Title: LAYER STRUCTURE OF NETWORK CONTROL PROTOCOL AND INTERFACE METHOD

Layer Interface Structure

- Application Software
  - Application Layer
    - APDU Header
      - Message
      - Home Code Control Sublayer
        - Data Link Layer
          - Frame Header
            - Home Code Control PDU (HCNPDU)
        - Physical Layer
          - Frame

Abstract: The present invention relates to a layer structure of network control protocol and a method of interfacing to each layer. A user, for example, who is locating at home or out-of-home, can control the operation or monitor the operation state of various appliances such as refrigerator or laundry machine through RS-485 network, low output RF network, power line network, or the like which is installed at home, thereby providing convenience in remote control and monitoring for the user. Furthermore, for the interface to each layer of network control protocol, it is a very useful invention to implement the interface to each layer more effectively by using proper primitive setting for the interface to each layer, besides that a header and trailer information required for each layer is added to a protocol data unit (PDU) transferred from a higher level to be transferred to a lower level.
For two-letter codes and other abbreviations, refer to the "Guidance Notes on Codes and Abbreviations" appearing at the beginning of each regular issue of the PCT Gazette.
DESCRIPTION

LAYER STRUCTURE OF NETWORK CONTROL PROTOCOL AND INTERFACE METHOD

1. Technical Field

The present invention relates to a layer structure of network control protocol and an interface method thereof for a user, for example, who is locating at home or out-of-home to effectively control appliances such as refrigerator or laundry machine connected to a network.

2. Background Art

In general, 'home network' means a network in which various digital appliances are connected to one another for the user to enjoy economical home services in a convenient and safe way anytime at home or out-of-home, and due to the development of digital signal processing technology, various types of appliances such as refrigerator or laundry machine are being gradually digitalized.

On the other hand, in recent years, home network has been more advanced, since operating system and multi-media technology for appliances has been applied to digital appliances, as well as new types of information appliances have appeared.

Moreover, a network which is established for providing file exchanges or internet services between personal computers and peripheral devices, a network between appliances for handling audio or video information, and a network established for home automation of various appliances such as refrigerator or laundry machine, appliance control such as remote meter reading, and the like are called a 'living network' in a general meaning.

Furthermore, in the network services in which small-scale
data transmission for the remote control, or operating state monitoring of the appliances included in the aforementioned network, for example, various appliances such as refrigerator or laundry machine, is the main object of their communication, each of appliances connected to one another should be directly controlled by a network manager, which is included in the network, with the use of the minimum required communication resources. However, its effective solution has not been provided yet, and thus it is a matter of urgency to provide its solution.

3. Disclosure of the Invention

Accordingly, the present invention is devised in consideration of the aforementioned situation, and it is an object of the invention to provide a layer structure of network control protocol and an interface method thereof for more effectively implementing an interface to each layer of the network control protocol, besides that a user, for example, who is locating at home or out-of-home can effectively control various appliances such as refrigerator or laundry machine connected to a network by using the minimum required communication resources.

In order to achieve the aforementioned object, a layer structure of network control protocol according to the present invention comprises a physical layer for providing a function of sending/receiving physical signal between devices; a data link layer for providing a medium access control function to use a shared transmission media; a network layer for providing a function of address management and/or transmit/receive control for a network connection between devices; an application layer for providing a flow control function for download and/or upload services, and the layer structure is characterized in that the data link layer includes a home code control sub-layer for providing at least any one of the setting function, management function, and processing function for logically classifying individual networks.

Furthermore, an interface method between the layers of
network control protocol according to the present invention is characterized in that in a method of interfacing to each layer of network control protocol configured by including a plurality of layers, a primitive for transmitting data or a packet and a primitive for receiving data or a packet are used for the interface between each of the layers.

Furthermore, an interface method between the layers of network control protocol according to the present invention is characterized in that the master and slave are separated for the interface between an application software and an application layer, and a service request primitive transferred from the application software is used in the master, and a primitive for transferring a request message sent from the master to an application software of the slave is used in the slave.

4. Brief Description of the Drawings

The above and other objects, features and advantages of the present invention will become apparent from the following description of preferred embodiments given in conjunction with the accompanying drawings, in which:

Fig. 1 illustrates a configuration of a network system according to the present invention;

FIG. 2 and FIG.3 illustrate a master-slave driven communication structure according to the present invention;

FIG. 4 illustrates a layer structure of LnCP according to the present invention;

FIGS. 5 through 7 illustrate embodiments on the communication cycle services according to the present invention;

FIG. 8 illustrates a layer structure of LnCP according to the present invention;

FIG. 9 illustrates an embodiment on a primitive for the interface between a network management sub-layer and a parameter management layer according to the present invention;

FIG. 10 illustrates an embodiment of a structure for the interface to each layer according to the present invention;
FIG. 11 illustrates an embodiment of a structure of the Universal Asynchronous Receiver/Transmitter (UART) frame according to the present invention;
FIG. 12 illustrates an embodiment of a primitive for the interface between a physical layer and a data link layer according to the present invention;
FIG. 13 illustrates an embodiment of a primitive for the interface between a data link layer and a network layer according to the present invention;
FIG. 14 illustrates a structure of the data link layer frame according to the present invention;
FIG. 15 illustrates an embodiment of a primitive in the master for the interface between a network layer and an application layer according to the present invention;
FIG. 16 illustrates an embodiment of a primitive in the slave for the interface between a network layer and an application layer according to the present invention;
FIG. 17 and FIG. 18 illustrate an embodiment of a primitive in the master for the interface between an application layer and an application software according to the present invention;
FIG. 19 illustrates an embodiment of a primitive in the slave for the interface between an application layer and an application software according to the present invention;
FIG. 20 illustrates an embodiment of a primitive for the interface between a parameter management layer and a network management sub-layer according to the present invention;
FIG. 21 illustrates a layer structure of LnCP according to another embodiment of the present invention;
FIG. 22 illustrates an embodiment for transferring primitives between layers in the master according to the present invention; and
FIG. 23 illustrates an embodiment for transferring primitives between layers in the slave according to the present invention.

5. Best Mode for Carrying Out the Invention
Hereinafter, a preferred embodiment of a layer structure of network control protocol and an interface method thereof according to the present invention will be described in detail with reference to the accompanying drawings.

FIG. 1 illustrates a configuration of network control protocol according to the present invention. A network control protocol which will be defined in a new way in this invention, for example, a LnCP (Living network Control Protocol) Internet server 100 and a living network control system 400 to which the LnCP is applied, as illustrated in FIG. 1, are connected through the Internet; besides, the LnCP Internet server 100 allows to perform an interface operation with a variety of communication terminals 200 such as personal computer, PDA, PCS, and the like.

On the other hand, the living network control system 400 is configured by including a home gateway 40, a network manager 41, a LnCP router 42, a LnCP adaptor 43, and appliances 44; moreover, for the configured means, as illustrated in FIG. 1, the data link layer having a non-standard transmission medium such as RS-485 network or low output RF network, or the data link layer having a standard transmission medium such as power line communication, IEEE 802.11, or ZigBee (IEEE 802.15.4) will be used.

Furthermore, the living network control system 400 is called, for example, a 'LnCP network', and the LnCP network, as illustrated in FIG. 1, is configured with an independent network for connecting the appliances belonging to the category of living network in an independent home through wire or wireless communication media.

On the other hand, the LnCP network is connected with a master device for controlling the operation of other appliances or monitoring the operation state thereof, and a slave device having a function for responding to a request of the master device and a function for notifying information about the state change of its own.

In addition, an environmental setting and management function for the appliances 44 connected to the LnCP network

5
are taken in charge by the network manager 41, and the appliances 44 can be directly to a network or indirectly connected to a network through the LnCP adaptor 43; and RS-485 network, RF network, power line network and the like in the LnCP network are connected with one another through the LnCP router 42.

Furthermore, the LnCP network provides a function for monitoring or controlling the state of an appliance installed at home for a user in the outside, and the connection function between the LnCP network and the outside Internet for this purpose is taken in charge by the home gateway 40, and the user himself can use the function for monitoring or controlling the state of an appliance connected to the LnCP network when he accesses the LnCP Internet server 100 to pass through authentication process.

Moreover, the user can access the LnCP Internet server 100 from an appliance connected to the LnCP network through the home gateway 40 to download the contents provided by the LnCP Internet server. The main features of the LnCP network for this purpose will be described in detail in the following.

