EIBA Handbook Series

Release 3.0

Volume 3:
System Specifications

Part 3: Medium Independent Layers

Chapter 3: Network Layer

23.03.1999

Table of Contents

1. Overview

The network layer (in the following also called "layer-3") is the layer between the data link layer and the network layer user. The EIB network layer conforms to the definitions of the ISO/OSI model (ISO 7498) network layer.

The network layer uses the L_Data service of the data link layer and provides N_Data, N_GroupData and N_Broadcast services to the network layer user, see Fig. 3/3/3-1.

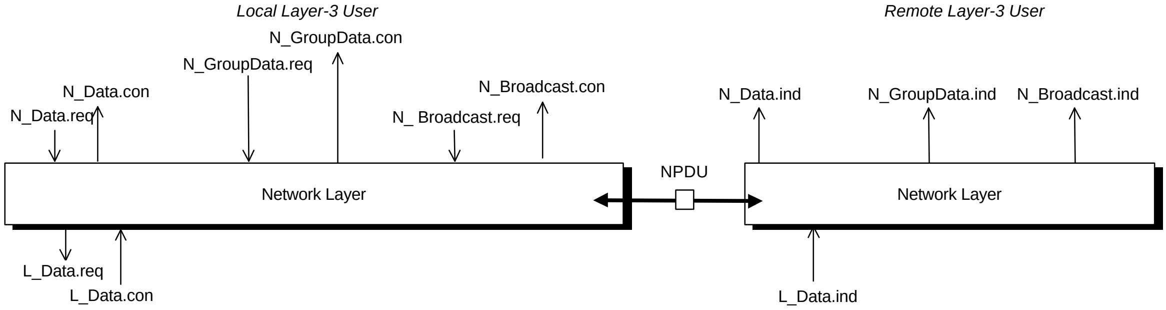

Fig. 3/3/3-1: Interactivity of the Network layer in respect to End Devices

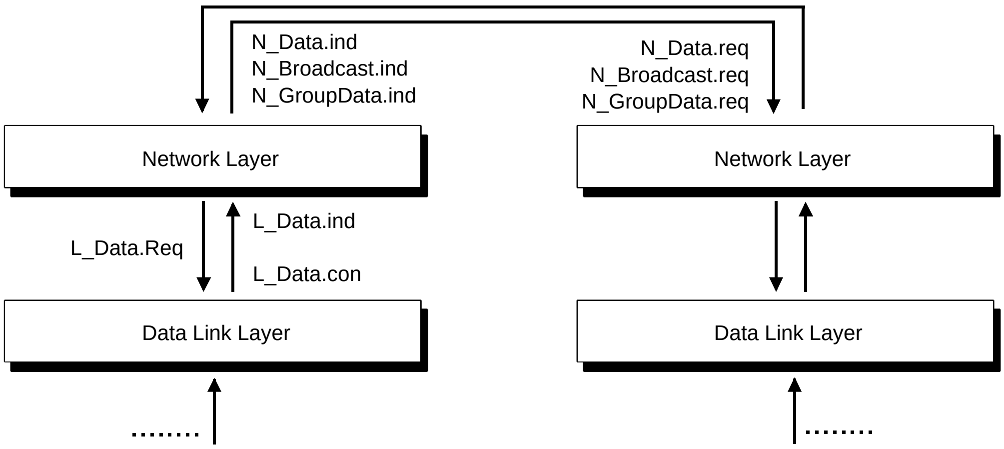

Fig. 3/3/3-2: General Functionality of a Router or Bridge

The network layer offers a segmentwise acknowledged datagram service for intersegmentary end-to-end communication. Intersegmentary end-to-end communication with filter characteristics needs devices called routers, see paragraph 2.6 "State Machine of Layer-3 for Routers". Routers are EIB devices which allow to couple two layer-2 protocol instances together, which are connected to different segments. A router delivers only messages from one segment to the other if the destination address of the received message is contained in the router's routing table. Furthermore a router allows to remove misrouted messages so that they cannot flood the network. A major cause for misroutings are wrong routing table entries. All the routing tables of an EIB network together define the allowed communication paths between any two end devices.

Intersegmentary end-to-end communication without filter characteristics needs devices called bridges, see also Fig. 3/3/3-2. Like a router a bridge allows to couple two layer-2 protocol instances together, which are connected to different segments. But a bridge does not have the filter property of a router and therefore does not need any routing table.

