

CMX V6.0/V5.1 (Solaris)

Communications Manager UNIX
TCP/IP via WAN/ISDN

Comments... Suggestions... Corrections...

The User Documentation Department would like to know your opinion of this manual. Your feedback helps us optimize our documentation to suit your individual needs.

Fax forms for sending us your comments are included in the back of the manual.

There you will also find the addresses of the relevant User Documentation Department.

Certified documentation according DIN EN ISO 9001:2000

To ensure a consistently high quality standard and user-friendliness, this documentation was created to meet the regulations of a quality management system which complies with the requirements of the standard DIN EN ISO 9001:2000.

cognitas. Gesellschaft für Technik-Dokumentation mbH
www.cognitas.de

Copyright and Trademarks

Copyright © 2001 Fujitsu Siemens Computers GmbH.

All rights reserved.

Delivery subject to availability; right of technical modifications reserved.

All hardware and software names used are trademarks of their respective manufacturers.

This manual is printed on
paper treated with
chlorine-free bleach.

Preface

CS-ROUTE functions

Startup

Configuring

Overview of administration and diagnosis

CS-ROUTE commands

FSS commands

Examples

What to do if...

Reference section and index

Contents

1	Preface	1
1.1	Brief description of CS-ROUTE	1
1.2	Target group	2
1.3	Summary of contents	2
1.4	Notational conventions	3
1.5	README files and man files	6
2	CS-ROUTE functions	7
2.1	Network access	9
2.2	Linking LANs via a WAN	9
2.3	WAN access	12
2.3.1	Direct data connection	12
2.3.2	Analog telephone network	12
2.3.3	X.25 network	14
2.3.4	ISDN network	14
2.3.5	Frame relay network	17
2.3.6	Line costs for WAN dial-up connections	17
2.4	Alternate routing	19
2.5	Point-to-point protocol	20
2.6	Access protection for a subnetwork connection	21
2.7	Access protection at user level	22
2.8	Van-Jacobson header compression	22
2.9	Message filters	23
2.10	Supported RFCs	23
3	Startup	25
3.1	Installing/Uninstalling	25
3.2	Operating requirements	25
4	Configuring	27
4.1	Local system	27
4.1.1	IP interface	28
4.1.2	Subnetwork interface	32
4.1.3	Message filters	32
4.2	Partner systems and their reachability	33
4.2.1	Facilities	34
4.2.1.1	Point-to-point protocol	35
4.2.1.2	Access protection for a subnetwork connection	36
4.2.1.3	Access protection at user level	37
4.2.1.4	Van-Jacobson header compression	37

Contents

4.2.2	Subnetwork routes	38
4.2.2.1	Alternate routing	38
4.2.2.2	Minimum integration	39
4.2.2.3	Maximum integration	40
4.2.2.4	X.32 dialing	41
4.2.3	Information on the partner system	42
4.2.4	Configuring IP routes	43
5	Overview of administration and diagnosis	47
6	CS-ROUTE commands	51
6.1	CS-ROUTE	53
6.1.1	Starting/stopping CS-ROUTE	53
6.1.2	Autostart function of CS-ROUTE autostart function	53
6.2	Object class clw	54
6.3	Object class if	56
6.4	Object class net	62
6.5	Object class fi	67
6.6	Object class lsn	73
6.7	Object class conn	74
6.8	Diagnostic commands	78
6.8.1	ADS trace	78
6.8.2	CLW trace	79
6.8.3	PPP trace	79
6.8.4	CS-ROUTE traces	79
6.8.4.1	csrtron	80
6.8.4.2	csrtroff	81
7	FSS commands	83
7.1	Object class FACIL	88
7.2	Object class NSAP	93
7.3	Object class SNPAROUTES	97
7.4	SUBNET object class	104
7.5	Object class PPPAUTH	106
8	Examples	109
8.1	TCP/IP via ISDN	110
8.1.1	Configuring an ISDN permanent connection	113
8.1.2	Van-Jacobson header compression	113
8.1.3	Access protection at user level	114
8.1.4	Configuration for PBXs	117
8.1.5	Configuration for asynchronous PPP	119
8.1.6	Configuring PPP in X.25	120

8.2	TCP/IP via X.25	121
8.2.1	Configuring an X.25 PVC	123
8.3	TCP/IP via frame relay	124
8.4	TCP/IP via a direct data connection	126
8.5	Alternate routing	128
8.6	X.31 minimum integration	130
8.7	X.31 maximum integration	132
8.8	X.32 dialing	134
8.9	Setting up message filters	137
9	What to do if...	139
9.1	CS-ROUTE is not started automatically after a system start . . .	139
9.2	The IP interface cannot be configured/started	140
9.3	The subnetwork interface cannot be activated	141
9.4	The WAN partner does not respond to the <i>ping</i> command . . .	142
9.5	The WAN partner can be reached actively but no passive connection can be made	146
9.6	The local system does not respond to the <i>ping</i> command from the WAN partner	147
	Glossary	149
	Abbreviations	157
	Related publications	161
	Index	163

1 Preface

1.1 Brief description of CS-ROUTE

When used with the the product CMX V5.1, CS-ROUTE permits connectionless IP traffic over a connection-oriented wide area network (WAN).

CS-ROUTE is able to use the following WANs: leased lines, telephone networks, ISDN, X25 and Frame Relay networks.

You can use CS-ROUTE to:


- run IP applications, which were previously restricted to LANs, over a WAN, transparent to the user.
- connect isolated LAN islands.

IP packets are exchanged between decentralized LANs across the WAN.

CS-ROUTE provides you with the functions required for these tasks, for example alternate routing if WAN partners can be reached via several routes and the point-to-point protocol for communication in multivendor networks.