First, digital information appliances have microcontrollers with a variety of performances to implement a specific function respectively. In a LnCP network according to the present invention, such a function is more effectively simplified to enable the operation in the microcontrollers with a variety performances by using the minimum resources of the microcontroller mounted on an appliance, and in particular microcontrollers with low performance are allowed to implement appliance specific functions while processing a function of the LnCP communication, and microcontrollers with high performance are allowed to process a function of multi-tasking.

Moreover, the main features of the LnCP network according to the present invention will be classified into master-slave driven communication structure, event driven communication support, a plurality of network manager support, 4-layer structure, communication cycle service, flexible address management, variable length packet communication, and standard.
message set provision.

On the other hand, the master-slave driven communication structure is used as a connection communication structure between appliances in the LnCP network, and it should have at least more than one master device, and the master device should have the information and control codes for the slave devices to be controlled. At this time, the master device will control other slave devices by following an already inputted program or receiving the input from the user.

For example, a message flow between the master device and the slave device, as illustrated in FIG. 2, operates in a manner that when a Request message is sent from the master to the slave, the slave sends a Response message on this to the master, and the LnCP network may have a multi-master and multi-slave driven communication structure, as illustrated in FIG. 3.

Furthermore, in the LnCP network, event-driven communication services are supported, and for example, the user can establish an event which is required by an appliance, and thereafter if the event established by the user occurs in the relevant appliance while performing an operation, then it will notify the fact or content of the event occurrence to the other appliances, or control the operation state of the other appliances in response of the event.

Moreover, the LnCP network comprises at least more than one network manager in charge of an environmental setting and management function for appliances, and a number of network managers can be supported if necessary, and in this case, the management information of appliances should be synchronized in order to prepare for a plurality of the network manager errors.

Furthermore, the LnCP network, as illustrated in FIG. 4, has a 4-layer structure with physical layer, data link layer, network layer, and application layer, and services are provided with a communication cycle unit in the LnCP network, and only one communication cycle can exist at a given time in the slave device.

In other words, it cannot be controlled by any master
device while the communication cycle being performed in a slave device, but a plurality of communication cycles for a plurality of slave devices can be performed at a given time in the master device, and there are four kinds of \{1-Request, 1-Response\}, \{1-Request, Multi-Response\}, \{1-Notification\}, and \{Repeated-Notification\} communication cycles.

For example, the \{1-Request, 1-Response\} communication cycle is a communication cycle that a master sends a Request packet to a slave, and the slave transfers a Response packet in response of this, and if an error occurs in the received packet, then the master sends a Re-Request packet, and the slave in turn transfers a Response packet on this, as illustrated in FIG. 5.

Moreover, in the \{1-Request, Multi-Response\} communication cycle, as illustrated in FIG. 6, one master sends a Request packet having the group address to a number of slaves, and each of slaves sends a Response packet on the Request packet. The master finishes the cycle if the maximum allowable receive time is passed, and at this time the master ignores this even when an error occurs in the Response packet received from the slave.

Furthermore, the \{1-Notification\} communication cycle, as illustrated in FIG. 7, is a cycle that one master device sends a Notification packet to a number of devices, and then finishes the communication at once; moreover, the Repeated-Notification} communication cycle is a cycle that it sends the same packet repeatedly, and then finishes the communication in order to secure the transmission reliability in the \{1-Notification\} communication cycle.

On the other hand, the LnCP network supports flexible address management, and the appliances provided with a LnCP function, for example, can automatically configure a network without user’s intervention by assigning an address to each kind of appliances at the time of shipment from the factory, and here the network manager has an algorithm for assigning its own specific address when appliances are connected to one another, since the same kind of appliances are initialized.
with the same address.

Moreover, in the LnCP network, its own group address is specified for the appliances belonging to the same kind, and thereby the group communication is enabled by using one message, and several kinds of appliances can be classified into a cluster in order to specify the group address for each cluster.

Furthermore, the LnCP network supports the packet communication with a variable length, and in a case where contents such as application programs related to the manipulation of appliances are downloaded or the data stored in appliances are uploaded, the length of a packet can be controlled by using the exchanged buffer size information of the appliances.

Moreover, the LnCP network provides the standard message set, and the standard message set suited for various appliances, for example, is defined in the application layer to control other appliances from a master device, and furthermore the message set is divided into the common area message set for the basic LnCP communication, the product application area message set for supporting the specific function of appliances, and the developer area message set for providing the specific function of manufacturers.

On the other hand, the foregoing message set can be extended if necessary, and also arguments may be added to the already defined messages. Hereinafter, the layer structure, which is one of the main features in the LnCP network according to the present invention, will be described in more detail.

FIG. 8 illustrates a layer structure of the LnCP protocol according to the present invention, and as describe above, the LnCP network to which this invention is applied, has a 4-layer structure having physical layer, data link layer, network layer, and application layer for controlling and monitoring the operation of appliances such as refrigerator, laundry machine, or the like.

On the other hand, the physical layer provides a physical
interface between devices, and a function for transmitting/receiving physical signals such as bits to be transmitted, and for the physical layer, a non-standard transmission medium with the data link layer such as RS-485, low output RF network, etc. and a standard wire or wireless transmission medium such as power line communication, IEEE 802.11, Ethernet, and ZigBee, can be used, and the LnCP adaptor can be used as a separate physical layer in order to realize the physical layer of devices in the LnCP network.

Furthermore, the data link layer provides the medium access control (MAC) function to use a shared transmission medium, and for the LnCP network, p-DCSMA (probabilistic Delayed Carrier Sense Multiple Access) should be used as a protocol for the medium access control (MAC) in case where the data link layer uses a non-standard transmission medium.

However, in case where the data link layer uses a standard transmission medium for the LnCP network, the medium access control function as defined in the relevant protocol can be used.

On the other hand, as illustrated in FIG. 8, the home code control sub-layer provides a home code setting, management and processing function for dividing individual networks logically when the LnCP network is configured by using a non-independent transmission medium such as power line communication, IEEE 802.11, Ethernet, and ZigBee, or low output RF, and it is preferred that the home code control sub-layer is not realized in case where individual networks are physically divided by an independent transmission medium such as RS-485.

Furthermore, the network layer provides a function of appliance address management, transmit/receive control, etc. for reliable network connection between devices, and the application layer provides a flow control function for download and upload services as well as a transmit/receive control function in order to implement application software services.

In addition, the application layer defines message sets
for network management, or appliance control and monitoring, and the application software implements a specific function of appliances, and exchanges data with the application layer through the interface as defined in the application layer.

Furthermore, as illustrated in FIG. 8, the network management sub-layer provides a parameter management function for setting node parameters, and a network management function for network configuration and management, and the parameter management layer can set or read the parameters used in each layer according to a request of the network management sub-layer.

Moreover, in the primitive for interface to the network management sub-layer, as illustrated in FIG. 9, a primitive (structure SetPar) for transferring a parameter value from the network management sub-layer to the parameter management layer, and a primitive (structure GetPar) for transferring a parameter value from the parameter management layer to the network management sub-layer.

On the other hand, in the a primitive (structure SetPar) for transferring a parameter value to the parameter management layer, 'uchar DestLayer' for specifying a layer to which the parameter value is transferred, and 'structure SetLayerPar' that its value varies according to the DestLayer value as the parameter for each layer are written, and the DestLayer will be '1' in case where the layer to which the parameter value is transferred is an application layer, '2' in case of an network layer, '3' in case of a data link layer, and '4' in case of a physical layer.

Moreover, the SetLayerPar will be 'SetALPar' in case of an application layer, 'SetNLPar' in case of an network layer, 'SetDLLPar' in case of a data link layer, and 'SetPHYPar' in case of a physical layer.

Furthermore, in the primitive (structure GetPar) for transferring a parameter value to the network management sub-layer, 'uchar SreLayer' for specifying a layer to which the parameter value has been transmitted, 'uchar PMLResult' for specifying whether the parameter value is successfully taken
from each layer, and 'structure GetLayerPar' that its value varies according to the SrcLayer value as the parameter for each layer are written, and the SrcLayer will be '1' in case where the layer to which the parameter value has been transmitted is an application layer, '2' in case of an network layer, '3' in case of a data link layer, and '4' in case of a physical layer.

