1 Communication Method of Operation

This chapter describes the **tasks and method of operation of the communication** and the principle of its function. This then should help, during configuring, to avoid configuring errors and to optimally utilize the link blocks (with regard to sequence and time response). This chapter also represents an **introduction to the world of terms** in the SIMADYN D communication.

The communication involves the areas:

**Links:** The system SIMADYN D can communicate via various links with other systems or devices. Whereby explanations are given as to how these links function in principle, which common features they share and what the user must be careful of during configuring.

**Network:** The network permits data transmission over several racks, without explicitly configuring the transmission paths over the racks by using function blocks. The principle method of operation is also described in this chapter.

**$ Signals:** The data transmission via $ signals is not part of the communication. Therefore the configuring of $ signals is also not described in this document.

1.1 Links Via External Communication Interfaces CS xy

The communication via external communication interfaces CS xy operates according to the following principle: P 32 / P 16 exchange data with a link board via the back plane bus. When serial links (i.e. SINEC H1, SINEC L2 DP etc.) are involved, then the data is “Restructured” and “Packed” by the link board firmware, such that it corresponds to the required telegram layout and protocol. No further preparation will take place for parallel links (i.e. coupling memory link):

The data to be transmitted is always transferred between a P 32/P 16 and a data interface on the link board.

Whether the data transfer between the P 32/P 16 and the link board occurs via the C or L bus, depends solely upon the link board bus connection (i.e. for CS7 via L bus or for CSH11 via C bus).
The performance of a link (i.e. how fast the data can be transferred to a data interface) has no bearing upon whether the data is transferred via the C or L bus, since both back planes have an identical structure and functionality.

The data interfaces are located on external link boards and not on the P32 local processor and can therefore be utilized by all the processors contained in a rack, both the P32 and P16. The sole prerequisite for the utilization of a data interface is that the processor module and the link board both contain the same bus connection.

The only exception for access possibilities to links is the PN local link on P32. This data interface is located on the RAM of the local P32 processor. Therefore it cannot be accessed by any other processors in this rack. It can only be utilized exclusively by the processor upon which it is configured.

1.1.1 General Method of Operation and Configuring

All link types respond in the same way, i.e. the method of link initialization from the P32/P16 and the way in which data is exchanged with the link partner, is independent of the link type. All link specific tasks, such as the set-up of a telegram or the node connect on a bus system is implemented only by the corresponding firmware on the link board and is totally unknown to a P32/P16. If a link board contains no firmware, such as the CS11, then this signifies that no link specific operations can be carried out. This only applies when the communications partner is also SIMADYN D (rack link and coupling memory link).

The basic initialization of a link board always occurs during the system start-up. Whereby a check is made as to whether the link board can be “Accessed”, the data interface is then formatted etc. These tasks are always instigated and implemented by the processor module, located to the left-hand side of the link board and contains the same bus connection as the link board.

No explicit configuring steps need to be carried out for the basic initialization of a link (Exception: Coupling Memory Link). The configuring of the desired link board in the Master program (arrangement drawing) is sufficient.

Utilization of the coupling memory boards MM11, MM21, MM3 and MM4 does not necessarily imply the utilization of the data interface (it is not mandatory to utilize this for communication purposes, but can also be used, for example, for $ signals. Therefore define at the connector CIP, on the MMx screen, whether the link is to be initialized ("Y" or "N"): The default setting for the connector is “N”, such that the data interface is only initialized when this connector is deliberately modified.

In general, the communication is configured at the function packet level; no definitions with regard to communication are made in the Master program. This gives the designer the advantage that configuring modifications with regard to communication can be implemented via the service utility and will become immediately active after resetting the rack (as long as the modifications have been saved in EEPROM). Therefore it becomes unnecessary to recompile all function packets and the Master program and to download them into a memory sub-module.

