is allocated. The power consumption of the first activation device is less than the power consumption of the associated active network components **16**, the power consumption of the second activation device is less than the power consumption of the associated active switch device **14** and of the associated central network device, respectively.

FIG. **2** shows the sequence in time of network function control signals **30**, designed as rectangular pulses (more precisely so-called NLPs—Normal Link Pulses) for testing the state of a connection (of a link) in an Ethernet network. For this purpose, a voltage U_diff is plotted against time t in a graph. The network function control signals **30**, that is to say the pulses **32** (NLPs), have a pulse width of 100 ns and a pulse spacing of 16 ms+/−8 ms.

According to one embodiment of the invention, these network function control signals **30** are used for activating a temporarily inactive network component **16** and for confirming the activation.

The following function results within a network system **10** in which a central network device **12** is connected via signals to the temporarily inactive network component **16** via a path inside the network system **10**. The path leads at least partially across a network segment **26** of the network system **10** and the network segment **26** connects, via signals, the network component **16** and an associated first activation device unbranched to a switch device **14** arranged in the path and to an associated second activation device. The following steps are provided:

- the central network device **12** addresses the first activation device of the network component **16** by the switch device **14** by sending a network function control signal **30**,
- the first activation device activates the network component **16** after receipt of the network function control signal **30** and subsequently in turn sends out a further network function control signal **30** to the second activation device of the switch device **14** for confirming the activation, whereupon
- the second activation device brings the switch device **14** into a transmitting/receiving state after receipt of the further network function control signal **30**.

In the text which follows, the resultant concept will be described again using a different terminology.

A Switched Ethernet consists physically of point-to-point connections. Thus, an energy detect principle can be applied individually for individual hosts **18**. It would not wake up the entire network since only one host **18**, especially a network component **16**, and one switch port (not shown in detail) of the switch device **14** are connected physically to a line (the network segment **26**).

By selectively controlling the activity on a link, a host **18**, especially a network component **16**, can accordingly be activated or deactivated selectively. For this purpose, the switch **24** has to switch the respective port on or off. When the port is switched on, link pulses (NLPs) **32** are applied to the line (the network segment **26**).

The link pulses **32** can be detected by an activation device designed as energy detect module and their presence can be indicated by an electrical output. The state of this output can trigger a wake-up (an activation) or a shut-down (an inactivation).

The ports are configured by the switch manager (the switch manager module) **22**. This is software responsible for the entire configuration, monitoring, and control of a switch **24**. Each port of a switch **24** is modeled and treated by the switch manage **22** as a finite state machine.

The network manager (the network manager module) **20** forms the centerpiece of the concept. It is the central management node that has the job of configuring and monitoring the entire network (network system **10**). This software has the overview of the entire topology of the network **10** and of the state of the individual hosts **18**. It can communicate with the individual switch managers **22** and thus allow individual hosts **18** to be activated and deactivated selectively.

The overall concept represents the interactions between the three individual modules (central network device **12** with network manager **20**, network component **16** with first activation device and switch device **14** with switch manager **22** and second activation device).

The first question arising with respect to the structure of the network management is whether this is a centralized or distributed management architecture. Although distributed management offers greater reliability, it is significantly more complex and difficult to handle. In addition, the necessary communication between the distributed management nodes causes additional data load. As a rule, a simple, uncomplicated management concept is the better choice. It is in this sense that the concept used here is also designed.

For this reason, the management concept presented is central (with central network device **12**). Thus, the network manager **20** is a central entity and not distributed over a number of network nodes. As already mentioned, network manager **20** is responsible for all management tasks concerning the network. Network manager **20** knows the topology of the network **10** and knows the state of all hosts **18** located in the network. Within the context of the present concept, only the power management field is covered but the network manager **20** can also handle all other necessary management functions.

At the next level of hierarchy, there is an arbitrary number of switch devices **14** with switches **24**. These are "managed switches" having an arbitrary number of ports. The switch manager **22** of the switch device **14** is responsible for configuring and controlling the switch **24** and can communicate with other network nodes.