Moreover, the PMLResult will be PAR_OK(1) in case where the parameter value is successfully taken from each layer, and PAR_FAILD(0) in case where the parameter value is not successfully taken from each layer, and the GetLayerPar will be 'RptALPar' in case of an application layer, 'RptNLPar' in case of an network layer, 'RptDLLPar' in case of a data link layer, and 'RptPHYPar' in case of a physical layer.

On the other hand, for a parameter value which is used in the parameter management layer, there is 'const unit ParTimeOut', and it specifies a waiting time (ms) for receiving RptALPar(or RptNLPar, RptDLLPar, RptPHYPar) after transferring GetALPar(or GetNLPar, GetDLLPar, GetPHYPar) to each layer.

Furthermore, if the parameter management layer receives the SetPar primitive from the network management sub-layer, then SetALPar, SetNLPar, SetDLLPar, or SetPHYPar primitives is transferred to a layer specified in the primitive, and the variable of which all bit values are '1' in the received primitive will be ignored in each layer (for example, 0xFF, 0xFFFF).

Moreover, if the GetPar primitive is received from the network management sub-layer, then GetALPar, GetNLPar, GetDLLPar, or GetPHYPar is transferred to a layer specified in the primitive, and if RptALPar, RptNLPar, RptDLLPar or RptPHYPar is received from each layer, then the GetPar primitive is included and the PARResult value becomes PAR_OK to be transferred to the network management sub-layer. If the primitive is not received from each layer within the time of ParTimeOut, then the PARResult value becomes PAR_FAILED to be transferred to the network management sub-layer.
On the other hand, the network management sub-layer provides a parameter management function for setting node parameters in individual devices, and a function for network configuration, environmental setting, and network operation management, and if there is a request from the application software and the master, then a parameter value in the following will be set or read in the relevant layer through the parameter management layer.

For example, the parameter value of AddressResult, NP_AliveInt, SvcTimeOut, and NP_BufferSize in case of an application layer, NP_LogicalAddress, NP_ClusterCode, NP_HomeCode, and SendRetries in case of a network layer, MinPktInterval in case of a data link layer, and NP_bps in case of a physical layer will be set or read.

In particular, if the network management sub-layer in the slave receives the UserReqRcv primitive including an application service belonging to the 'device node parameter setting service' or 'device node parameter acquisition service' from the application layer, then parameter values will be set or read in the relevant layer through the parameter management layer, and the result will be transferred to the application layer through the UserResSend primitive. Application services for the parameter management of each layer are as follows.

For example, there are SetOption, SetAliveTiem, SetClock, and GetBufferSize services in case of an application layer; SetTempAddress, SetAddress, GetAddress services in case of an network layer; no relevant services in case of a data link layer; and SetSpeed service in case of a physical layer.

On the other hand, the network management sub-layer provides a network management function such as InCP network configuration, environmental setting, and network operation management, and general network management functions will operate on the application layer in the master, and a part of the synchronization functions of network information during a plurality of network management periods will operate on the application layer in the slave.
Furthermore, for the interface to the application layer, there are the interface to the application layer in the slave, and the interface to the application layer in the master. For the interface to the application layer in the slave, UserReqRcv and UserResSend primitives are used, and for the interface to the application layer in the master, UserReq, UserDLReq, UserULReq. UserRes, UserEventRcv, and ALCompleted primitives are used.

On the other hand, in a method of interfacing to each layer of network control protocol according to the present invention, as illustrated in FIG. 10, the header and trailer information required in each layer is added to the protocol data unit (PDU) transferred from a higher layer to be transferred to a lower layer.

For example, the APDU (Application layer PDU) as a packet transferred between an application layer and a network layer, comprises an APDU header and a message, and the NPDU (Network layer PDU) as a packet transferred between a network layer and a data link layer or home code sub-layer, comprises an NPDU header such as an ADPU and its own address, an address of the destination appliance, a packet type according to the importance of the transmitted message, a NPDU trailer, and an APDU.

Furthermore, the HCNPDU (Home Code Control Sub-layer PDU) as a packet transferred between a network layer and a data link layer comprises a NPDU and a home code. In the physical layer of the LnCP network, as illustrated in FIG. 11, a Universál Asynchronous Receiver/Transmitter (UART) frame structure will be used for the interface between a device and an LnCP adaptor or LnCP router.

For example, the packet received from the higher layer is converted into the UART frame unit with 10 bits to be transferred through a transmission medium, and the UART frame in a LnCP network comprises a start bit with 1-bit, a data with 8-bit, and a stop bit with 1-bit, and a parity bit is not used, and the UART frame is sequentially transferred from the start bit to the stop bit.
On the other hand, in case where the UART is used in the LnCP network, additional frame header and frame trailer are not used, and 9600 bps will be used for the communication speed, or 4800 bps or 19200 bps will be used according to the performance of the device.

Furthermore, as illustrated in FIG. 12, in the primitives for the interface between a physical layer and a data link layer, there are the primitive 'FrameSend' for transferring 1 byte data from the data link layer to the physical layer, the primitive 'FrameRcv' for transferring 1 byte data from the physical layer to the data link layer, and the primitive 'RptLineStatus' in the line state that is transferred to data link layer. If a UART frame in the line state exists, then LINE_BSY is transferred, and otherwise LINE_IDLE is transferred.

On the other hand, as illustrated in FIG. 13, in the primitives for the interface between a data link layer and a network layer, there are the primitive 'PktSend' for transferring a packet from the network layer to the data link layer, the primitive 'PktRcv' for transferring a packet from the data link layer to the network layer, and the primitive 'DLLCompleted' for notifying a result of the packet transmission from the data link layer to the network layer.

Furthermore, a packet (NPDU/HCNPDU) of the network layer, a byte data length of the NPDU/HCNPDU, and transmission priority (SvcPriority) are written in the PktSend; a packet (PDU) of the network layer and a byte data length of the PDU (PDULength) are written in the PktRcv; and if a packet transmission procedure is successfully completed as a result of the packet transmission(DLLResult), then SEND_OK(1) is written, and otherwise SEND_FAILED(0) is written, and in case where the DLLResult is SEND_FAILED(0), a value for classifying the failure reason (DLLFailCode) is written in the DLLCompleted.

On the other hand, as illustrated in FIG. 14, a data link frame structure is configured by adding the frame header and trailer to the NPDU/HCNPDU, and if a non-standard transmission
medium is used in the data link layer, then a null field will be written in the frame header and trailer, and if a standard transmission medium is used, then it will follow the regulation as defined in the relevant protocol. The NPDU field is a data unit that is transferred from the higher network layer.

Furthermore, the HCNPDU is a data unit that a home code, which is used in case where the physical layer is a non-independent transmission medium, is added in front of the NPDU, and the NPDU and HCNPDU will not be divided for processing in the data link layer.

On the other hand, the interface to the network layer will be different for the master and slave. In the master, as illustrated in FIG. 15, ReqMsgSend, MsgRcv, and NLCompleted primitives will be used for the interface between a network layer and an application layer. In the primitive ReqMsgSend for transferring a message from the application layer of the master to the network layer, an identification number of the communication cycle (CycleID), an ADPU including a request message created in the application layer of the master (ReqADPU), a byte data length of the ADPU (APDULength), an address of the receiver device (DstAddress), an address of the sender device (SrcAddress), a type of communication cycle service in the master (NLService, example: 0=Acknowledged, 1=Non-acknowledged, 2=Repeated-notification), a waiting time for a response packet after sending a request packet to the master in case where the NLService is selected as 'Acknowledged' (Response TimeOut), a time period between consecutive notification packets in case where the NLService is selected as 'Repeated-notification' (RepNotiInt), and a transmission priority of the request message (SvcPriority) will be written.

Moreover, in the primitive MsgRcv for transferring a packet from the network layer of the master to the application layer, an identification number of the communication cycle (CycleID), an APDU for being transferred to the application layer (ResEventAPDU), a byte data length of the ADPU
(APDULength), an address of the receiver device (DstAddress), and an address of the sender device (SrcAddress) will be written.

Furthermore, in the primitive NLCompleted for
5 transferring a packet process state from the network layer to the application layer, an identification number of the communication cycle (CycleID) and a result of the communication cycle implementation (NLResult) are written. Here, if the communication cycle is successfully completed,
10 then CYCLE_OK(1) will be written, and otherwise CYCLE_FAILED(0) will be written; and if the NLService is CYCLE_FAILED, then a value for classifying the failure reason (NLFailCode) will be written, and if the NLResult is CYCLE_OK, then the number of repeated transmission (NLSuccessCode) will be written.