All function blocks, that access a data interface, possess a CTS connector for the identification of the allocated link board. This connector requires the definition of the configured board name and possibly the physical connector (only for CS7), upon which the desired data interface is located. If the name “D08H1”, for example, is defined for a CS11 link board, then the name “D08H1” must be specified at the CTS connector of all transmitters / receivers that wish to exchange data with this CS11 link board. Additionally specify the physical connector X01, X02 or X03 for a CS7 (CS7 can “Run” three links simultaneously).
The **name allocation for link boards** is not restricted within the SIMADYN D conventions. A restriction only exists when a rack is to be a component of a network (review the 3rd section in this chapter). It should therefore be clear at an early stage of the design, whether the configuring of a network is planned!

### 1.1.2 Link Central Blocks

The initialization, enabling and monitoring of a link is always implemented by a link central block. A central block exists for each link type (i.e. the central block @CSH11 for the link SINEC H1). The block names always begin with an ‘@’ and are always configured in the communications FP “Transmit”. Exactly one central block must be configured for each link; i.e. two central blocks @CS11 must be configured when two CS11 are utilized in a rack.

The link central blocks can be configured on all processors in a rack. Therefore it is possible, for example, to configure all link central blocks in a communications FP “Transmit” on a processor, independent of where the link boards are located. This type of centralization of all link central blocks simplifies, for example, the diagnostics.

All link central blocks have no transmit or receive functionality. They are given the task of copying the configured initialization information (configured on the INIT connectors) to the data interface and to determine whether the data interface is in an error free state. If this is the case and the firmware has completed all its initialization operations, then the link central block enables the link for all transmitters and receivers on the same rack. The data transmission can now begin. **The link enable is always only initiated, for timing reasons, after the start of the cyclic operation (after several processing cycles).**

The block outputs of the link central blocks contain information regarding the status of the link and possibly the status of the firmware.

All link central blocks must be configured in a sampling time of 32ms ± TA ± 256ms.

No data exchange will occur via a link for which no link central block has been configured. Therefore all the configured transmitters / receivers cannot access this data interface and must therefore wait for the central block enable. The wait state is indicated at the transmitter / receiver YTS connector outputs. **No error message will be generated!**

If a link central block detects an error during the initialization or the firmware shows no reaction or a faulty response or when the link central block is operating to an incorrect link type (i.e. the @CSH11 to a CS21 board), then it inserts an entry into the communication fault panel and no longer processes the link board.

### 1.1.3 Transmitter and Receiver

A transmitter / receiver implies function blocks that access a data interface (write and / or read). Therefore the message evaluation function block MSI (copies messages from the message buffer into a data interface) is just as much a transmitter as the process data transmit block CTV.

All links respond in the same manner to all transmitters / receivers; they do not differentiate between the individual links. Therefore it is not necessary to specify which type of link is been utilized at the transmitter / receiver block inputs.

All transmitters / receivers possess an address connector, in addition to a CTS connector. This connector is known as the AT (Address Transmit) for all transmitters and AR (Address Receive) for all receivers. This connector is known as US (user; only for P32) for all function blocks that simultaneously act as transmitter and receiver. This type of block is, for example, the service function block SER; it receives commands on the one hand and transmits an acknowledge on the other.
Transmitters / receivers also exist that possess several AT or AR connectors (CCC4 and CDC4). The number of AT, AR or US connectors “Tell” a function block how many data areas it needs to reserve in a data interface in order to be able to operate without error. A transmitter / receiver reserves a data area on the link board for each AT or AR connector. This is configured at the CTS connector. A function block will reserve two data areas for US connectors, one for a receive and one for transmit. The connect request and reservation of data areas is implemented sequentially, i.e. the individual transmitter / receivers can only register one after the other. There is no sequence defined, it is simply first come, first connected.

It is also possible, that a transmitter / receiver no longer finds sufficient memory area on the link board. When this occurs, this transmit / receiver disengages with an entry into the communication fault panel (i.e. the block no longer processes the data interface). This problem can only be resolved by utilizing a further link board and reconfiguring.