At the ports of these switches **24**, other switches **24** can be connected as the next level of hierarchy. In this context, there can be an arbitrary number of such hierarchy levels. The end points of these tree branches are the hosts **18**. The switches **24** themselves can also be located on host devices (hosts **18**) and form the switch devices. The switch manager **22** and the host software can run as two processes on one and the same CPU.

The switch manager **22** represents the communication partner of the network manager **20**. If a node is to be activated, the network manager **20** contacts the corresponding switch (manager) **22**, **24**, to which the relevant node is connected, and requests activation of the corresponding port. Thus, each active switch manager **22** must have a valid path to the network manager **20**.

As shown in FIG. **1**, it is possible to configure the network **10** as a tree having an arbitrary number of hierarchy levels.

The root of the tree is the central network device **12** with network manager **20** and switch **24**, to which it is connected. The following levels consist of hosts (end points) **18** or other switch devices **14** with switches **24** and, as a rule, with hosts **18** as interfaces to the respective next hierarchy level.

The port via which a switch **24** is connected to the next hierarchy level above will be called root port (analogously to the designation in the Spanning Tree protocol) in the text which follows. It is important that each active manageable node with switch manager **22** needs a valid connection to the network manager **20**.

In the text which follows, some basic principles, on which the power manager concept is based, will be described in detail.

The network function control signals **30** shown in FIG. **2**, designed as link pulses or NLPs, respectively, are short voltage pulses applied by a subscriber in the Ethernet to its transmitting line of network segments **26**, **28** whilst there is no transmitting traffic. Control signals **30** are used for testing the state of the link. A subscriber detects a link error when no pulses **32** and no data traffic are received for 50 ms-150 ms. In the case of 10 BASE-T Ethernet, these pulses **32** are called Link Integrity Test (LIT) pulses, in the case of 100 BASE-TX and auto negotiation, they are called Normal Link Pulses (NLP). Auto negotiation (100 BASE-TX) uses a sequence of up to 33 such pulses **32**, the communication parameters of the transmitter (speed, full- or half-duplex) being encoded in this sequence. These sequences are called "Fast Link Pulse (FLP) bursts".

The basic shape of the NLPs is shown in FIG. **2**. The precise specification of the pulse shape can be read up about in IEEE 802.3 Clause 14.3.1. The Link Integrity Test itself (that is to say also the sequence in time of pulses **32**) is specified in IEEE 802.3 Clause 14.2.1.7 (page 321).

The FLP bursts which are used for auto negotiation have the same shape but at a maximum, only 33 and at a minimum, 17 such pulses are sent spaced apart by 125 μs. The bursts are also spaced apart by 16 ms+/−8 ms.

If the PHY of a port is activated, it sends out such pulses. Depending on the configuration of the switch/controller (IOOSASE-TX), as FLP bursts (auto negotiation) or as NLPs, if auto negotiation is deactivated.

The Energy Detect Module (EDM) is a system capable of detecting NLPs or FLP bursts, and indicating their presence in a suitable form. It is necessary for each port on each device, i.e. a 4-port switch must have four EDMs or one EDM with four inputs and outputs. The EDM must be connected to the Rx line of the port but must not influence the reception of frames.

Advantageously, all IP addresses of the Ethernet network are static and known to the network manager (or the corresponding software, respectively). However, the concept is not restricted to this. The possibilities for issuing IP addresses are open. For example, dynamic assignment by means of DHCP would be possible. The way in which the network manager learns the IP addresses of the hosts **18** is open.

Furthermore, it is provided with advantage that in the initial state of the network system **10**, the network manager module **20** and the associated switch **24** of the central network device **12** are always active. This is made clear again by the active role of the central network device **12** in the method for activating the temporarily inactive network component of a network system. Although this permanent activity is not absolutely necessary since a device activated from "outside", during an attempt of establishing a connection to the network manager **20**, would mandatorily activate the device, but as a rule, the network management software will wish to establish a type of basic state of the network **10**, i.e. activate selected control devices. The concept is not restricted to this, either.