Two different network layer users must be distinguished:

- The Network Layer user in an end device: This is the Transport Layer, see Chapter 3/3/4 "Transport Layer".

- The network layer user in a router: That user is a download server for the routing table. See Chapter 3/4/1 "User Layer".

2. Network Services and Protocol

2.1 NPDU

The NPDU corresponds to the LPDU of an L_Data-Frame without the controlfield, source address, destination address, destination addressflag and the information length. The NPDU is shown in the following figure (Fig. 3/3/3-3).

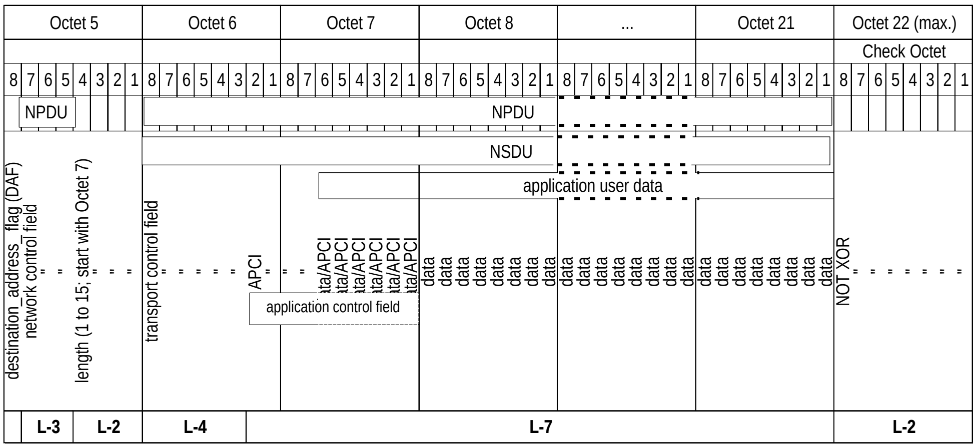

Fig. 3/3/3-3: Format of the NPDU

2.2 Network Layer Services

2.2.1 N_Data Service and End Device Network Protocol

The N_Data service is confirmed locally. The local user of layer-3 prepares an NSDU for the remote user of layer-3, the destination is addressed with a physical address. The local user of layer-3 applies the N_Data.req primitive to pass the NSDU to the local layer-3. The local layer-3 accepts the service request and passes it with an L_Data.req with DAF = ´0´ to the local layer-2.

The local layer-3 encodes the NSDU to the LSDU by adding the routing_counter and mapping the arguments destination_address and class to the corresponding arguments destination_address and class of the L_Data.req primitive.

The remote layer-3 maps an L_Data.ind primitive with DAF = ‘0’ to an N_Data.ind primitive. The argument l_sdu is mapped to the argument n_sdu. The arguments destination_address and class are mapped to the corresponding arguments destination_address and class of the N_Data.ind primitive.

The local layer-3 maps the L_Data.con primitive to the N_Data.con primitive. The argument l_status is mapped to the corresponding argument n_status of the N_Data.con primitive.

```
N_Data.req(destination_address, class, n_sdu)
 destination_address:           physical address of the destination
 class:                         system, alarm, high or low priority
 n_sdu:                         this is the user data to be transferred by
                                layer-3

N_Data.con(n_status)
 n_status: OK;                  N_Data sent successfully with L_Data
           not_ok;              transmission of the associated L_Data
                                request frame didn't succeed

N_Data.ind(source_address, destination_address, class, n_sdu)
 source_address:                physical address of the EIB end device that
                                requested the N_Data service
 destination_address:           the physical address of this device
 class:                         system, alarm, high or low priority
 n_sdu:                         this is the user data that has been
                                transferred by layer-3
```

2.2.2 N_Groupdata Service and End Device Network Protocol

The N_Groupdata service is confirmed locally. The local user of layer-3 prepares an NSDU for the remote user of layer-3, the destination is addressed with a group address. The local user of layer-3 applies the N_GROUPDATA.req primitive to pass the NSDU to the local layer-3. The local layer-3 accepts the service request and passes it with an L_Data.req with DAF = ´1´ to the local layer-2.

The local layer-3 encodes the NSDU to the LSDU by adding the routing_counter and mapping the arguments destination_address and class to the corresponding arguments destination_address and class of the L_Data.req primitive.