On the other hand, as illustrated in FIG. 16, in the slave, ReqMsgRcv, ReqMsgSend, EventMsgSend, and NLCompleted primitives will be used for the interface between a network layer and an application layer. In the ReqMsgRcv for
20 transferring a request message received to the application layer from the network layer of the slave, an APDU for being transferred to the application layer (ReqAPDU), a byte data length of the APDU (APDULength), an address of the receiver device (DstAddress), an address of the sender device (SrcAddress), a type of communication cycle service in the slave (NLService, example: 0=Acknowledged, 1=Non-acknowledged), and if a result of the duplicated packet detection is normal, then NORMAL_PKT(1) will be written, and if it is detected as a repeated packet, then DUPLICATED_PKT(0) will be written.

Moreover, in the primitive ResMsgSend for transferring a response message from the application layer of the slave to the network layer, an identification number of the communication cycle (CycleID), an ADPU including a response message created in the application layer of the slave (ResADPU), and a byte data length of the ADPU (APDULength) will be written.

Furthermore, in the primitive EventMsgSend for
transferring an event message from the application layer of
the slave to the network layer, an identification number of
the communication cycle (CycleID), an ADPU including an event
message created in the application layer of the slave
(EventADPU), a byte data length of the ADPU (APDULength), an
address of the receiver device (DstAddress), an address of the
sender device (SrcAddress), a transmission service in the
network layer (NLService, example: 1=Non-acknowledged,
2=Repeated-notification), a time period between consecutive
notification packets in case where the NLService is selected
as 'Repeated-notification' (RepNotiInt), and a transmission
priority of the event message (SvcPriority) will be written.

On the other hand, in the primitive NLCompleted for
transferring a packet process state from the network layer to
the application layer, an identification number of the
communication cycle (CycleID) and a result of the
communication cycle implementation (NLResult) will be written.
Here, if the communication cycle is successfully completed,
then CYCLE_OK(1) will be written, and otherwise

CYCLE_FAILED(0) will be written; and if the NLService is
CYCLE_FAILED, then a value for classifying the failure reason
(NLPailCode) will be written, and if the NLResult is CYCLE_OK,
then the number of repeated transmission (NLSuccessCode) will
be written.

Furthermore, as illustrated in FIG. 17 and FIG. 18, the
master will use UserReq, UserDLReq, UserULReq, UserRes,
UserEventRcv, and ALCompleted primitives for the interface to
application software. In the service request primitive UserReq
transferred from the application software in the master, a
combination of product codes and command codes as a service
code of the application layer (ALSvcCode), a request message
comprising command codes and input arguments (ReqMsg), a byte
data length of the ReqMsg (ReqMsgLength), an address of the
receiver device (DstAddress), a type of transmission service
(ALService, example: 0=Request-response message, 1=Request-
message-only, 2=Repeated-message, 3=Event-message-only), a
waiting time (ms) for a response packet after sending a
request packet in the master in case where the ALService is a Request-response-message or a time period between consecutive notification messages in case where the ALService is a Repeated-message (TimeOut), and a transmission priority (SvcPriority) will be written.

Moreover, in the primitive UserRes for transferring a result of the service implementation on a request message of the master to the application software, a combination of product codes and command codes as a service code of the application layer (ALSvcCode), a response message comprising command codes and return arguments (ResMsg), a byte data length of the ResMsg (ResMsgLength), and an address of the sender device (SrcAddress) will be written.

On the other hand, in the primitive UserEventRcv for transferring an event message received from the slave to an application software of the master, a combination of product codes, command codes and event codes as a service code of the application layer (ALSvcCode), an event message received from the slave (EventMsg), a byte data length of the EventMsg (EventMsgLength), an address of the sender device (SrcAddress) will be written.

Furthermore, in the request primitive UserDLReq for sending data to a device connected to the network by a network manager, a combination of product codes and command codes as a service code of the application layer (ALSvcCode), a data file name to be downloaded (DownloadFile), a type of the transmission service fixed to 'Request-response-message(0)' (ALService), an address of the receiver device (DstAddress), a waiting time of a response packet after sending a request packet in the network manager (TimeOut), and a transmission priority fixed to '1' (SvcPriority) will be written.

Moreover, in the request primitive UserULReq for receiving data from a device connected to the network by a network manager, a combination of product codes and command codes as a service code of the application layer (ALSvcCode), a data file name to be uploaded (UploadFile), a type of the transmission service fixed to 'Request-response-message(0)'
(ALService), an address of the receiver device (DstAddress), a
waiting time of a response packet after sending a request
packet in the network manager (TimeOut), and a transmission
priority fixed to '1' (SvcPriority) will be written.

Furthermore, in the ALCompleted primitive for
transferring a result of the service implementation from an
application layer of the master to an application software, a
combination of product codes and command codes as a service
code of the application layer (ALSvcCode), and a result of the
service implementation (ALResult) will be written; and if a
service requested by the application software is successfully
completed, then SERVICE_OK(1) will be written, and otherwise
SERVICE_FAILED(0) will be written, and a value for classifying
the failure reason (ALFailCode) will be written in case where
the ALResult is 'SERVICE_FAILD'.

On the other hand, as illustrated in FIG. 19, the slave
will use UserResSend, UserReqRcv, and UserEventSend primitives.
In the primitive UserReqRcv for transferring a request message
(including download and upload) sent from the master to an
application software of the slave, a combination of product
codes and command codes as a service code of the application
layer (ALSvcCode), a data included in the request message sent
from the master (ReqData), a length of the ReqData
(ReqDataLength), and an address of the sender device
(SrcAddress) will be written.

Moreover, in the primitive UserResSend for transferring a
response message on a request message of the master to an
application layer of the slave, a combination of product codes
and command codes as a service code of the application layer
(ALSvcCode), a data included in the response message to be
sent to the master (ResData), and a length of the ResData
(ResDataLength) will be written.

Furthermore, in the primitive UserEventSend for
transferring a state variable value of the event message in
the slave to be transmitted to the master, a combination of
product codes, command codes and event codes as a service code
of the application layer (ALSvcCode), a type of the
transmission service (ALService, example: 2=Repeated-message, 3=Event-message-only), an event code (EventCode), and a state variable value of the event message (StateValue) will be written.

On the other hand, as illustrated in FIG. 20, the interface between a parameter management layer and a network management sub-layer will use SetPar and GetPar primitives. In the SetPar for transferring a parameter value from the network management sub-layer to the parameter management layer, a layer to which the parameter value is transferred (DestLayer) will be written; for example, and the DestLayer will be written as '1' in case of an application layer, '2' in case of an network layer, '3' in case of a data link layer, and '4' in case of a physical layer.

Furthermore, for a parameter (SetLayerPar) for each layer that varies according to the DestLayer value, 'SetALPar' in case of an application layer, 'SetNLPAR' in case of an network layer, 'SetDLLPar' in case of a data link layer, and 'SetPHYPar' in case of a physical layer will be written.

Moreover, in the primitive GetPar for transferring a parameter value from the parameter management layer to the network management sub-layer, a layer to which the parameter value has been transmitted (SrcLayer) will be written; for example, '1' in case of an application layer, '2' in case of an network layer, '3' in case of a data link layer, and '4' in case of a physical layer will be written.

Furthermore, a result of the implementation (PMLResult) will be written, and it will be PAR_OK(1) in case where the parameter value is successfully taken from each layer, and otherwise it will be PAR_FAILD(0), and for a parameter (GetLayerPar) for each layer that varies according to the SrcLayer value, 'RptALPar' in case of an application layer, 'RptNLPAR' in case of an network layer, 'RptDLLPar' in case of a data link layer, and 'RptPHYPar' in case of a physical layer will be written.

On the other hand, FIG. 21 illustrates a layer structure of LnCP according to another embodiment of the present
invention, and the layer structure of LnCP has a four layer structure of physical layer, data link layer, and network layer, and application layer. The application layer includes an application sub-layer for managing a network management function and a device information object, and the data link layer includes a home code control sub-layer.

Moreover, the physical layer provides a physical interface between devices and a function of sending and receiving a physical signal such as bits to be transmitted. For the physical layer, the data link layer having a non-standard transmission medium such as RS-485 network or low output RF network, or having a standard wire or wireless transmission medium such as power line communication, IEEE 802.11, or ZigBee (IEEE 802.15.4) will be used, and the LnCP adaptor as a separate physical layer will be used in order to realize the physical layer of devices in the LnCP network.