CS-ROUTE also provides you with different levels of access protection, which check whether connections may be set up, together with a header compression procedure which reduces costs.

A detailed description of the CS-ROUTE functions can be found in chapter “CS-ROUTE functions” on page 7.

 The term WAN is used in this manual for all Wide Area Networks, including ISDN. By contrast, the other manuals for CMX/CCP V5.1 restrict the term WAN to Wide Area Networks without ISDN.

1.2 Target group

This manual is intended for network and system administrators.

Basic knowledge of the SOLARIS operating system and data communications is required.

1.3 Summary of contents

The following brief description of the contents should give you a general overview of the CS-ROUTE manual.

The preface contains a brief description of the product CS-ROUTE. It also covers general topics such as notational conventions and provides information on using README and manual supplement files.

Chapter 2 covers the functionality of CS-ROUTE in detail.

Chapters 3 through 5 contain information on operating CS-ROUTE. Chapter 3 describes how to activate CS-ROUTE, chapter 4 configuring CS-ROUTE and chapter 5 covers administration and diagnosis.

Chapter 6 contains the CS-ROUTE commands, and chapter 7 the FSS commands.

Chapter 8 provides you with configuration examples for various application scenarios.

Chapter 9 describes potential causes of problems with CS-ROUTE and their solutions.

The manual closes with a glossary, followed by lists of abbreviations and related publications and an index.

1.4 Notational conventions

All the command descriptions have (as far as possible) the same format:

- Command description
- Syntax
- Syntax description
- Exit status
- Error messages
- Files
- Example
- See also

The elements listed here are described below.

Command description

The first section of each command description contains the following:

- Information on how to use the command
- The different tasks of the various command formats, where a number of formats are available
- The environment in which the command is to be used (e.g. entries in files, access rights)
- Background information

Syntax

cmd_[**-a**]_[**-b**]_[**-c**]_[**-d**arg1]_[**-f**arg2] file...

You must enter *cmd* and specify one or more files for *file*, separating each entry with a blank. You can also specify:

- One or more of the options *-a*, *-b*, *-c*. These options can be specified individually (**-a**₋**-b**₋**-c**) or together (**-abc**).
- The *-d* option, where *arg1* must be replaced with an argument. No blanks may be entered between the option and the argument.
- The *-f* option, where *arg2* must be replaced with an argument.

The metasyntax used has the following meaning:

Bold characters

Constants. Bold characters must be entered exactly as they are displayed.

Normal characters

Variables. These characters represent other characters that you select and enter.

[] Options. Arguments enclosed in square brackets are optional and need not be specified. The square brackets themselves are not to be input unless specifically requested.

_ Mandatory blanks.

... The previous expression can be repeated. If blanks, which are not part of the expression, have to be entered between the repetitions, a _ (blank) is placed before the ... characters.

| Selection options. You must select exactly one of the expressions separated by the vertical line.

{ } Curly brackets enclose selection options and the curly brackets themselves are not to be input.

underscoring

Default

Syntax description

Here you will find a description of the options and arguments (input files, parameters, variables, etc.) which can be entered when issuing a command. No distinction is made between constants and variables in continuous text. All syntax elements, file names, path names and commands are represented in *italics* in the continuous text.

Exit status

An exit status is the value returned to the calling process by a command that has been executed. This value provides information on the execution of the command. The exit status is a numeric value and is stored in the ? variables. You can query the exit status by specifying the *echo \$?* command.

The exit status is only described if it does not comply with the following rule:

- 0 after the command has been executed correctly
- ≠0 if errors occur

Errors

Important error messages are indicated and explained, and information is given on how to avoid and eliminate errors.

Error messages are usually output to *stderr*; the standard error output. the screen normally acts as the standard error output.

Files

The files accessed or created by the command in question are specified here.

Example

Examples are used to explain the main function of the command, the use of basic options, and useful combinations of options and arguments. System input is represented in application examples in constant-width semi-bold font. All of these input lines are terminated with the ENTER key. This key is therefore not used at the end of lines.

System output is represented in constant-width font, except for in continuous text, where it is in *italics*.

See also

Here you will find reference to other commands with similar functionality, or which work together with the command in question. References to other publications which deal with this command are also provided.

Notes and warnings



This symbol refers to vital information which must be observed.



Caution!

This symbol refers to dangers which may lead to loss of data or damage to the device.

1.5 README files and man files

Information on any functional changes and additions to the current product version described in this manual can be found in product-specific Readme files. These can be found in the readme package that is shipped with the product.

For CMX, there are also online manual pages that can be accessed once the product has been installed.

2 CS-ROUTE functions

CS-ROUTE provides you with a routing service on your Solaris server which allows you to access various subnets and connect LANs via WAN (LAN-WAN routing).

CS-ROUTE allows communication via:

- Direct data connection
- Analog telephone network
- X.25 network
- ISDN network
- Frame relay network

CS-ROUTE supports routing of:

- TCP/IP

Please note that the term “subnetwork” has a different meaning in connection with WANs than it does in connection with IP (see section “Configuring IP routes” on page 43).

CS-ROUTE provides one or more IP interfaces and subnetwork interfaces. As shown in figure “Structure and embedding CS-ROUTE” on page 8, CS-ROUTE comprises the following components: the CLW driver (Connectionless WAN Access), which provides the network access functions and the LAN-WAN routing for TCP/IP, the ADS components and the PPP driver.

As soon as data is available to be sent at the higher-level IP layer, one or more subnetwork connections to the corresponding WAN partner system are set up (depending on the configuration). To reduce costs and share resources, the subnetwork connections are time monitored. If no data is sent or received during a (configurable) period of time, the subnetwork connection is shut down. As soon as data for output to the WAN partner system exists, the connection is set up again. The interim shutting down of a connection and the new setting up of the subnetwork connection is transparent to the application. See section “Line costs for WAN dial-up connections” on page 17.