To provide a simpler description, the processes running in the network system (network) **10** to activate and deactivate hosts **18** and part-networks can be considered at two levels, namely the hardware level and the software level.

At the hardware level, the manner in which hosts **18** and switches **24** physically activate one another, deactivate one another, and can be notified by an external activation/deactivation is specified.

At the network level, it is defined how shutdown and wake-up processes (deactivation and activation processes) are running in the network, using the mechanisms specified at the hardware level.

Whilst the hardware level thus defines how two adjacent network nodes (hosts **18** and switches **24**) interact with one another physically, the network level specifies the principle according to which a node in the network can activate an arbitrary other node via the network management software, how the network management software activates the node and how it can deactivate a node.

Interactions at the Hardware Level

Two adjacent nodes, for example a switch **24** and a host **18**, must be capable of activating one another when it is necessary, to inform one another about any activation that has taken place or to deactivate one another when it is demanded by the network manager **20**.

The switch manager **22** treats each port of its switch **24** as a finite state machine (FSM). By this model, a port can be controlled and monitored in a simple manner. Thus, a manager of a 4-port switch simultaneously manages four mutually independent FSMs. Firstly, the model will be explained briefly here. After that, it is explained by using a state machine how the individual mechanisms of the hardware level are running.

Each port has four normal states:

UP means that the PHY of the respective port is in the activated, normal state and a valid link exists. Both sides (switch and host) transmit and receive NLPs and can transmit frames if required.

DOWN designates the deactivated state of the port. The PHY is in the power-down state, no NLPs are transmitted, data cannot be transmitted.

HOST STARTUP is a state of transition in which the PHY is activated and applies NLPs to the transmitting line. The host connected to the port is not yet active and does not yet send any link pulses. Thus, no valid link is recognized yet by the switch **24**.

HOST SHUTDOWN is also a state of transition, this time for shutting down a connected host **18**. The port PHY is deactivated and does not send any NLPs. The host **18** is, however, still active and sends NLPs which are indicated by the EDM of the port.

Furthermore, there are two error states:

LINK FAIL indicates that a previously valid link has unexpectedly broken down.

ERROR is a global error state into which the system changes on the occurrence of other errors, the type or cause of the error being stored.

Activating a host **18** designed as network component **16** by a switch occurs as follows.

A switch **24** must be capable of activating a deactivated host **18** connected to it on request of the network manager **20**. The so-called "port state machine" shown in FIG. **4** meets this requirement.

If the network manager **20** requests the activation of the host **18**, this leads to a state transition into the HOST STARTUP state. In this context, the PHY of the port is activated and begins to send NLPs. The state machine remains in this state as long as no NLPs are received from the host **18**.

If the EDM of the host **18**, that is to say the first activation device, detects the link pulses **32** (NLPs) of the switch device **14**, it triggers the booting process of host **18** in a suitable manner. As soon as the Ethernet controller of host **18** is started, its PHY begins in turn to send link pulses **32**. These are detected by the EDM of the switch, that is to say the second activation device, and indicated. This event leads to a change of state of the FSM into the UP state. Both sides detect NLPs, that is to say the link is valid and the connection has been established. It is now possible to transmit frames.

If an error occurs during the booting process of host **18** and no NLPs are sent back, a time out event takes place, the finite state machine (FSM) changes into the global error state and indicates a STARTUP_TIMEOUT.

Notification of the switch device **14** about an external activation by host **18** occurs as follows.

If a host **18** is not activated following an initiative of the network manager **20** but from the outside or by a user, respectively, and if its switch **24** is still deactivated, the host **18** must inform its switch about this so that it activates its port PHY.

The finite state machine (FSM) is in the DOWN state. As soon as the network controller of host **18** begins to send NLPs, these are detected by the EDM of the switch and indicated. This leads to a change of state from DOWN to UP, the PHY of the port being activated and in turn sending NLPs. Both sides will now detect NLPs, the link is valid and frames can be transmitted.