Furthermore, the data link layer provides a medium access control (MAC) function to use a shared transmission medium, and for the LnCP network, p-DCSMA (probabilistic Delayed Carrier Sense Multiple Access) may be used as a protocol for the medium access control (MAC) in case where the data link layer uses a non-standard transmission medium, and sends/receives services to/from the network layer through a data link service access point (DL-SAP).

On the other hand, as illustrated in FIG. 21, the home code control sub-layer provides a home code setting, management and processing function for logically dividing individual networks when the LnCP network is configured by using a non-independent transmission medium such as power line communication, IEEE 802.11, Ethernet, and ZigBee, or low output RF, and it is preferable that the home code control sub-layer is not realized in case where individual networks are physically divided by an independent transmission medium such as RS-485.

Furthermore, the network layer provides a function of appliance address management, transmit/receive control, etc. for reliable network connection between devices, and
sends/receives services to/from the application layer through a network link service access point (NL-SAP).

On the other hand, the application sub-layer is included in the application layer, and provides a network management function and a device information object, and the network management function provides a parameter management function for setting node parameters and a function for network configuration and management, and the device information object receives and manages device information from a device, and provides a reply function for the related request.

Moreover, the application layer provides a transmit/receive control function in order to implement a service of application software, and additionally provides a flow control function for download/upload services, and defines a message set for the network management or appliance control and monitoring, and sends/receives services to/from the application software through an application layer service access point (AL-SAP).

Furthermore, the application software performs a specific function of appliances, and exchanges data with the application layer through an interface defined in the application layer. As illustrated in FIG. 21, the parameter management plane can set or read parameters used in each layer according to a request of the network management function in the application layer.

On the other hand, as another embodiment according to the present invention, as illustrated in FIGS. 22 and 23, a layer structure of LnCP protocol in the master and slave has a four layer structure of physical layer, data link layer, and network layer, and application layer. The application layer includes an application sub-layer and a device information object, and the data link layer includes a home code control sub-layer.

Furthermore, in the master, as illustrated in FIG. 22, UserReq, UserDReq, UserULReq, ALCompleted, UserRes, and UserEventRcv primitives will be used for the interface to application software. In the service request primitive UserReq
transferred from the application software in the master, a combination of product codes and command codes as a service code of the application layer (ALSvcCode), a request message comprising command codes and input arguments (ReqMsg), a byte data length of the ReqMsg (ReqMsgLength), an address of the receiver device (DstAddress), a type of transmission service (ALService), a waiting time (ms) for a response packet after sending a request packet in the master in case where the ALService is a Request-response-message or a time period between consecutive notification messages in case where the ALService is a Repeated-message (TimeOut), and a transmission priority (SvcPriority) will be written.

Moreover, in the primitive UserRes for transferring a result of the service implementation on a request message of the master to the application software, a combination of product codes and command codes as a service code of the application layer (ALSvcCode), a response message comprising command codes and return arguments (ResMsg), a byte data length of the ResMsg (ResMsgLength), and an address of the sender device (SrcAddress) will be written.

On the other hand, in the primitive UserEventRcv for transferring an event message received from the slave to an application software of the master, a combination of product codes, command codes and event codes as a service code of the application layer (ALSvcCode), an event message received from the slave (EventMsg), a byte data length of the EventMsg (EventMsgLength), an address of the sender device (SrcAddress) will be written.

Furthermore, in the request primitive UserDLReq for sending data to a device connected to the network by a network manager, a combination of product codes and command codes as a service code of the application layer (ALSvcCode), a data file name to be downloaded (DownloadFile), a type of the transmission service fixed to 'Request-response-message(0)' (ALService), an address of the receiver device (DstAddress), a waiting time of a response packet after sending a request packet in the network manager (TimeOut), and a transmission
priority fixed to '1' (SvcPriority) will be written.

Moreover, in the request primitive UserULReq for receiving data from a device connected to the network by a network manager, a combination of product codes and command codes as a service code of the application layer (ALSvcCode), a data file name to be uploaded (UploadFile), a type of the transmission service fixed to 'Request-response-message(0)' (ALSvcService), an address of the receiver device (DstAddress), a waiting time of a response packet after sending a request packet in the network manager (TimeOut), and a transmission priority fixed to '1' (SvcPriority) will be written.

Furthermore, in the ALCompleted primitive for transferring a result of the service implementation from an application layer of the master to an application software, a combination of product codes and command codes as a service code of the application layer (ALSvcCode), and a result of the service implementation (ALResult) will be written; and if a service requested by the application software is successfully completed, then SERVICE_OK(1) will be written, and otherwise SERVICE_FAILED(0) will be written, and a value for classifying the failure reason (ALFailCode) will be written in case where the ALResult is 'SERVICE_FAILLD'.

On the other hand, as illustrated in FIG. 23, the slave will use UserResSend, UserReqRcv, and UserEventSend primitives. In the primitive UserReqRcv for transferring a request message (including download and upload) sent from the master to an application software of the slave, a combination of product codes and command codes as a service code of the application layer (ALSvcCode), a data included in the request message sent from the master (ReqData), a length of the ReqData (ReqDataLength), and an address of the sender device (SrcAddress) will be written.

Moreover, in the primitive UserResSend for transferring a response message on a request message of the master to an application layer of the slave, a combination of product codes and command codes as a service code of the application layer (ALSvcCode), a data included in the response message to be
sent to the master (ResData), and a length of the ResData (ResDataLength) will be written.

Furthermore, in the primitive UserEventSend for transferring a state variable value of the event message in
5 the slave to be transmitted to the master, a combination of product codes, command codes and event codes as a service code of the application layer (ALSvcCode), a type of the transmission service (ALService), an event code (EventCode), and a state variable value of the event message (StateValue) will be written.

On the other hand, the interface between a parameter management plane and a network management sub-layer will use SetPar and GetPar primitives. In the SetPar for transferring a parameter value from the network management sub-layer to the parameter management layer, a layer to which the parameter value is transferred (DestLayer) will be written; for example, and the DestLayer will be written as '1' in case of an application layer, '2' in case of an network layer, '3' in case of a data link layer, and '4' in case of a physical layer.

Furthermore, for a parameter (SetLayerPar) for each layer that varies according to the DestLayer value, 'SetALPar' in case of an application layer, 'SetNLPar' in case of an network layer, 'SetDLLPar' in case of a data link layer, and 'SetPHYPar' in case of a physical layer will be written.

Moreover, in the primitive GetPar for transferring a parameter value from the parameter management layer to the network management sub-layer, a layer to which the parameter value has been transmitted (SrcLayer) will be written; for example, '1' in case of an application layer, '2' in case of an network layer, '3' in case of a data link layer, and '4' in case of a physical layer will be written.

Furthermore, a result of the implementation (PMLResult) will be written, and it will be PAR_OK(1) in case where the parameter value is successfully taken from each layer, and otherwise it will be PAR_FAILD(0), and for a parameter (GetLayerPar) for each layer that varies according to the SrcLayer value, 'RptALPar' in case of an application layer,
'RptNLPar' in case of an network layer, 'RptDLLPar' in case of a data link layer, and 'RptPHYPar' in case of a physical layer will be written.

The layer structure of network control protocol and the interface method having the foregoing configuration according to the present invention will provide convenience in remote control and monitoring for the user, thereby more effectively implementing the interface to each layer of the network control protocol.

As described above, while the present invention has been disclosed for the purpose of illustration with reference to the aforementioned preferred embodiment, the living network may be called a network with a different name, furthermore, more various appliances can be connected to this network, and it will be understood by those skilled in the art that the foregoing embodiment can be improved, modified, substituted or added in a variety of ways without departing from the technical spirit and scope of the invention as defined by the appended claims.
CLAIMS

1. A layer structure of network control protocol comprising:
   a physical layer for providing a function of
   sending/receiving physical signal between devices;
   a data link layer for providing a medium access control
   function to use a shared transmission media;
   a network layer for providing a function of address
   management and/or transmit/receive control for a network
   connection between devices; and
   an application layer for providing a flow control
   function for download and/or upload services,
   wherein the data link layer includes a home code control
   sub-layer for providing at least any one of the setting
   function, management function, and processing function for
   logically classifying individual networks.