CS-ROUTE functions

The PPP driver

The point-to-point protocol (PPP) provides a standard method for the secure transfer of TCP/IP packets over WAN (ISDN, X.25, frame relay). PPP is currently implemented over X.25, ISDN, leased lines (X.21/V.24) and Frame Relay.

In particular, PPP provides the following functionality:

- With PPP as the standard in the world of routers, interoperability with non-SNI routers is a given.
- With its authentication procedures (CHAP and PAP), PPP provides access protection at user level (see also the section “Access protection at user level” on page 22).

The ADS component provides the CLW and PPP drivers with a uniform access interface to configuration data. The locality of this data remains transparent to the CLW and PPP drivers.

The structure of CS-ROUTE and its embedding in the complete system is shown in simplified form in the following diagram:

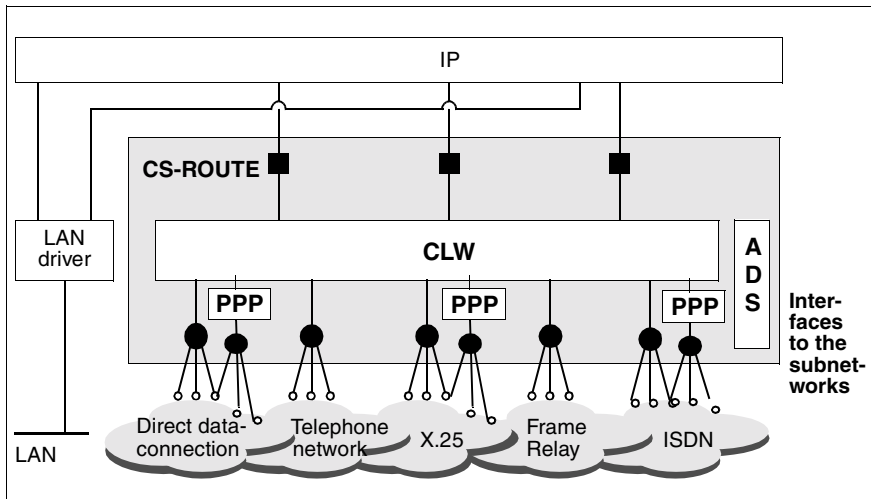


Figure 1: Structure and embedding CS-ROUTE

CS-ROUTE is conceived for media without broadcast capabilities, i.e. relieves the network of broadcast messages.

This chapter illustrates the functions of CS-ROUTE on the basis of two typical applications. This is followed by descriptions of other CS-ROUTE functions that provide useful support for routing.

2.1 Network access

The Solaris server is connected directly to a WAN. There are two ways to connect clients: they are either connected directly to the WAN with appropriate network access software or they communicate with the Solaris server via a router.

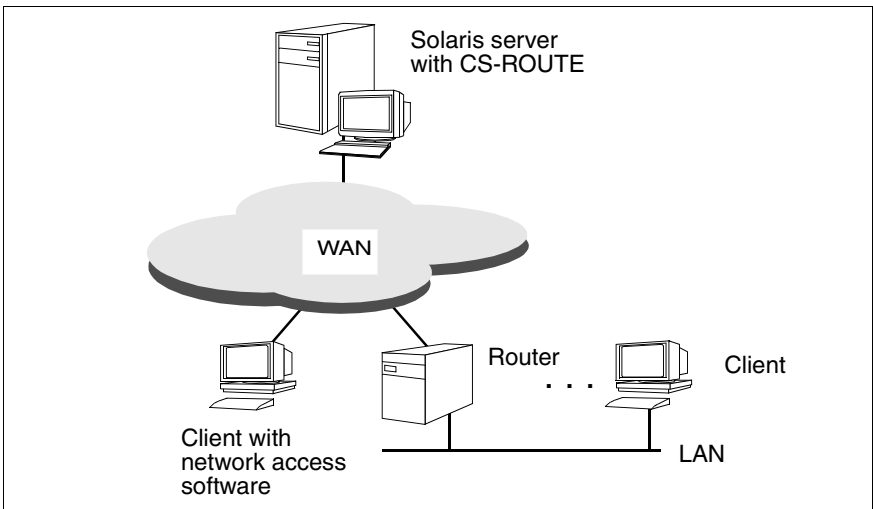


Figure 2: Network access

2.2 Linking LANs via a WAN

CS-ROUTE supports LAN-WAN routing and thus enables independent LANs to be linked via a WAN. This allows complex router networks to be set up across LANs and WANs.

All clients and Solaris servers or routers can communicate with each other.

CS-ROUTE sets up the necessary WAN connection and handles the required packet fragmentation and reassembly of the TCP/IP LAN packets for transport across the WAN.

The routing itself is dynamic. The routing daemons for handling routing protocols and information communicate via CS-ROUTE.