Activation of the switch device by host **18** occurs as follows.

As soon as the network controller of host **18** begins to send NLPs, they are detected by the EDM of the switch. The device that contains the switch manager **22** must now be booted in a suitable manner. The switch manager **22** must thereupon activate switch **24** and place it into its basic state. All FSMs are in the DOWN state after the booting process.

A switch device **14** activates a switch device **24** of the next hierarchy level below as follows.

If a switch **24** of the next hierarchy level below is to be activated, this is effected by the same mechanism. The hierarchically higher switch receives from the network manager **20** the request to activate the corresponding port. It makes no difference to it whether a host **18** or a switch **24** is connected to the port. The FSM changes into the HOSTSTARTUP state and the PHY is activated (NLPs are transmitted).

The EDM of the hierarchically lower switch **24** indicates the NLPs and switch **24** is started up and immediately activates its root port. This leads to a change of state from HOST STARTUP to UP in the hierarchically higher switch **24**, both sides detect link pulses and the connection is established.

A switch device **24** activates a switch device **24** of the next hierarchy level above as follows.

Host **18** is activated from the outside and in consequence activates its respective switch **24**. This, in turn, must activate the hierarchically next switch **24** above in order to establish a connection to the network manager **20**. For this purpose, the switch manager **22** must know via which port it is linked to the next hierarchy level above (it must know its root port). For example, a port number is defined which applies globally for all switches **24** as connection to the next hierarchy level (e.g. port **1**). Another possibility would be a memory entry which is specified individually for each switch.

According to the state diagram, the FSM of the root port changes immediately into the HOST STARTUP state after the booting. The higher-level switch is woken by the NLPs sent via the root port.

Deactivating host **18** by the switch device **14**/switch **24** occurs as follows. If a switch device **14** receives the request from the network manager **20** to deactivate a host (port) **18**, this can also be achieved by the port state machine. The request by the network managers **20** has the consequence that the FSM of the port changes from the UP state into the HOST-SHUTDOWN state, the PHY of the port being deactivated (no further NLPs are sent). The FSM remains in this state until no further NLPs arrive from the host **18**.

The network controller of the host will report a "link fail" as soon as it receives no further NLPs from the switch. This event can be used as trigger for the shutdown process of the host. However, as an alternative, the output of the host EDM could also be used since it will no longer indicate any link activity. The precise procedure remains open. If the host shuts down, it will stop itself sending NLPs. In the port FSM of the switch, this leads to a change in state from HOST SHUTDOWN to DOWN. The interface is thus deactivated, the host shutdown and the request is met.

If something goes wrong when shutting down the host and further NLPs are sent by it, the state machine of the switch changes into the ERROR state after a defined time and deposits a shutdown timeout as error.

Deactivating of a switch by a switch occurs as follows. If a switch is intended to deactivate a hierarchically lower switch, the mechanism proceeds similarly to the deactivation of a host by a switch. The case where a switch deactivates a hierarchically higher switch does not exist since it would cut its own connection to the network manager **20** by this which is impossible by definition.

It does not make a difference to the switch whether a host **18** or a switch **24** is connected to the port to be deactivated. The hierarchically lower switch (manager) to be deactivated knows via which port it is connected to the next hierarchy level above. If it receives no further NLPs from this port, this is the trigger for it to initiate shutdown. By definition, the network manager **20** has already switched off all ports of the switch **24** to be deactivated before it shuts down switch **24** itself.

Initially, the port FSM changes into the LINK FAIL state since no further NLPs are received. From there, the next change of state leads into the DOWN state, the interface is now deactivated. The switch manager can now initiate the shutdown process of the switch and then shut itself down.

Interactions at the network level—communication between network manager **20** and switch manager occurs as follows.