Before data can be exchanged between the transmitter and receiver, they must first “Detect” each other and “Synchronize” to one another. The detection is implemented via the configuring definitions at the CTS and AT, AR or US connector. The synchronization is only then possible when a transmitter identifies its partner as a receiver (or vice versa), when the length of a data area conforms, when the user data structure is compatible and when the transmission modes are identical (specification at the connector MOD or MOx for transmitters / receivers). If one of these check criteria does not conform, then the synchronizing transmitter / receiver disengages with a communication fault.

A further important communication feature for external communication interfaces is the restart capability of transmitters / receivers. This implies that transmitters / receivers will always find and synchronize themselves to their old data areas. The significance of this feature becomes obvious when we review the rack link. A maximum of 4 racks can be linked with the rack link. Whereby the racks can be disconnected and reconnected in random sequence (as often as necessary). This is only possible when the transmitters / receivers within the rack, in which each CS21 is installed, synchronize themselves to their “Old” data areas at each restart. If a new data area were to be reserved each time, then the data interface would soon be “Used Up” and a reset of the rack containing the CS11 would then be inevitable.

The restart of transmitters / receivers does however hide a danger, about which a clear understanding should be gained here:

During a system start-up (either the first or the xth), a transmitter / receiver cannot determine – when it identifies a “Suitable” data area during the connect request – whether this data area was “Once” used by itself or whether this data area perhaps is currently being utilized by a completely different transmitter / receiver (also transmitter or receiver). A transmitter / receiver always assumes that no other transmitter / receiver exists with the same configuring data (exceptions: the transmission “Select” permits any number of transmitters and “Multiple” permits any number of receivers; see chapter “Terms and Connectors”).

A simple criterion exists for the differentiation between transmitters / receivers, i.e. the configuring of a channel name at the AT, AR or US connector (see the chapter “Terms and Connectors”).

The channel name on a data interface must be uniquely defined by the designer for each transmitter / receiver. (Exceptions: for the transmission mode “Select” any number of transmitters are permitted and for “Multiple” any number of receivers are permitted; see chapter “Terms and Connectors”). Otherwise an uncoordinated multiple utilization of data areas and an eventual disabling of transmitters and receivers (communication faults) will occur.
1.1.4 Compatible User Data Structures

The user data structures contain the specifications for location and data formats of the corresponding user data as well as the number of utilized differing data formats. The structures can contain differing lengths, due to the differing data formats in a user data buffer.

In order to make the data exchange between the transmitters and receivers practical, it is necessary that the user data be interpreted in the same manner by both the transmitter and receiver. When, for example, a transmitter transmits two individual bytes, then the receiver should not process these bytes as a 16 bit data word. Therefore the user data structure is made known by the transmitters / receivers and checked by the link partner before the actual data transmission operation takes place.

Since the link partner is not necessarily a SIMADYN D function block (especially not when one thinks of the bus systems SINEC H1 or SINEC L2 FMS), the connector types in SIMADYN D (i.e. N2 or NF) may not be utilized as data types. Only standardized data types may be exclusively utilized, such as Integer 16. The following table shows which standardized data types can be utilized for the SIMADYN D connector types:

<table>
<thead>
<tr>
<th>Standardized Data Type</th>
<th>SIMADYN D Connector Type</th>
<th>Length in Bytes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Integer 16</td>
<td>I2</td>
<td>2</td>
</tr>
<tr>
<td>Integer 32</td>
<td>I4</td>
<td>4</td>
</tr>
<tr>
<td>Unsigned 8</td>
<td>B1, V1</td>
<td>1</td>
</tr>
<tr>
<td>Unsigned 16</td>
<td>N2, D2, O2, R2, T2, V2</td>
<td>2</td>
</tr>
<tr>
<td>Unsigned 32</td>
<td>N4, V4, O4, T4, R4</td>
<td>4</td>
</tr>
<tr>
<td>Unsigned 64</td>
<td>ND, TD</td>
<td>8</td>
</tr>
<tr>
<td>Floating Point</td>
<td>NF, TF</td>
<td>4</td>
</tr>
<tr>
<td>Octet String</td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>Time and Date</td>
<td></td>
<td>6</td>
</tr>
</tbody>
</table>