Linking LANs via one WAN

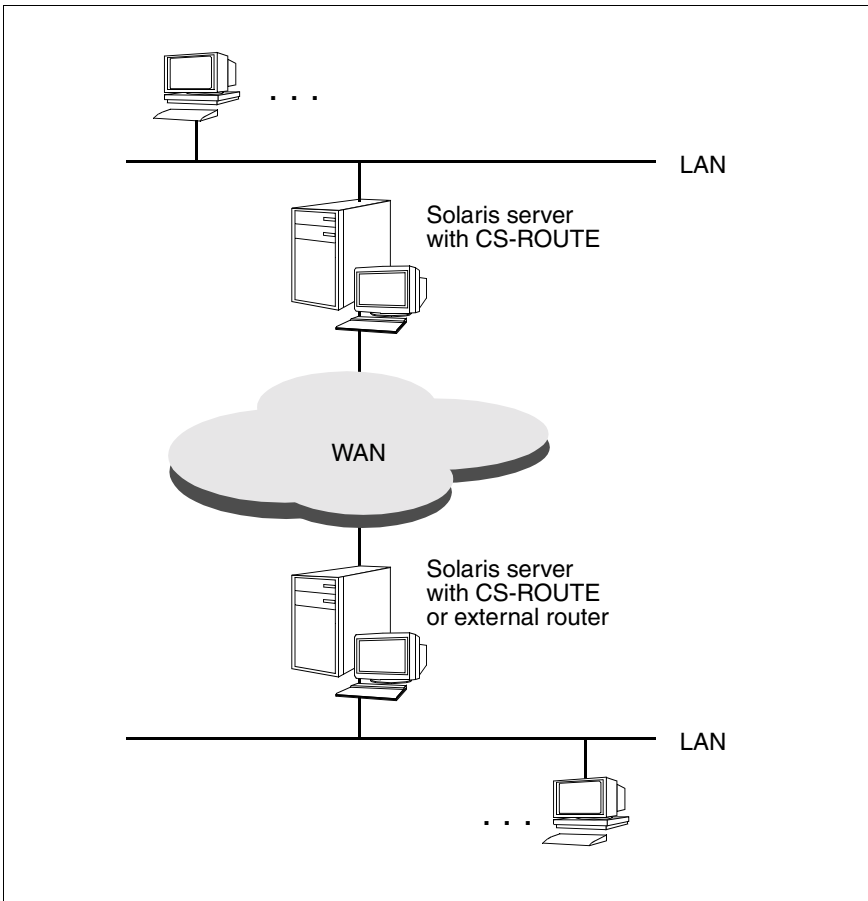


Figure 3: Linking LANs via one WAN

Linking LANs via multiple WANs

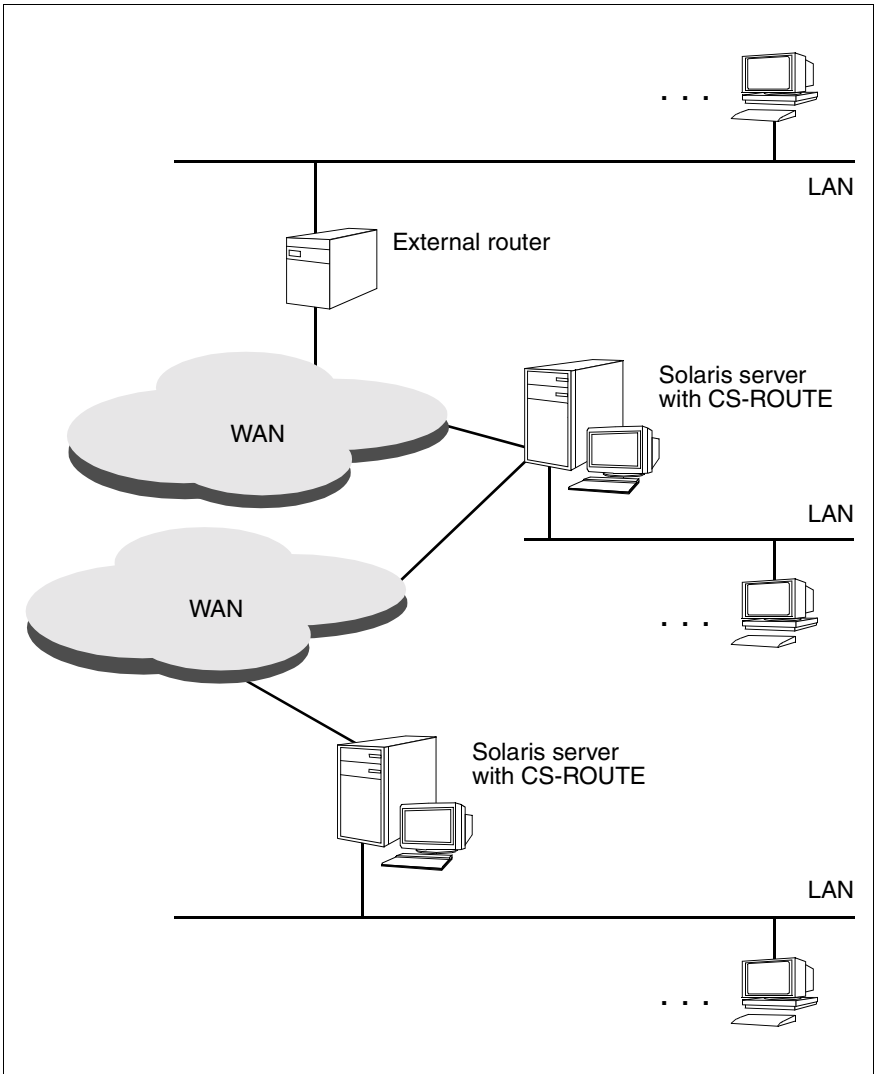


Figure 4: Linking LANs via multiple WANs

2.3 WAN access

CS-ROUTE supports access to direct data connections and to analog telephone, X.25, ISDN and frame relay networks.

2.3.1 Direct data connection

Two systems (hosts or routers) are connected via a dedicated line (point-to-point connection).

The line uses only one network protocol (IP), depending on the configuration. Reconfiguration is required if the line is to be used differently.

2.3.2 Analog telephone network

You can reach partners in the X.25 network via two-step dialing over the analog telephone network. The Solaris system with CS-ROUTE is connected to an X.25 network as a packet oriented DTE (Data Terminal Equipment) via the analog telephone network. This conforms to X.32 and is also called X.32 dialing (see also the manual “CMX/CCP, WAN Communication” [3]).