As already discussed, the network manager is responsible for requesting the switch managers to activate or deactivate their ports. Since the network manager knows the topology of the network and the state of the nodes at any time, it can thus establish any desired configuration in getting individual hosts or entire part-networks to become activated or deactivated. This presupposes that every active switch manager has a valid connection to the network manager (root of the tree). If the end point of a branch is active, the entire branch must therefore be active.

The manner of communication between the network manager and the switch managers is not established. It is the requirement that the network manager can inform the switch manager about its intention and the port which is involved and that the switch manager **22** can convey messages to the network manager **20** when one of its connected nodes has been activated. The network manager **20** can thus update its state table.

The Simple Network Management Protocol (SNMP), for example, is well suited for this. The Interface Management Information Base (IF-MIB) is available which, among other things, contains the managed object with the object ID (OID) (ifAdminStatus). This object specifies the desired state of an interface (port). If the network manager has an SNMP controller process and the switch managers have in each case an SNMP agent, the network manager **20** can send an SNMP SET packet to the switch manager **22** and set the object ifAdminstatus to the desired value. The notification about when a node has been activated can be carried out via an SNMP TRAP packet sent by the agent. In response to the TRAP, the switch manager **22** would have to send back an SNMP GET packet and read out the values of the ifAdminstatus object for each port and correspondingly update its state table. Since SNMP also offers a multiplicity of further management options, this basis can also be used as a platform for further network management.

A less extensive, slimmer method could be to define own wake-up, shut down and notification frames which contain the relevant port number. The disadvantage of the methods consists in that it is not standardized and does not offer an existing platform for further management functions.

Host requests are handled as follows. In most cases, an active host needs a "dialog partner" with which it can exchange data. As a rule, the network manager knows the existing dependencies between various hosts and will itself activate all necessary network nodes in the appropriate order.

However, the possibility also exists that a host needs a different host for a short term which, however, is currently inactive. The principle of centralized management does not provide for a host waking another one independently. The host must request the activation of the other one from the network manager. The network manager can then decide whether it wakes up the requested host (authorizations could play a role here) and, if necessary, perform the wake-up. As soon as the notification arrives from the "destination switch" that the requested host has been activated, the network manager can convey the confirmation to the host from which the request came.

Cascaded switches are extended as follows. It has already been said that each switch manager must have a path to the network manager, that is to say there can be no "gaps" in the branches of the tree.

Activation of a node via a switch **24** which is already active if a node is to be activated is connected to a switch **24** that can already be reached, the case is trivial. The network manager **20** requests switch **24** to activate the corresponding port.

Activation of a branch by the network manger **20** becomes more complicated if the node to be activated is on an as yet inactive switch or at the end of a branch of inactive switches, respectively. The network manager **20** knows the topology and must then wake up each switch **24** along the path to the destination node sequentially until it can reach the destination switch and can thus activate the desired node. Thus, the entire branch up to the destination node is activated.

Deactivation of a branch by the network manager **20** of a branch proceeds analogously. The network manager **20** knows the topology and deactivates sequentially all nodes which are located below the switch **24** actually to be switched off.

FIG. **3** illustrates the activation of a branch **34** by a user:

If a host **18**, which is part of an inactive branch **34** is activated from the outside, the "host activates switch" case initially occurs. Considering again the port state machine of switch **24**, the recursive propagation of the wake-up through the entire branch **34** becomes clear:

- The first switch **24**-A attempts to set up a connection to the (not yet accessible) network manager **20** and initially activates its root port. However, the root port FSM remains in the HOST STARTUP state until NLPs are received from the next switch **24**-B above.
- The next switch **24**-B above will boot up and will firstly activate again its root port in order to establish the connection to the network manager **20**. However, it remains in the HOST STARTUP state until it receives NLPs from its root port, that is to say from switch **24**-C. This pattern continues recursively up to the first switch **24**-Z already

active. If the entire branch **34** was inactive, this would be the highest switch **24** to which the network manager **20** itself is connected.