2. The layer structure of network control protocol
   according to claim 1, wherein in the physical layer, a non-
   standard transmission medium or a standard wire or wireless
   transmission medium is used, and a separate adapter including
   a communications module is selectively used for connection.

3. The layer structure of network control protocol
   according to claim 1, wherein the application layer includes
   an application sub-layer for providing at least any one of a
   network management function, and a function of providing a
   device information object.

4. The layer structure of network control protocol
   according to claim 3, wherein the network management function
   provides a parameter management function for setting node
   parameters, and a function for network configuration and
   management.

5. The layer structure of network control protocol
   according to claim 3, wherein the device information object
   receives and manages device information from a device, and
   provides a reply function for the related request.
6. The layer structure of network control protocol according to claim 1, wherein the application layer defines a message set for the network management or device control and monitoring, and exchanges data with an application software for performing a specific function of the device.

7. The layer structure of network control protocol according to claim 1, wherein the network control protocol further includes an application software for performing a specific function of the device and performing a data exchange with the application layer through an interface defined in the application layer.

8. A method of interfacing to each layer of network control protocol configured by including a plurality of layers, wherein a primitive for transmitting data or a packet and a primitive for receiving data or a packet are used for the interface between each of the layers.

9. The method of interfacing to each layer of network control protocol according to claim 8, wherein a plurality of the layers are configured by including at least any one of a physical layer, a data link layer, a network layer, and an application layer.

10. The method of interfacing to each layer of network control protocol according to claim 9, wherein for the interface between the physical layer and the data link layer among a plurality of the layers, at least any one of a primitive for transferring data from the data link layer to the physical layer, a primitive for transferring data from the physical layer to the data link layer, and a primitive in a line state for being transferred to the data link layer is used.

11. The method of interfacing to each layer of network control protocol according to claim 10, wherein in case of the primitive in a line state for being transferred to the data link layer, if a Universal Asynchronous Receiver/Transmitter (UART) frame exists, then LINE_BSY is transferred, and otherwise LINE_IDLE is transferred.

12. A method of interfacing to each layer of network control protocol.
control protocol according to claim 9, wherein for the interface between the data link layer and the network layer among a plurality of the layers, at least any one of a primitive for transferring a packet from the network layer to the data link layer, a primitive for transferring a packet from the data link layer to the network layer, and a primitive for notifying a result of the packet transmission from the data link layer to the network layer is used.

13. The method of interfacing to each layer of network control protocol according to claim 12, wherein at least any one of a packet of the network layer, a byte data length of the packet, and a transmission priority is written in the primitive for transferring a packet from the network layer to the data link layer.

14. The method of interfacing to each layer of network control protocol according to claim 12, wherein at least any one of a packet of the network layer and a byte data length of the packet is written in the primitive for transferring a packet from the data link layer to the network layer.

15. The method of interfacing to each layer of network control protocol according to claim 12, wherein in case of the primitive for notifying a result of the packet transmission from the data link layer to the network layer, if a packet transmission procedure is successfully completed as a result of the packet transmission, then SEND_OK(1) is written, and otherwise SEND_FAILED(0) is written, and in case of a transmission failure, a value for classifying the failure reason (DLLFailCode) is written.

16. The method of interfacing to each layer of network control protocol according to claim 9, wherein the master and slave are separated for the interface between each of the layer, and at least any one of a primitive for transferring a message from the application layer to the network layer, a primitive for transferring a packet from the network layer to the application layer, and a primitive for notifying a packet processing state from the network layer to the application layer is used among a plurality of the layers in the master.
17. The method of interfacing to each layer of network control protocol according to claim 16, wherein at least any one of a recognition number of communication cycle, an ADPU including a request message created in an application layer of the master, a byte data length of the ADPU, an address of the receiver device, an address of the sender device, and a type of the communication cycle service in the master is written in the primitive for transferring a message from the application layer to the network layer.

18. The method of interfacing to each layer of network control protocol according to claim 16, wherein at least any one of a recognition number of communication cycle, an ADPU for being transferred to an application layer, a byte data length of the ADPU, an address of the receiver device, and an address of the sender device is written in the primitive for transferring a packet from the network layer to the application layer.

19. The method of interfacing to each layer of network control protocol according to claim 16, wherein a recognition number of communication cycle and a result of communication cycle implementation are written in the primitive for transferring a packet processing state from the network layer to the application layer, and if the communication cycle is successfully completed, then CYCLE_OK(1) is written, and otherwise CYCLE_FAILED(0) is written.

20. The method of interfacing to each layer of network control protocol according to claim 9, wherein the master and slave are separated for the interface between each of the layers, and at least any one of a primitive for transferring a received request message from the network layer to the application layer, a primitive for transferring a response message from the application layer to the network layer, a primitive for transferring an event message from the application layer to the network layer, and a primitive for notifying a packet processing state from the network layer to the application layer is used among a plurality of the layers in the slave.
21. The method of interfacing to each layer of network control protocol according to claim 20, wherein at least any one of an ADPU for being transferred to an application layer, a byte data length of the ADPU, an address of the receiver device, an address of the sender device, a type of communication cycle service in the slave, and a result of repeated packet detection is written in the primitive for transferring a received request message from the network layer to the application layer.

22. The method of interfacing to each layer of network control protocol according to claim 20, wherein at least any one of a recognition number of communication cycle, an ADPU including a response message created in an application layer, a byte data length of the ADPU is written in the primitive for transferring a response message from the application layer to the network layer.

23. The method of interfacing to each layer of network control protocol according to claim 20, wherein at least any one of a recognition number of communication cycle, an ADPU including a event message created in an application layer of the slave, a byte data length of the ADPU, an address of the receiver device, an address of the sender device, and a transmission service in the network layer is written in the primitive for transferring an event message from the application layer to the network layer.

24. A method of interfacing to each layer of network control protocol, wherein the master and slave are separated for the interface between an application software and an application layer, and a service request primitive transferred from the application software is used in the master, and a primitive for transferring a request message sent from the master to an application software of the slave is used in the slave.

25. The method of interfacing to each layer of network control protocol according to claim 24, wherein the service request primitive transferred from the application software is used in the master, and
additionally at least any one of a primitive for transferring a result of service implementation on a request message to the application software, a primitive for transferring an event message received from the slave to an application software of the master, a request primitive for sending data to a device connected to the network by a network manager, a request primitive for receiving data from a device connected to the network by a network manager, and a primitive for transferring a result of service implementation from an application layer of the master to the application software is used.

26. The method of interfacing to each layer of network control protocol according to claim 24, wherein a primitive for transferring a request message sent from the master to the application software of the slave is used in the slave, and additionally at least any one of a primitive for transferring a response message on a request message of the master to an application layer of the slave, and a primitive for transferring a state variable value of an event message of the slave to be transmitted to the master to the application layer is used.

27. The method of interfacing to each layer of network control protocol according to claim 25, wherein at least any one of a combination of product codes and command codes, a request message (ReqMsg) comprising command codes and input arguments, a byte data length of the ReqMsg, an address of the receiver device, and a type of transmission service is written as a service code of the application layer, in the service request primitive transferred from the application software.

28. The method of interfacing to each layer of network control protocol according to claim 25, wherein at least any one of a combination of product codes and command codes, a response message (ResMsg) comprising command codes and return arguments, a byte data length of the ResMsg, and an address of the sender device is written as a service code of the application layer, in the primitive for transferring a result of service implementation on a request message to the
application software.

29. The method of interfacing to each layer of network control protocol according to claim 25, wherein at least any one of a combination of product codes, command codes and event codes, an event message (EventMsg) received from the slave, a byte data length of the EventMsg, and an address of the sender device is written as a service code of the application layer, in the primitive for transferring an event message received from the slave to an application software of the master.

30. The method of interfacing to each layer of network control protocol according to claim 25, wherein at least any one of a combination of product codes and command codes, a data file name to be downloaded, a type of transmission service fixed to a specified value, an address of the receiver device, a waiting time for a response packet after sending a request packet from a network manager, and a transmission priority fixed to a specified value is written as a service code of the application layer, in the request primitive for sending data to a device connected to the network by a network manager.

31. The method of interfacing to each layer of network control protocol according to claim 25, wherein at least any one of a combination of product codes and command codes, a file name for storing the data to be uploaded, a type of transmission service fixed to a specified value, an address of the receiver device, a waiting time for a response packet after sending a request packet from a network manager, and a transmission priority fixed to a specified value is written as a service code of the application layer, in the request primitive for receiving data from a device connected to the network by a network manager.