The following are required for connecting the Solaris system to an X.25 network via X.32 dialing:

- Access to an analog telephone network.
- The X.25 network provider must support X.32 dialing.

With X.32 dialing, the connection is set up to the partner system in two steps:

1. Your Solaris system initially sets up a dial-up connection to the interconnection system, or vice versa.

After dialing, the Solaris system is connected to the X.25 network as a packet-oriented DTE.

2. As soon as the connection exists, the Solaris system sets up an X.25 connection to the partner system, or vice versa. The X.25 connection is a switched virtual call (SVC).

Once the connection is established, additional SVCs can be set up from the Solaris system to any other DTE or from any other DTE to the Solaris system.

The connection is shut down as soon as the last SVC is shut down. Shutdown is carried out by the partner who set up the original connection.

Only switched virtual calls (SVCs) can be operated via X.32 dialing.

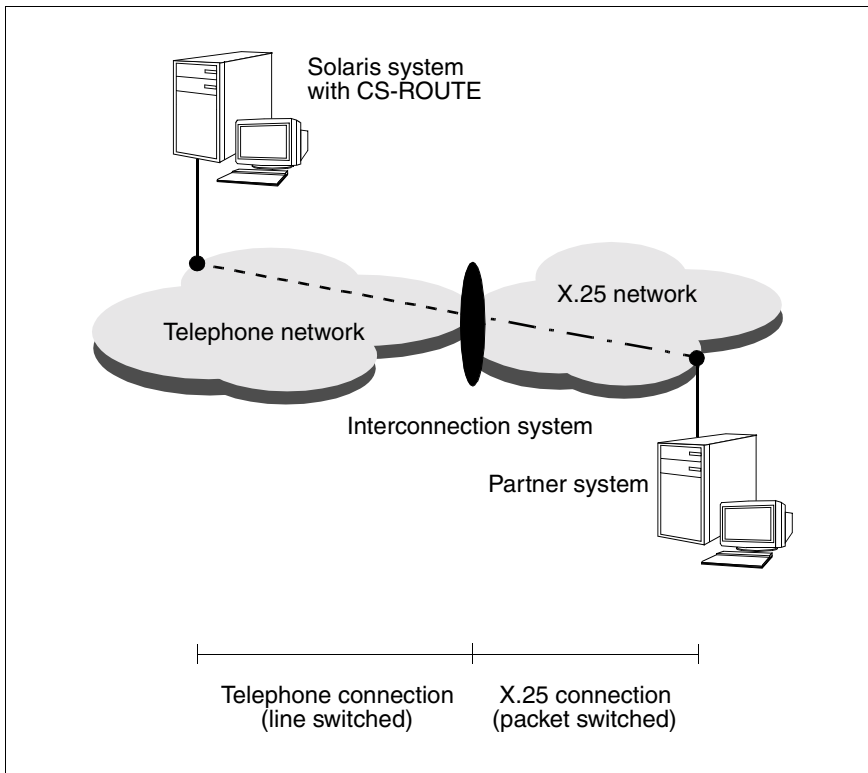


Figure 5: X.25 communication via an analog telephone network: X.32 dialing

2.3.3 X.25 network

Switched Virtual Call (SVC) and Permanent Virtual Call (PVC) are supported.

The point-to-point protocol can be used for communication with non-SNI routers.

2.3.4 ISDN network

Dial-up connections and permanent connections are supported. The point-to-point protocol (PPP) can be used with external routers.

If direct access to the X.25 network is either not possible or is too complex or expensive, two additional options exist for reaching partners in the X.25 network: either two-step dialing (X.31 case A: minimum integration) or direct transfer of X.25 data via the ISDN connection (X.31 case B: maximum integration).

Minimum integration

The Solaris system with CS-ROUTE can be connected as a packet oriented DTE via ISDN to an X.25 network. This complies with X.31 case A and is also called minimum integration (see also the manual “CMX/CCP, ISDN Communication” [4]).

The connection is set up in two steps to the partner system with minimum integration:

1. Your Solaris system initially sets up an ISDN connection to an access unit, or vice versa.
2. As soon as the ISDN connection exists, one or more X.25 connections are set up via the access unit, transparent to the ISDN network.

The access unit provides the transition from the ISDN network to a packet switching network and is, for example, a packet switching unit with an ISDN input and X.25 output.

You must obtain the ISDN number of the access unit from your X.25 network provider for communicating with partners on the X.25 network.