1. **Time and Date:**

```
<table>
<thead>
<tr>
<th>6 Octet</th>
<th>3 Octet</th>
<th>2 Octet</th>
<th>1 Octet</th>
</tr>
</thead>
<tbody>
<tr>
<td>0000 xxxx</td>
<td>xxxx xxxx</td>
<td>xxxx xxxx</td>
<td>dddd dddd</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Significat Value $2^{27}$</td>
<td>Significant Value $2^{15}$</td>
<td>Significant Value $2^{00}$</td>
<td>Significant Value $2^{00}$</td>
</tr>
</tbody>
</table>
```

**Value Range:** The Octets 6 to 3 specify the time ($00^{00}$ hours to $24^{00}$ hours, Granularity = 1ms, 0 ms £ x £ 86400000 ms, the sixth Octet has the first 4 bits unassigned). The Octets 1 and 2 deliver the date relative to 1.1.1984 (Granularity = 1 Day, 0 Days £ d £ 65535 Days).
As can be seen from the table, various SIMADYN D connector types can be transmitted between transmitters and receivers, without the occurrence of error messages (i.e. a transmitter transmits an N2 value and the receiver interprets the value as an O2 type).

**In addition, it is always permitted, to transmit Octet Strings to all other data formats!** This does not cause errors during the synchronization of transmitters and receivers (i.e. it is permissible that a transmitter transmit 2 Octets and the receiver receives 1 Integer 16).

### 1.1.5 Number of Link Boards in a Rack

The number of link boards, i.e. CS7, CSH11, CS11 and CS21 (coupling memory does not apply here, since only one coupling memory may be utilized per rack), are restricted by several system limits: rack size, configuration (pure P32, pure P16 or a mixed operation of P32 / P16) and the available address space.

The largest rack in the SIMADYN D system can accommodate 24 slots. At least one processor board must however be installed in a rack and this therefore leaves a theoretical 22 slots.

The number of installable CS boards is, with respect to this system limit, only fixed by the number of slots in a rack.

The two system limits “Configuration” and “Available Address Space” are directly related:

~ If only P16 (pure P16 rack) or P16 and P32 (mixed P32 / P16 rack), are installed in a rack, then only the special peripheral address space is available for the communication. The address space for the special peripherals consists of 3 x 64 K byte.

~ If only P32 (pure P32 rack), is installed in a rack, then an expanded address space is available, in addition to the special peripheral address space. The address space for special peripherals consists here also of 3x64 K byte. The address space for the expanded area consists of 2 M byte. Therefore a total of approximately 2.2 M byte is available for the CS peripherals.

The individual CS boards always utilize a constant address space; i.e. the CS7 always requires 64 K byte, independent of how many sub-modules (1–3) are configured. The location of the address area (special peripheral or expanded area) is dependent upon the board type and the configuration:

<table>
<thead>
<tr>
<th>Board Type</th>
<th>Pure P32 Operation</th>
<th>Pure P16 Operation, P32/P16 Mixed Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Special Peripheral</td>
<td>Expanded Area</td>
</tr>
<tr>
<td>CS11</td>
<td>16 K Byte</td>
<td>-</td>
</tr>
<tr>
<td>CS21</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>CS12 / 13 / 14</td>
<td>-</td>
<td>128 K Byte</td>
</tr>
<tr>
<td>CS22</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>CSH11</td>
<td>-</td>
<td>256 K Byte (Can utilize 64 k Byte)</td>
</tr>
<tr>
<td>CS61</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>CS7</td>
<td>-</td>
<td>64 K Byte</td>
</tr>
<tr>
<td>EP22</td>
<td>-</td>
<td>16 K Byte</td>
</tr>
<tr>
<td>EP3</td>
<td>-</td>
<td>16 K Byte</td>
</tr>
</tbody>
</table>
Examples: Rack with 24 Slots