The remote layer-3 maps an L_Data.ind primitive with DAF = '1' and destination_address<>'0' to an N_Groupdata.ind primitive. The argument l_sdu is mapped to the argument n_sdu. The arguments destination_address and class are mapped to the corresponding arguments destination_address and class of the N_Groupdata.ind primitive.

The local layer-3 maps the L_Data.con primitive to the N_Groupdata.con primitive. The argument l_status is mapped to the corresponding argument n_status of the N_Groupdata.con primitive.

```
N_Groupdata.req(destination_address, class, n_sdu)
 destination_address:           group address of the destination
 class:                         system, alarm, high or low priority
 n_sdu:                         this is the user data to be transferred by
                                layer-3

N_Groupdata.con(destination_address, n_status)
 destination_address:           group address of the destination
 n_status: OK;                  N_Groupdata sent successfully with L_Data
                                service
           not_ok;              transmission of the associated L_Data
                                request frame didn't succeed
```

```
N_Groupdata.ind(source_address, destination_address, class, n_sdu)
 source_address:                  physical address of the EIB end device that
                                  requested the N_Groupdata service
 destination_address:             the addressed group address of this device
 class:                           system, alarm, high or low priority
 n_sdu:                           this is the user data that has been
                                  transferred by layer-3
```

2.2.3 N_Broadcast Service and End Device Network Protocol

The N_Broadcast service is confirmed locally. The local user of layer-3 prepares an NSDU for all the remote user of layer-3, the destination is addressed with the broadcast address (destination address = ´0´ and DAF = ´1´). The local user of layer-3 applies the N_Broadcast.req primitive to pass the NSDU to the local layer-3. The local layer-3 accepts the service request and passes it with an L_Data.req with DAF = ´1´ to the local layer-2.

The local layer-3 encodes the NSDU to the LSDU by adding the routing_counter and mapping the arguments destination_address and class to the corresponding arguments destination_address and class of the L_Data.req primitive.

The remote layer-3 maps an L_Data.ind primitive with DAF = ´1´ and destination_address = ´0´ to an N_Broadcast.ind primitive. The argument l_sdu is mapped to the argument n_sdu. The argument class is mapped to the corresponding argument class of the N_Broadcast.ind primitive.

The local layer-3 maps the L_Data.con primitive to the N_Broadcast.con primitive. The argument l_status is mapped to the corresponding argument n_status of the N_Broadcast.con primitive.

```
N_Broadcast.req(class, n_sdu)
 class:                           system, alarm, high or low priority
 n_sdu:                           this is the user data to be transferred by
                                  layer-3

N_Broadcast.con(n_status)
 n_status: OK;                    N_Broadcast sent successfully with L_Data
                                  service
           not_ok;                transmission of the associated L_Data
                                  request frame didn’t succeed

N_Broadcast.ind(source_address, class, n_sdu)
 source_address:                  physical address of the EIB end device that
                                  requested the N_Broadcast service
 class:                           system, alarm, high or low priority
 n_sdu:                           this is the user data that has been
                                  transferred by layer-3
```

2.3 Parameters of Layer-3

The following parameters influence the behavior of layer -3 and are required inside layer-3 in order to operate correctly:

routing_counter   will be added to the frame by Layer-3

device_type   information about the device: either EIB end device or bridge or router.

2.4 State Machine of Layer-3 for End Devices

The state machine of layer-3 for EIB end devices is to map services as described in paragraph 2.2. The value of the routing counter is added when layer-4 applies a layer-3 request primitive. The routing counter value seven is reserved: a routing counter set to 7 is not changed by the Network Layer.

2.5 State Machine of Layer-3 for Bridges

Bridges and routers do also have a layer-3 but their layer-3 state machine differs from EIB end devices.

If an L_Data.ind with a routing_counter in [1...6] is received, the bridge decrements the routing_counter and transmits the service parameters of the L_Data.ind with the corresponding service parameters (source address, destination_address, DAF, class, l_sdu) of a L_Data.req to the other side.

If an L_Data.ind with a routing_counter value of seven is received, the bridge transmits the service parameters of the L_Data.ind with the corresponding service parameters of a L_Data.req to the other side.

Otherwise the layer-3 of the bridge discards the L_Data.ind.