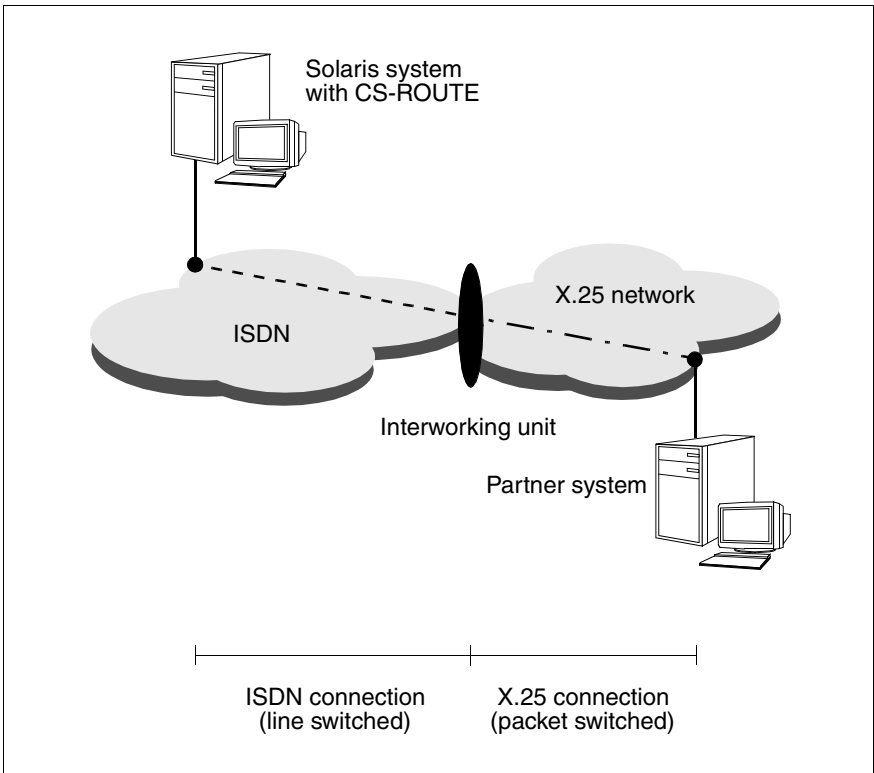


Figure 6: X.25 communication via ISDN: minimum integration

Maximum integration

If your Solaris system with CS-ROUTE has a Euro-ISDN access, you can use maximum integration (see also the manual “CMX/CCP, ISDN Communication” [4]). Maximum integration means that the ISDN network provides the user with packet switched services (according to X.25) on request. X.25 data is transferred directly over the ISDN connection.

The X.25 functionality is provided in the D channel or B channel, depending on the network, and must be ordered from the network provider for the ISDN access. The X.25 functionality is only provided in the B channel for an S_{2m} access.

With X.25 access via a B channel, a dial-up connection is initially set up to a packet handler on request. The ISDN exchange is thereby only informed of the requirement for X.25 functionality when the connection is set up. After the ISDN connection has been set up, it can be used as a normal X.25 main access.

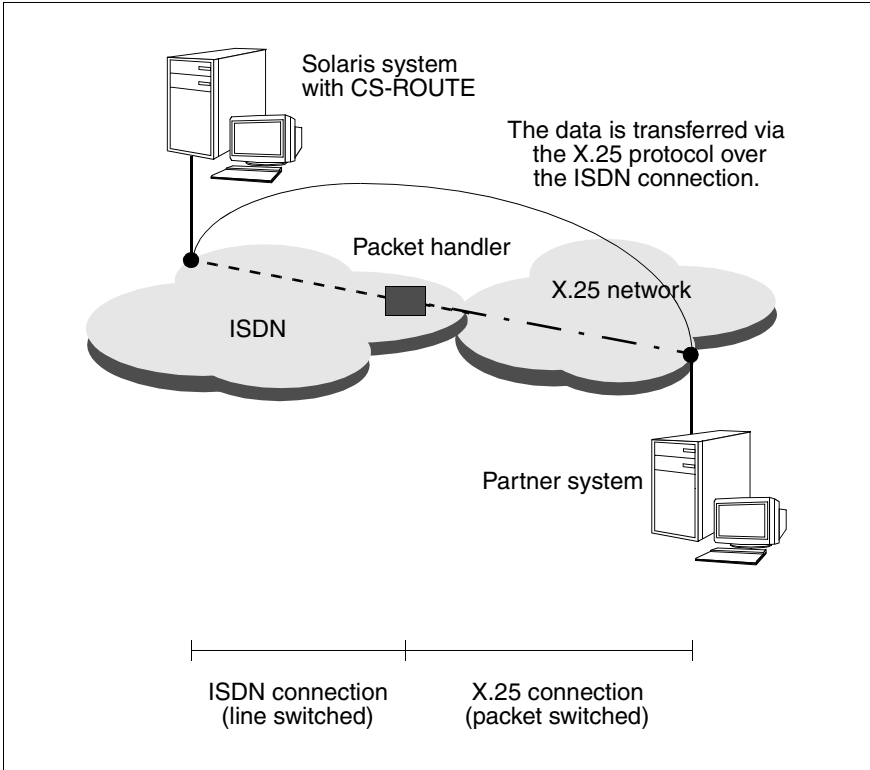


Figure 7: X.25 communication via ISDN: maximum integration

Reaching GSM subscribers

CS-ROUTE provides an asynchronous transmission procedure (asynchronous PPP) for setting up connections to mobile telephone subscribers who dial in from the D1/D2 network via ISDN.

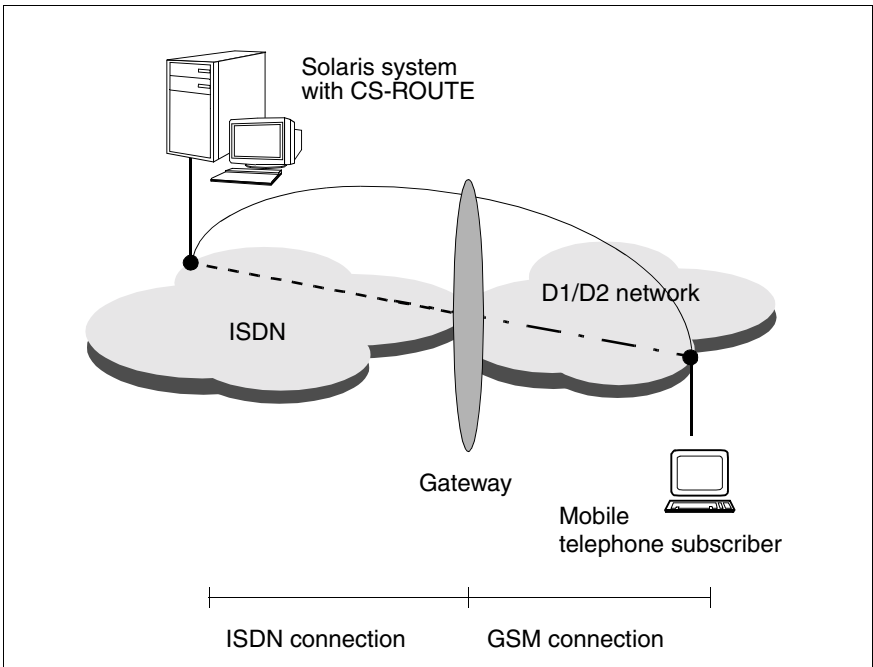


Figure 8: Dialing into ISDN from a mobile connection (GSM)

2.3.5 Frame relay network

Permanent Virtual Circuits (PVCs) are supported.