1) For two P32 and one coupling memory: 19 slots are still available that can be utilized as follows:
   a) 12 CS11 and 2 CSH11
   b) 7 CSH11, 4 CS7 and 4 CS11
   c) 8 CSH11 and 11 CS11
   d) 12 CS11

2) For five P32 and one coupling memory: 13 slots (SPS) are available, that can be utilized as follows:
   a) 1 CS7 and 11 CS11
   b) 12 CS11 and 1 CSH11
   c) 7 CSH11 and 3 CS7

3) For four P32 and one P16 with a coupling memory (mixed operation): 13 slots (SPS) are available, that can be utilized by link modules with a total of 3x64 K bytes (192 KB).
   a) 1 CS7, 4 CS11 and 4 CSH11 (192 KB for 10 EP)
   b) 3 CS7 (192 KB for 6 EP)
   c) 12 CSH11 (192 KB for 12 EP)
   d) 1 CSH11, 1 CS7 and 7 CS11 (192 KB for 10 EP)

The CS7, CSH11, CS11, CS21, CS12/13/14 and CS22 can be accessed and processed by all processors in a rack, i.e. all transmit and receive blocks in a rack can address these boards. The CS61 can only be accessed by the P16.

1.1.6 Reorganization of a Data Interface

The possibility exists of deliberately reformatting a data interface without interrupting or interfering with the cyclic block operation.

This functionality is required to possibly find new “Space” on a data interface. Memory space allocation on a data interface without subsequent utilization of this area can, for example, occur during the configuring of a peer to peer network. A data path via several racks and link boards is possibly created between a transmitter and a receiver. The path is initially determined by the network and subsequently successively initialized. If a rack, upon which a sub-section of this path is to be created, is disconnected, then the network searches for an alternative path. The previously created path sub-sections are then possibly no longer required and remain as space blockers and unutilized on the data interface.

Therefore other transmitters / receivers may possibly then not have sufficient memory space available.

The functionality “Reorganization” therefore permits a data interface to be “Retro” formatted once again.

All link central blocks possess an CDV connector (B1 type). When a positive edge occurs at this connector, then the central block disables the link and formats the data interface after approximately 10 seconds. The data interface is subsequently re-enabled.
All transmitters / receivers and even firmware revert, during the disabling and reformatting, to a wait state. Channel reconnect and synchronization, just as with a system restart, occurs after the re-enable.

A further application case is the rack link. Data areas can also be reserved here, that are subsequently not utilized:

When, for example, individual racks (occurs only for racks containing the CS21) are disconnected and the number of transmitters or receivers are reduced by reconfiguring, then the “Previously” reserved data areas on the CS11 remain intact (this is the case for a CS21 - CS21 communication when the CS11 rack is only utilized as a data interface and not as link or communications partner).

These unutilized areas are eliminated by a CS11 reorganization.

### 1.2 Links Via the P16 Local Processor Interfaces

The 16 bit processors (P16) possess one or two configurable serial interfaces, as opposed to the 32 bit processors (P32), i.e. PM3 or PT31. In these cases, the utilization of an additional communications board CSxy is not necessary.

The protocols DUST1, DUST2, DUST3, USS and peer to peer can be run via the local processor P16 interface.

A link central block must be configured for each interface, as is the case for the communication via the external communication interfaces CSxy. The name always begins with an @..:

- DUST1: @DU1
- DUST2: @DU2
- DUST3: @DU3
- USS: @USS
- Peer to Peer @PTP

A processor board with configurable serial interface requires an interface module to be configured as a further hardware component for each utilized interface. A user name can be specified at a physical connector (X01/X02) for the identification of a link and for documentation purposes. This name will however not be checked. Whether actual communication takes place via this connector is exclusively specified in the FP configuring.