32. The method of interfacing to each layer of network control protocol according to claim 25, wherein at least any one of a combination of product codes and command codes and a result of service implementation is written as a service code of the application layer, in the primitive for transferring a result of service implementation from the application layer of
the master to the application software.

33. The method of interfacing to each layer of network control protocol according to claim 26, wherein at least any one of a combination of product codes and command codes, a data (ReqData) included in a request message sent from the master, a length of the ReqData, and an address of the sender device is written as a service code of the application layer, in the primitive for transferring a request message sent from the master to an application software of the slave.

34. The method of interfacing to each layer of network control protocol according to claim 26, wherein at least any one of a combination of product codes and command codes, a data (ResData) included in a response message to be sent to the master, and a length of the ResData is written as a service code of the application layer, in the primitive for transferring a response message on a request message of the master to an application layer of the slave.

35. The method of interfacing to each layer of network control protocol according to claim 26, wherein at least any one of a combination of product codes, command codes, and event codes, a type of transmission service, an event code, and a state variable value of an event message is written as a service code of the application layer, in the primitive for transferring a state variable value of an event message of the slave to be transmitted to the master to the application layer.
FIG. 2

Device 1

Master

Request

Device 2

Slave

Response

FIG. 3

Multi-Master and Multi-Slave

Master

Slave

Master

Slave

Master

Slave
FIG. 4

LnCP Layer Structure
FIG. 5

Device 1

Master

Request

Response

Device 2

Slave

Device 1

Master

Request

Error

Re-Request

Response

Device 2

Slave

1 Request & 1 Response
FIG. 6

1 Request & Multi-Response
FIG. 7

Device 1

Master

Notification

Device 2

Slave

Device 3

Slave

1 Notification
FIG. 8

LnCP Protocol Stack

Application Software

Network Management Sublayer

Application Layer

Network Layer

Home Code Control Sublayer

Parameter Management Layer

p-DCSMA

RS-485

RF

PLC Protocol

Wireless Protocol (IEEE 802.11 ZigBee)
FIG.9

Primitive

structure SetPar {
    uchar DestLayer;
    structure SetLayerPar;
}

structure GetPar {
    uchar SrcLayer;
    uchar PMLResult;
    structure GetLayerPar;
}
FIG. 10

Layer Interface Structure

<table>
<thead>
<tr>
<th>Layer</th>
<th>Contents</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application Software</td>
<td>Message</td>
</tr>
<tr>
<td>Application Layer</td>
<td>APDU Header</td>
</tr>
<tr>
<td></td>
<td>Message</td>
</tr>
<tr>
<td>Network Layer</td>
<td>NPDU Header</td>
</tr>
<tr>
<td></td>
<td>APDU</td>
</tr>
<tr>
<td></td>
<td>NPDU Trailer</td>
</tr>
<tr>
<td>Home Code Control Sublayer</td>
<td>Home Code</td>
</tr>
<tr>
<td></td>
<td>NPDU</td>
</tr>
<tr>
<td>Data Link Layer</td>
<td>Frame Header</td>
</tr>
<tr>
<td></td>
<td>HCNPDU</td>
</tr>
<tr>
<td></td>
<td>Frame Trailer</td>
</tr>
<tr>
<td>Physical Layer</td>
<td>Frame</td>
</tr>
</tbody>
</table>
FIG. 11

UART Frame Structure

Logic 1

Start 0 1 2 3 4 5 6 7 Stop

Logic 0

Bit LSB MSB Bit

FIG. 12

Primitive for interface between physical layer and datalink layer

```
structure FrameSend {
    uchar UART_byte;
}

structure FrameRcv {
    uchar UART_byte;
}

structure RptLineStatus {
    uchar LineStatus;
}
```
FIG. 13

Primitive for interface between datalink layer and network layer

```
structure PktSend {
    uchar NPDU/HCNPDU;
    uchar NPDULength;
    uchar SvcPriority;
}

structure PktRcv {
    uchar PDU;
    uchar PDULength;
}

structure DLLCompleted {
    uchar DLLResult;
    uchar DLLFailCode;
}
```
FIG.14

Data Link Layer Frame Structure

<table>
<thead>
<tr>
<th>Frame Header</th>
<th>NPDU/HCNPDU</th>
<th>Frame Trailer</th>
</tr>
</thead>
</table>

FIG. 15

Primitive for interface between network layer and application layer in master

```c
structure ReqMsgSend {
    uchar CycleID;
    uchar ReqAPDU;
    uchar APDULength;
    unit DstAddress;
    unit SrcAddress;
    uchar NLService;
    unit ResponseTimeOut;
    unit RepNotiInt;
    uchar SvcPriority;
}

structure MsgRcv {
    ulong CycleID;
    uchar ResEventAPDU;
    uchar APDULength;
    unit DstAddress;
    unit SrcAddress;
}

structure NLCompleted {
    ulong CycleID;
    uchar NLSuccessCode;
    uchar NLSuccessCode;
}```
FIG. 16

Primitive for interface between network layer
and application layer in slave

structure ReqMsgRcv {
    uchar ReqAPDU;
    uchar APDULength;
    unit DstAddress;
    unit SrcAddress;
    uchar NLService;
    uchar DuplicateCheck;
}

structure ResMsgSend {
    ulong CycleID;
    uchar ResAPDU;
    uchar APDULength;
}

structure EventMsgSend {
    ulong CycleID;
    uchar EventAPDU;
    uchar APDULength;
    unit DstAddress;
    unit SrcAddress;
    uchar NLService;
    uchar RepNotiInt;
    uchar SvcPriority;
}

structure NLCOMPLETED {
    ulong CycleID;
    uchar NLResult;
    uchar NLFallCode;
    uchar NLSuccessCode;
}
FIG. 17

Primitive for interface between application layer
and Application Software in master

```c
structure UserReq {
    ulong ALSvcCode;
    RequestMessageReqMsg;
    uchar ReqMsgLength;
    unit DstAddress;
    uchar ALService;
    unit TimeOut;
    uchar TimeOut;
    uchar SvcPriority;
}
structure UserRes {
    ulong ALSvcCode;
    ResponseMessage ResMsg;
    uchar ResMsgLength;
    unit SrcAddress;
}
structure UserEventRcv {
    ulong ALSvcCode;
    EventMessage EventMsg;
    uchar EventMsgLength;
    unit SrcAddress;
}
structure UserDLReq {
    ulong ALSvcCode;
    FILE DownloadFile;
    uchar ALService;
    unit DstAddress;
    unit TimeOut;
    uchar SvcPriority;
}
FIG.18

Primitive for interface between application layer and Application Software in master

```
structure UserULReq {
    ulong ALSvcCode;
    File UploadFile;
    uchar ALService;
    unit DstAddress;
    unit TimeOut;
    uchar SvcPriority=1;
}
structure ALCompleted {
    ulong ALSvcCode;
    uchar ALResult;
    uchar ALFailCode;
}
```
FIG. 19

Primitive for interface between application layer and Application Software in slave

structure UserReqRcv {
    uchar ALSvcCode;
    uchar ReqData;
    uchar ReqDataLength;
    unit SrcAddress;
}

structure UserResSend {
    uchar ALSvcCode;
    uchar ResData;
    uchar ResDataLength;
}

structure UserEventSend {
    uchar ALSvcCode;
    uchar ALService;
    unit EventCode;
    uchar StateValue;
}
FIG.20

Primitive for interface between parameter management layer and network management sublayer

```c
structure SetPar {
    uchar DestLayer;
    structure SetLayerPar;
}

structure GetPar {
    uchar SrcLayer;
    uchar PMLResult;
    structure GetLayerPar;
}
```
Fig. 21

LnCP Protocol Stack

Application Software

AL-SAP

Application Layer

Message set

Application Sublayer

Network Management Function

Device Information Object

Parameter Management Plane

Network Layer

NL-SAP

DataLink Layer

Homecode Control Sublayer

Physical Layer

RS-485

RF

PLC Protocol

802.15.4
FIG. 23

Slave

Application Software

SetPar
GetPar

UserResEnd
UserEventSend
UserReqRcv

SetALPar
GetALPar
RptALPar

RequestSend
EventMsgSend
NLCompleted
ReqMsgRcv

SetNLPar
GetNLPar
RptNLPar

Network Layer

PktSend
DLLCompleted
PktRcv

Slave

Application Sublayer

SetDLLPar
GetDLLPar
RptDLLPar

DataLink Layer

Homecode Control Sublayer

SetPHYPar
GetPHYPar
RptPHYPar

Physical Layer

RS-485
RF
PLC Protocol

802.15.4
**INTERNATIONAL SEARCH REPORT**

A. **CLASSIFICATION OF SUBJECT MATTER**

**H04L 12/28(2006.01)j**

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

B. **FIELDS SEARCHED**

Minimum documentation searched (classification system followed by classification symbols)