2.3.6 Line costs for WAN dial-up connections

Many transport system applications (add-on products), especially in the TCP/IP world, have monitoring algorithms which control connections or partner systems cyclically (polling) or distribute information (e.g. routing messages). For the most part, this is done automatically without the user being aware of this data transfer; in some cases, an appropriate configuration is required.

This kind of data transfer has no impact where LANs are concerned because of the small volume of data involved. If, however, individual systems or system groups (LAN islands) are connected via WAN (e.g. X.25 or ISDN), the polling will incur line costs.

In the case of packet-switched networks like X.25, these short messages have hardly any impact, since charges are volume oriented (note: some X.25 network operators also charge SVC connection setup charges).

In the case of circuit-switched networks (ISDN, telephony) on the other hand, charges are calculated according to the number and duration of connections and if the add-on products are configured inappropriately, considerable costs may be incurred. See also section "Object class net" on page 62.

For many applications, unintentional polling can be deactivated or reduced to a great extent:

openFT:

Polling (5 min.) if jobs in the address book cannot be processed. For example, if no FT application has been started on the partner system or the partner reports *temporary errors* due to memory bottlenecks, the job can be deleted. You will find more information in the "*openFT* for UNIX" [5] user guide under the index entry "deleting, job".

AS/X:

sends „keep alive“ messages if no data is being transferred via existing connections. For more information, see the „Advanced Server UNIX Overview and Installation“ manual [6], under "Planning for WINS Server Replication across WANs"; various parameter combinations are possible.

SINIX TE:

via TCP/IP for DPTG2 5 seconds polling: can be deactivated via RTTY ;
for V2.1 via modem (ISDN card as modem): channel operation can be set (one channel)

2.4 Alternate routing

The alternate routing function optimizes the connection options of your Solaris system with those of partner systems in the WAN. If alternative routes to a partner system are configured, the required route is selected as needed. If the number is occupied or a line fails, the desired connection is set up via an alternative route in either the same subnetwork or a different one.

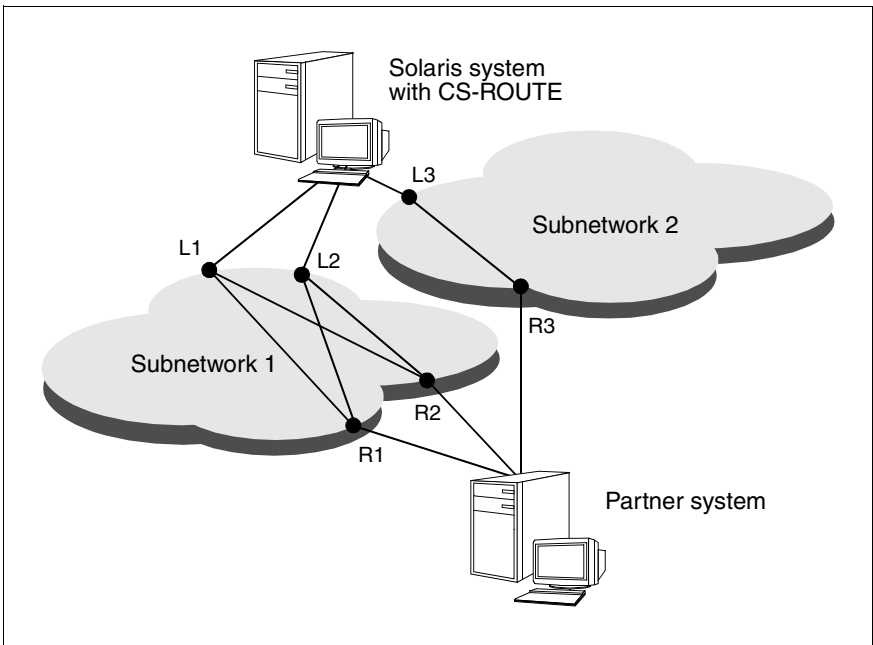


Figure 9: Alternative routing

If it is not possible to set up the connection via the local access L1, route L2 - R1, L2 - R2 or L3 - R3 is selected. If the partner system cannot be reached via access R2, the connection is set up via route L1 - R1, L2 - R1 or L3 - R3.

2.5 Point-to-point protocol

CS-ROUTE supports the point-to-point protocol (PPP) for ISDN, X.25, dedicated lines (X.21/V.24) and Frame Relay.

Many routers require the point-to-point protocol (PPP) for communicating with other routers. PPP increases interoperability in multivendor networks and allows communication with CISCO, 3COM, SHIVA and other routers.

PPP via ISDN

CS-ROUTE supports PPP for TCP/IP data packages ISDN (see the manual “CMX/CCP, ISDN Communication” [4]).