#### Procedures for the Interface Definitions (Example PM16):

- Selection of the basic function “Set Module Parameters” for the selected module parameter drawing.
- Selection of the desired module parameters in the board parameter element of the module.

<table>
<thead>
<tr>
<th>D01_P1</th>
<th>PM16</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSM</td>
<td>0</td>
</tr>
<tr>
<td>ISE</td>
<td>N</td>
</tr>
<tr>
<td>X01</td>
<td>DUST1</td>
</tr>
<tr>
<td>X02</td>
<td>0</td>
</tr>
<tr>
<td>X5C</td>
<td>*LUEB</td>
</tr>
<tr>
<td>X5D</td>
<td>&lt;</td>
</tr>
<tr>
<td>X5A</td>
<td>&gt;</td>
</tr>
<tr>
<td>X5B</td>
<td>&gt;</td>
</tr>
</tbody>
</table>

- Pop up the module parameter form window for entering and for modifying the parameters and parameter comments.
- Enter, modify and delete the module parameter in the form window entry panels.

- Enter, modify or delete the single line parameter comment with a maximum 37 characters in the input panel.

- The entries are checked for syntax when exiting the form window (upper case letters, digits, underscore), transferred to the data memory and graphically displayed when the entries are correct.

Further specifications are required on a P16, as a special feature of the indirect communication, whereby the tasks for linking to the data transmission and the tasks for the data acquisition as well as distribution to the various block types are divided up. A P16 possesses the connectors CCT, CCR and COP for this purpose. A cross reference between a so called telegram block in a special communications FP transmit / receive and at least one data block (indirect transmitter / receiver) in a permanent FP is created via these connectors. This cross reference consists of the telegram name and the sampling time of the corresponding “Telegram Block” in the special communications FP.

**Procedures for the Telegram Configuring:**

- Selection of the basic function “Set Module Parameters” for the selected module parameter drawing.

- Selection of the desired telegram types CCT, CCR or COP,

- Pop up the form window for expanding, modifying or deleting indirect telegram names with a maximum of six characters and the sampling time specification.
Example for the Form Window:

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>CCT</td>
<td>CCR</td>
<td>COP</td>
</tr>
</tbody>
</table>

Selection Panels for the Indirect Telegrams

<table>
<thead>
<tr>
<th>End</th>
<th>Cancel</th>
<th>-D01_P1</th>
<th>CCT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td>?</td>
</tr>
<tr>
<td>CTVAX</td>
<td>T3 &amp;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CTVAX</td>
<td>T2 &amp;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>CTVAX</td>
<td>T2 &amp;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>......</td>
<td>T2 &amp;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>T2 &amp;</td>
<td>T2 &amp;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>T2 &amp;</td>
<td>T2 &amp;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>T2 &amp;</td>
<td>T2 &amp;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>T2 &amp;</td>
<td>T2 &amp;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>T2 &amp;</td>
<td>T2 &amp;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>T2 &amp;</td>
<td>T2 &amp;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>T2 &amp;</td>
<td>T2 &amp;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>T2 &amp;</td>
<td>T2 &amp;</td>
<td></td>
<td></td>
</tr>
<tr>
<td>T2 &amp;</td>
<td>T2 &amp;</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

1.3 Network

Large equipment networks structure the tasks to be fulfilled along strict technological lines and divide them, due to their complexity, between several racks. Whereby it must be guaranteed that link partners, which are not directly coupled to one another, can also communicate.

A further requirement is the diagnostic possibilities from a central control booth or station for all racks of a network (the term “Island” is frequently used in this context).

Configure the component network for these types of transport tasks. The network only provides access to SIMADYN D racks that are coupled together either via the bus system SINEC H1 or SINEC L2 FMS or the rack link. The network cannot be utilized with SINEC L2 DP!!