2.6 State Machine of Layer-3 for Routers

If an L_Data.ind with DAF = ´1´ and routing_counter in [1...6] is received and the destination address is listed in the routing table, the router shall decrement the routing_counter and shall transmit the service parameters of the L_Data.ind with the corresponding service parameters of a L_Data.req to the other side.

If an L_Data.ind with DAF = ´1´ and the routing_counter equals seven is received the router shall transmit the service parameters of the L_Data.ind with the corresponding service parameters of a L_Data.req to the other side.

If an L_Data.ind with DAF = ´0´ and destination address equal to the physical address of the router is received, the router shall process the L_Data.ind identical to an EIB end device, see paragraph 2.2.1.

If an L_Data.ind with DAF = ´0´ is received and the destination address is listed in the routing table, the router shall transmit the service parameters of the L_Data.ind with the corresponding service parameters of an L_Data.req to the other side. Additionally, if the routing counter value was in [1..6] the router shall decrement it.

Otherwise layer-3 of the router shall discard the L_Data.ind.

An N_Data.req service invoked by the layer-3 user at the router shall be processed as described in paragraph 2.2.1.

More Detailed Routing Algorithm

For a router there are five possible courses of action in response to a received LSDU:

- ROUTE_UNMODIFIED: The LSDU is routed from the first line to another line without modification of the routing counter value. The LSDU is routed to the second line after an ACK character is sent back to the originator of the LSDU.
- ROUTE_DECREMENTED: The LSDU is routed from the first line to another line after the routing counter value is decremented. The LSDU is routed to the second line after an ACK character is sent back to the originator of the LSDU.
- FORWARD_LOCALLY: The LSDU is processed to an NSDU and given to the local network layer user after an ACK character is sent back to the originator of the LSDU.
- IGNORE_TOTALLY: The LSDU is ignored; no acknowledgment is sent back to the originator of the LSDU.
- IGNORE_ACKED: The LSDU is ignored, nonetheless an ACK character is sent back to the originator of the LSDU.

The following sub-clauses specify the routing algorithm for a router, which can either be a line coupler or a backbone coupler, depending on his position in the EIB twisted-pair topology.

Abbreviations:

C	routing counter value contained in the N protocol header
D	low order octet of the destination address, i.e. device address part
G	group address
SD	low nibble of high order octet plus low order octet, i.e. zone sub-address part
Z	high nibble of high order octet of the destination address, i.e. zone part
ZS	high order octet of the destination address, i.e. hierarchy information part

2.6.1 Routing in case of a Group Destination Address

if G found in routing table and $0_H < C < 7_H$ **then** ROUTE_DECREMENTED
if G found in routing table and $C = 0_H$ **then** IGNORE_ACKED[1]
elsif $C = 7_H$ **then** ROUTE_UNMODIFIED
else IGNORE_TOTALLY

[1] The ACK is sent by the Data Link Layer.

2.6.2 Routing in case of a Physical Destination Address: Line Coupler

2.6.2.1 Main Line to Subline Routing

if ZS = own subline address **then**
if D <> 00_H **then**
if C = 7_H **then** ROUTE_UNMODIFIED
elsif $0_H < C < 7_H$ **then** ROUTE_DECREMENTED
else IGNORE_ACKED
else FORWARD_LOCALLY
else IGNORE_TOTALLY

2.6.2.2 Subline to Main Line Routing

if ZS <> own subline address **then**
if C = 7_H **then** ROUTE_UNMODIFIED
elsif $0_H < C < 7_H$ **then** ROUTE_DECREMENTED
else IGNORE_ACKED
elseif D = 00_H **then** FORWARD_LOCALLY
else IGNORE_TOTALLY

2.6.3 Routing in case of a Physical Destination Address: Backbone Coupler

2.6.3.1 Backbone to Main Line Routing

if Z = own zone address **then**
if SD <> 00_H **then**
if C = 7_H **then** ROUTE_UNMODIFIED
elsif $0_H < C < 7_H$ **then** ROUTE_DECREMENTED
else IGNORE_ACKED
else FORWARD_LOCALLY
else IGNORE_TOTALLY

2.6.3.2 Main Line to Backbone Routing

if Z <> own zone address **then**
if C = 7_H **then** ROUTE_UNMODIFIED
elsif $0_H < C < 7_H$ **then** ROUTE_DECREMENTED
else IGNORE_ACKED
elseif SD = 00_H **then** FORWARD_LOCALLY
else IGNORE_TOTALLY