The switch **24**-Z, previously active, already has a connection to the network manager **20** and will respond immediately to the NLPs of the next switch **24**-Y below. At the same time, it will inform the network manager **20** about the activation of the corresponding port.

Since the next switch **24**-Y below now receives NLPs, the FSM of its root port changes from the HOST START to the UP state and there is a connection to the next switch **24**-Z below and thus to the network manager **20**. Switch **24**-Y can now respond to the next switch **24**-X below (the FSM of the port changes from HOST STARTUP to UP and NLPs are transmitted). At the same time, switch **24**-Y informs the network manager **20** about the activation of switch **24**-X.

This pattern continues up to the end of branch A so that the notifications are sent to the network manager in the "from top to bottom" order (the wake-ups, in contrast, propagated "from bottom to top" through the branch). This is necessary too, since a switch can notify the network manager only when it has a connection to it.

A port state machine is illustrated in FIG. **4**.

Starting from a "reset" (or "boot") state, the alternative events are initially obtained that the port is not a root port (E**0**) which leads to a "down" state of the port, or that the port is a root port (E**1**) which leads to a "startup" state.

However, the "startup" state can also be reached from the "down" state by a "wake-up" request (E**2**) by the network manager by means of activating the PHY actions (A**0**) and timer resetting (A**1**). If activity of the host **18** is lacking and the timer (E**3**) is running, the port remains in the "startup" state.

Starting from the "startup" state, the "up" state is reached by a host activity (E**4**) in which the network manager **20** is activated (A**2**). As an alternative, only an "error" state is reached from "startup" when host activity is lacking and there is a timeout (E**5**).

From the "up" state, the "shutdown" state is reached by a "shutdown" request (E**6**) by the network manager **20**, PHY being deactivated (A**3**) and the timer being reset (A**1**).

If the host **18** is subsequently still active and the timer is running (E**7**), the port remains in the "shutdown" state. If there is no further host activity (E**8**), the network manager **20** is notified and the "down" state is reached. If, in contrast, there is a timeout while the host **18** is still active (E**9**), the "error" state occurs.

From the "error" state, the "reset" state is reached by a reset (E**10**) in which PHY is deactivated (A**3**).

As an alternative to the "shutdown" request (E**6**) a "link fail" state is reached by a lacking link activity (E**11**) from the "up" state whereupon the Deactivate PHY action follows (A**3**) and the "down" state is reached.

This completely describes the port state machine shown in FIG. **4**.

In the text which follows, the concept is illustrated by selected examples with reference to FIG. **5**:

The network is a switched Ethernet that comprises three switches **24**-S**1**, **24**-S**2**, **24**-S**3** to which a plurality of hosts **18** are connected (see FIG. **5**). The switch managers **22** are switch manager **22**-alpha, switch manager **22**-beta and switch manager **22**-tau. Some hosts **18** can be activated and deactivated from the outside (e.g. by a user or an event) and are called "awakeable" in the text which follows. On the other hand, other hosts **18** can only be woken up and deactivated within the network, i.e. on the initiative of a switch **24** or of another host **18** (not wakeable).

Switch **24**-S**1** has a special role. It forms the root of the topology tree and its switch manager **22**-alpha is at the same time the central network manager **20**.

In the examples, SNMP is used for the communication between network manager **20** and the various switch managers **22**. The network manager **20** has an SNMP client for sending SNMP packets, the switch managers have an SNMP agent.

Activation of host **18**-omicron occurs as follows. This example demonstrates the activation of individual hosts **18** in the network by the network manager **20**. In the initial situation, the entire network system **10** is deactivated apart from the host **18**-S**1**/alpha constructed as central network device **12**. The network manager **20** of this central network device **12** would now like to establish a state in which the hosts **18**-omicron and **18**-epsilon are active.

Initially, host **18**-omicron is activated. It is connected directly to the root switch S**1** (port S**1**-**1**). The network manager **20** accesses the SNMP client of host **18**-alpha and sends an SNMP SET packet to the SNMP agent from host **18**-alpha. As a consequence, object "ifAdminstatus" for port S**1**-**1** is set to TRUE.