The network is a pure P32 functionality. P16 can only be utilized as an end user and only when it is utilized in the mixed operation with P32 in one rack (this is described in more detail later on in this chapter).
1.3.1 Network in P32

**Example:** Central diagnostics and communication from indirectly linked racks in a network consisting of 6 racks. The network task could be, for example, transport service requests from the central control booth to rack 6 and rack 5 and to transport the corresponding replies (from the function block SER) in the opposite direction. A further task could be to support the data exchange between rack 2 and rack 4.
The functionality of the network can be divided into three areas:

~ **Routing Network:** This component permits the routing dialog between a diagnostic or service station and all P32 function blocks, that possess the US connector. These are currently the P32 functions blocks **SER (Service)** and **@TCI (System Trace)**. The transport tasks of the network are therefore,

~ Determine whether the desired rack and the board exist (i.e. were configured and are in operation),
~ Transport the data packet (request) over several racks to the destination interface,
~ Transfer the data packet to the desired function block (and remember the transmitter),
~ "Send" the reply back to the transmitter.

The network searches for the shortest path (when several are available) and is capable of correctly returning the replies, for several similar requests to the same function block, correctly to the transmitters. No restriction applies to the amount of requests that are "Running" simultaneously over the network, the network automatically decides when it no longer can accept further transport requests (since, for example, all data paths are "j ammed").

If several requests from a specific source are “Currently Running”, then the network cannot guarantee that the replies will arrive in the same sequence as the requests were sent. The reasoning behind this, is that request paths have different lengths and that other paths will be selected, when “jamming” occurs (when possible), via which the data transport will then run faster. This is also possible for requests that require a cyclic reply. **The source must evaluate the replies accurately and cannot expect that they will automatically arrive in the correct sequence.**

~ **Peer to Peer Network:** This functionality is based upon a data path between a transmitter and a receiver or firmware (on a link board). As opposed to the routing network, the path, once created, is not used just once but remains intact and reserved for both nodes for which it was created. Therefore the destination node will not need to be searched for each time. This component is available for all P32 function blocks, that possess an AT, AR or US connector. The location of the destination node data interface is specified at this connector (on which rack and which link board).

~ **Network Status:** Indicates the network status, i.e. the network status shows which rack and interface can be accessed. The status is updated cyclically.

The individual network functionalities are configured with the following function blocks:

~ **@NMC:** Responsible for all logistical tasks of the routing and peer to peer network. It must be configured for both network components. It is configured once per rack. The block automatically determines whether it implements tasks for the peer to peer or routing network or both network components.

~ **NRI:** This is the interface block for the routing network. The block must be configured for each SER or @TCI block, that is to be accessed via the routing network. The block must be configured on the same processor that contains the function block @NMC.
NTC and NTD: Both blocks must be configured for the peer to peer network. The function block NTC implements the search and initialization of the data path for a connection, the NTD implements the copying of data packets between two data interfaces. If several function blocks NTD have been configured, then the function block NTC defines which data interfaces are to be processed by which NTD’s. This definition occurs only once and no longer switches in cyclic operation. The performance of the peer to peer network can be improved when several NTD function blocks are configured.

NSI and NSL: Both blocks are necessary for the network status. A function block NSI should be configured for each rack. The NSL function blocks must be implemented as interface blocks on the bus systems (i.e. SINEC H1).

The exact application and the function scope of the blocks can be reviewed in the P32 function block descriptions and in the chapter “Network” in this manual.

1.3.2 P16 as an End Node in the Network

The network functionality is not available for the P16. The data packet distribution can only be expedited by the P32. The P16 can, however be utilized as an end node in the P32 / P16 mixed operation:

~ **Routing Network:** The function block @MO2 can be utilized (has an AT and an AR connector instead of the US connector). The function block @MO2 is the equivalent to the function block SER on P32.

~ **Peer to Peer Network:** All function blocks with an AT or AR connector can be utilized (i.e. CTD801).