IPC 8: H04L 12/28

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

- Korean Patents and applications for inventions since 1975
- Korean Utility models and applications for Utility models since 1975
- Japanese Utility models and application for Utility models since 1975

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

- eKIPASS(KIPO internal) "LnCP", "home network"

C. **DOCUMENTS CONSIDERED TO BE RELEVANT**

<table>
<thead>
<tr>
<th>Category</th>
<th>Citation of document, with indication, where appropriate, of the relevant passages</th>
<th>Relevant to Claim No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y</td>
<td>WO 2004/107088 A2 (LG ELECTRONICS, INC.) 9 December 2004 See page 11 lines 25 - page 14 lines 5, figs. 2-3B, claims 1-3, 7-10</td>
<td>1, 8, 24</td>
</tr>
<tr>
<td>Y</td>
<td>WO 2004/107092 A2 (LG ELECTRONICS, INC.) 9 December 2004 See page 11 lines 13 - page 13 lines 19, figs. 2-3B</td>
<td>1, 8, 24</td>
</tr>
</tbody>
</table>

☐ Further documents are listed in the continuation of Box C.  
☒ See patent family annex.

Date of the actual completion of the international search  
12 JUNE 2006 (12.06.2006)

Date of mailing of the international search report  
13 JUNE 2006 (13.06.2006)

Name and mailing address of the ISA/KR  
Korean Intellectual Property Office  
920 Dunsan-dong, Seo-gu, Daejeon 302-701, Republic of Korea

Facsimile No. 82-42-472-7140

Form PCT/ISA/210 (second sheet) (April 2005)

Authorized officer  
JEON, Yong Hai  
Telephone No. 82-42-481-5657
<table>
<thead>
<tr>
<th>Patent document cited in search report</th>
<th>Publication date</th>
<th>Patent family member(s)</th>
<th>Publication date</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>AU2003246126AA</td>
<td>21.01.2005</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AU2003246141A1</td>
<td>21.01.2005</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AU2003246141AA</td>
<td>21.01.2005</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AU2003246146AA</td>
<td>21.01.2005</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AU2003246151A1</td>
<td>21.01.2005</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AU2003246151AA</td>
<td>21.01.2005</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AU2003246156AA</td>
<td>21.01.2005</td>
</tr>
<tr>
<td></td>
<td></td>
<td>US20040240451A1</td>
<td>02.12.2004</td>
</tr>
<tr>
<td></td>
<td></td>
<td>US2004240451A1</td>
<td>02.12.2004</td>
</tr>
<tr>
<td></td>
<td></td>
<td>US2004240451AA</td>
<td>02.12.2004</td>
</tr>
<tr>
<td></td>
<td></td>
<td>W02004107088A2</td>
<td>09.12.2004</td>
</tr>
<tr>
<td></td>
<td></td>
<td>W02004107088A3</td>
<td>16.06.2005</td>
</tr>
<tr>
<td></td>
<td></td>
<td>W02004107089A2</td>
<td>09.12.2004</td>
</tr>
<tr>
<td></td>
<td></td>
<td>W02004107089A3</td>
<td>16.06.2005</td>
</tr>
<tr>
<td></td>
<td></td>
<td>W02004107090A2</td>
<td>09.12.2004</td>
</tr>
<tr>
<td></td>
<td></td>
<td>W02004107090A3</td>
<td>16.06.2005</td>
</tr>
<tr>
<td></td>
<td></td>
<td>W02004107090A2</td>
<td>09.12.2004</td>
</tr>
<tr>
<td></td>
<td></td>
<td>W02004107091A2</td>
<td>09.12.2004</td>
</tr>
<tr>
<td></td>
<td></td>
<td>W02004107091A3</td>
<td>16.06.2005</td>
</tr>
<tr>
<td></td>
<td></td>
<td>W02004107091A2</td>
<td>09.12.2004</td>
</tr>
<tr>
<td></td>
<td></td>
<td>W02004107092A2</td>
<td>09.12.2004</td>
</tr>
<tr>
<td></td>
<td></td>
<td>W02004107092A3</td>
<td>16.06.2005</td>
</tr>
<tr>
<td></td>
<td></td>
<td>W02004107092A2</td>
<td>09.12.2004</td>
</tr>
<tr>
<td></td>
<td></td>
<td>W02004107093A2</td>
<td>09.12.2004</td>
</tr>
<tr>
<td></td>
<td></td>
<td>W02004107093A3</td>
<td>20.01.2006</td>
</tr>
<tr>
<td></td>
<td></td>
<td>W02004107569A2</td>
<td>09.12.2004</td>
</tr>
<tr>
<td></td>
<td></td>
<td>W02004107657A1</td>
<td>09.12.2004</td>
</tr>
<tr>
<td></td>
<td></td>
<td>W02004107658C2</td>
<td>05.01.2006</td>
</tr>
<tr>
<td></td>
<td></td>
<td>W02004107659A1</td>
<td>09.12.2004</td>
</tr>
<tr>
<td></td>
<td></td>
<td>W02004107661A1</td>
<td>09.12.2004</td>
</tr>
<tr>
<td></td>
<td></td>
<td>W02004107663A1</td>
<td>09.12.2004</td>
</tr>
<tr>
<td></td>
<td></td>
<td>W02004107708A1</td>
<td>09.12.2004</td>
</tr>
<tr>
<td></td>
<td></td>
<td>W02004107710A1</td>
<td>09.12.2004</td>
</tr>
<tr>
<td>Patent document cited in search report</td>
<td>Publication date</td>
<td>Patent family member(s)</td>
<td>Publication date</td>
</tr>
<tr>
<td>----------------------------------------</td>
<td>-----------------</td>
<td>-------------------------</td>
<td>------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AU2003246126AA</td>
<td>21.01.2005</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AU2003246141A1</td>
<td>21.01.2005</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AU2003246141AA</td>
<td>21.01.2005</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AU2003246146A1</td>
<td>21.01.2005</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AU2003246151A1</td>
<td>21.01.2005</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AU2003246151AA</td>
<td>21.01.2005</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AU2003246156AA</td>
<td>21.01.2005</td>
</tr>
<tr>
<td></td>
<td></td>
<td>US20040240451A1</td>
<td>02.12.2004</td>
</tr>
<tr>
<td></td>
<td></td>
<td>US2004240451A1</td>
<td>02.12.2004</td>
</tr>
<tr>
<td></td>
<td></td>
<td>US2004240451AA</td>
<td>02.12.2004</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WO2004107088A2</td>
<td>09.12.2004</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WO2004107088A3</td>
<td>16.06.2005</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WO2004107088A2</td>
<td>09.12.2004</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WO2004107089A2</td>
<td>09.12.2004</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WO2004107089A3</td>
<td>16.06.2005</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WO2004107090A2</td>
<td>09.12.2004</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WO2004107090A3</td>
<td>16.06.2005</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WO2004107090A2</td>
<td>09.12.2004</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WO2004107091A2</td>
<td>09.12.2004</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WO2004107091A3</td>
<td>16.06.2005</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WO2004107091A2</td>
<td>09.12.2004</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WO2004107092A2</td>
<td>09.12.2004</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WO2004107092A3</td>
<td>16.06.2005</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WO2004107093A2</td>
<td>09.12.2004</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WO2004107093A3</td>
<td>20.01.2006</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WO2004107569A2</td>
<td>09.12.2004</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WO2004107657A1</td>
<td>09.12.2004</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WO2004107658C2</td>
<td>05.01.2006</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WO2004107659A1</td>
<td>09.12.2004</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WO2004107661A1</td>
<td>09.12.2004</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WO2004107663A1</td>
<td>09.12.2004</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WO2004107708A1</td>
<td>09.12.2004</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WO2004107710A1</td>
<td>09.12.2004</td>
</tr>
</tbody>
</table>