The SNMP agent informs the switch manager of switch **24**-S**1** about ifAdminstatus being set to true, which leads to a change in state of the state machine of the port from DOWN to HOST STARTUP and thus to the activation of the PHY of port S**1**-**1**, whereupon NLPs are sent out. The EDM of host **18**-omicron (a network component **16**) detects these NLPs and thereupon triggers the booting process of host **18**-omicron. As soon as the network controller of host **18**-omicron is active in turn and sends NLPs, the EDM of the switch **24**-S**1** detects this and a HOST STARTUP to UP change in state takes place. In this context, the network manager is informed that the node connected to port S**1**-**1** has been activated.

Activation of host **18**-epsilon (also a network component **16**):

Next, host **18**-epsilon is also to be activated. The network manager **20** knows that it can reach host **18**-epsilon via switch **24**-S**2** and thus via its port S**1**-**4**. It also knows that switch **24**-S**2** and its switch manager **22**-beta are still inactive and thus have to be activated first. Firstly, an SNMP SET packet is again sent to the SNMP client from host **16**-alpha (in this case the central network device **12**) in order to turn on port S**1**-**4**. According to "switch activates switch of the next hierarchy level below", the EDM of switch **24**-S**2** detects the NLPs, the switch **24** and switch manager **22** are started up and the root port of switch **24**-S**2** is activated. Switch **24**-S**2** and switch manager **22**-beta are now active and there is a valid connection. The network manager **20** is also notified that the node connected to port S**1**-**4** (that is to say S**2**/beta) has been activated.

To activate also host **18**-epsilon, the same procedure is adopted as in the previous case during the activation of host **18**-omicron. The only difference is that SNMP client and agent are no longer located in the same device but the SNMP SET packet is sent to switch manager **22**-beta.

Activation of host **18**-pi by the user is as follows.

The entire network system **10** apart from switch **24**-S**1** and switch manager **22**-alpha (which, at the same time, is network manager **20**), will be completely deactivated again. Host **18**-pi is activated from the outside (by a user or an event). The cascade consists of three switches, **24**-S**3**, **24**-S**2** and **24**-S**1**. Host **18**-pi firstly wakes up switch **24**-S**3** and switch manger **22**-tau and waits for NLPs from switch **24**-S**3**. As the first action, switch manager **22**-tau will activate the root port of S**3** (port S**3**-**1**) and wait for NLPs from switch **24**-S**2**.

Analogously, switch device **14**-S**2**/beta will wake up and switch manager **22**-beta, in turn, will activate the root port of switch **24**-S**2**, that is to say S**2**-**1**. Switch **24**-S**1**, which is already awake, thereupon turns on the PHY of port S**1**-**4**.

Switch device **14**-S**2**/beta detects the NLPs from switch **24**-S**1**. The root port FSM changes into the UP state and the switch manager **22** now begins to process the NLPs arriving from port S**2**-**4**. The FSM of this port changes into the UP state, the PHY is activated and NLPs are sent back. In addition, the network manager **20** is informed that the node connected to S**2**-**4** has been activated.

The same process now takes place one level lower with switch device **14**-S**3**/tau and port S**3**-**2**. As soon as the NLPs have then be answered by host **18**-pi, the network manager **20** is informed that the node connected to the port S**3**-**2** has been activated.

The wake-up has run through the cascade from bottom to top, but the notifications to the network manager **20** have done so from top to bottom.

Deactivation of the entire branch **34** occurs as follows. The network manager **20** decides that the branch **34** just activated (that is to say switch **24**-S**2**, **24**-S**3**, host **18**-pi) should be deactivated again. This is done from bottom to top: the network manager **20** will sequentially deactivate host **18**-pi, switch device **14**-S**3**/tau and switch device **14**-S**2**/beta. If, for example, host **18**-epsilon were still active, the network manager **20** would also deactivate it before it shuts down switch device **14**-S**2**/beta.

Initially, host **18**-pi is deactivated by an SNMP SET packet being sent to the switch manager **22**-tau, the ifAdminstatus object for port S**3**-**2** being set to FALSE. The FSM of the port changes into the HOST SHUTDOWN state, deactivates the PHY and in this state waits until no further NLPs are sent by host **18**-pi either. After that, the state changes to DOWN and the network manager **20** is informed that the node at port **53**-**2** has been deactivated.

After that, host **18**-alpha sends a further SNMP SET packet to host **18**-beta in order to analogously deactivate port S**2**-**4**. This shuts down the switch device **14**-S**3**/tau.

Lastly, host **18**-alpha sends an SNMP packet to itself in order to deactivate port S**1**-**4**. Switch device **14**-S**2**/beta is now also shut down and the entire branch **34** is deactivated.

Thus, while there have shown and described and pointed out fundamental novel features of the invention as applied to a preferred embodiment thereof, it will be understood that various omissions and substitutions and changes in the form and details of the devices illustrated, and in their operation, may be made by those skilled in the art without departing from the spirit of the invention. For example, it is expressly intended that all combinations of those elements and/or method steps which perform substantially the same function in substantially the same way to achieve the same results are within the scope of the invention. Moreover, it should be recognized that structures and/or elements and/or method steps shown and/or described in connection with any disclosed form or embodiment of the invention may be incorporated in any other disclosed or described or suggested form or embodiment as a general matter of design choice. It is the intention, therefore, to be limited only as indicated by the scope of the claims appended hereto.

The invention claimed is:

1. A method for activating at least one temporarily inactive network component of a network system for a vehicle, especially a motor vehicle, comprising:

transmitting by a central network device of the network system a signal to a first network component via a path inside the network system that leads at least partially across a network segment of the network system;

connecting at least partially in response to the signal the first network component and its associated first activation device connected unbranched to at least one switch device arranged in a path and to an associated second activation device; and

the central network device addressing the associated first activation device by the at least one switch device by sending a network function control signal whereby the at least one temporarily inactive machine network component is activated.

2. The method as claimed in claim **1**, further comprising:

activating the first network component by the associated first activation device after receipt of the network function control signal; and

subsequently the first network component sending out a further network function control signal to the associated second activation device to confirm the activation.

3. The method as claimed in claim **2**, wherein the second activation device brings the at least one switch device into a transmitting/receiving state after receipt of the further network function control signal.

4. The method as claimed in claim **1**, wherein the network system is an Ethernet network.

5. The method as claimed in claim **1**, wherein the network system has a tree topology formed by the central network device, the at least one switch device and the at least one network component.

6. The method as claimed in claim **1**, wherein at least one of the first network component and the central network device is at least part of a control device of a vehicle component.

7. A network system of a vehicle, configured to activate at least one temporarily inactive network component, comprising:

at least one network component having an associated first activation device;

at least one switch device;

a central network device connected for signal communication to the at least one network component via a path within the network system, the path leading at least partially across a network segment of the network system and the network segment connecting for signal communication the at least one network component and the associated first activation device unbranched to the at least one switch device arranged in the path and to an associated second activation device,

wherein the central network device addresses the associated first activation device via the at least one switch device by sending a network function control signal.

8. The network system as claimed in claim **7**, wherein the network system is an Ethernet network.

9. The network system as claimed in claim **7**, wherein the network system is a tree topology formed by the central network device, the at least one switch device and the at least one network component.

10. A motor vehicle comprising:

a network system configured to activate at least one temporarily inactive network component, having:

at least one network component having an associated first activation device;

at least one switch device;

a central network device connected for signal communication to the at least one network component via a path within the network system, the path leading at least partially across a network segment of the network system and the network segment connecting for signal communication the at least one network component and the associated first activation device unbranched to the at least one switch device arranged in the path and to an associated second activation device,

wherein the central network device addresses the associated first activation device via the at least one switch device by sending a network function control signal.

* * * * *