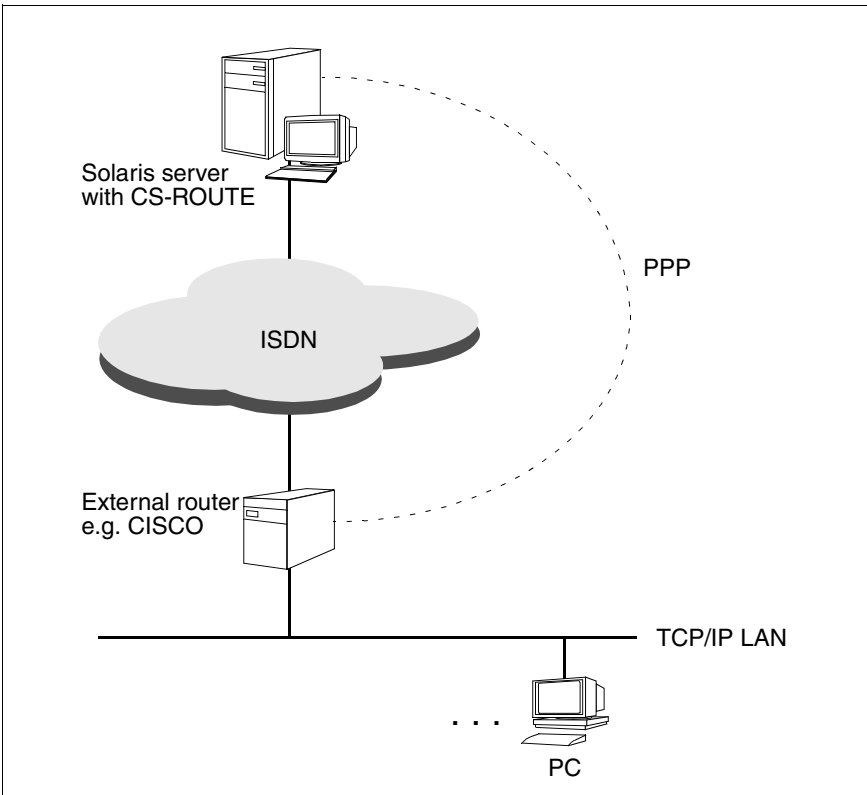


Figure 10: PPP via ISDN

PPP via X.25

CS-ROUTE supports the point-to-point protocol via X.25. This PPP variant is particularly important for connecting Solaris systems with NT systems, since communication with NT systems is only possible via PPP in X.25 (see the manual “CMX/CCP, WAN Communication” [3]).

2.6 Access protection for a subnetwork connection

Objects of the SUBNET class allow you to define access protection for a group of similar connections, which are identified by the subnetwork ID that is common to these connections:

1. All incoming calls are permitted for this subnetwork connection.
2. All incoming calls are rejected.
3. The access limitations as defined in the object class SNPAROUTES by the assigned FACIL object on the basis of the remote subnet address apply:

You can define the following for each subnetwork route:

- Incoming and outgoing calls are permitted.
- Only incoming calls are permitted.
- Only outgoing calls are permitted.
- Neither incoming nor outgoing calls are permitted.

Note: If there is no SUBNET configuration then point 3 applies. See also chapter “FSS commands” on page 83.

2.7 Access protection at user level

User-specific access protection can be implemented within the PPP with the help of the authentication protocols PAP and CHAP.

The Password Authentication Protocol (PAP) is a simple method of determining a partner's identify. The identification and the password are exchanged during the connection setup phase (2-way-handshake). PAP is not a very secure authentication protocol since the passwords are sent over the lines in plain text and thus provide no protection against monitoring.

The Challenge Handshake Authentication Protocol (CHAP) is a much more secure procedure for determining the identity of a partner. The procedure is based on a joint password (Secret) which is only known to the two partners. This joint password (Secret) is not transferred and cannot therefore be monitored.

The system requesting authentication sends a random number to the partner to be identified. The latter encrypts this random number using its Secret and the encryption algorithm MD5. The result is sent back to the system requesting authentication. In this system, the result is compared to the expected value and the appropriate response (authentication successful yes/no) is sent (3-way-handshake).

2.8 Van-Jacobson header compression

Van-Jacobson header compression (VJHC) is a process which appreciably reduces the headers of TCP/IP data packets. TCP/IP headers are between 40 and 120 bytes long. In the worst case, reflected copy, the relationship of header to user data is 40:1. This unfavorable relationship is of little importance in LANs but can, however, have a large adverse effect when transferring data over slow WAN lines. Van-Jacobson header compression can reduce the size of the header to as low as 3 bytes.

Using Van-Jacobson header compression is particularly useful for short user messages (dialog-oriented applications) for reducing the volume of data to be transferred and for reducing costs with volume-dependent charging.

CS-ROUTE supports the VJHC variant with TCP/IP header compression, i.e. both the TCP and IP headers are compressed.

2.9 Message filters

Thanks to this new function, you can either permit or prevent the onward routing of specific TCP-/IP messages. This is performed using so-called filters that can be generated using a CS-ROUTE command. The way filters work is described in section “Message filters” on page 32, and their set up is described in section “Object class fi” on page 67 .

2.10 Supported RFCs

CS-ROUTE supports the following Requests for Comments (RFCs):

RFC 877

J. Korb, Standard for the transmission of IP datagrams over public data networks

RFC 1009

R. Braden, J. Postel, Requirements for Internet gateways

RFC 1144

V. Jacobson, Compressing TCP/IP headers for low-speed serial links

RFC 1321

R. Rivest, The MD5 Message-Digest Algorithm

RFC 1171

D. Perkins, The Point-to-Point Protocol for the Transmission of Multi-Protocol Datagrams Over Point-to-Point Links

RFC 1172

D. Perkins, The Point-to-Point Protocol (PPP) Initial Configuration Options

RFC 1332

G. McGregor, The PPP Internet Protocol Control Protocol (IPCP)

RFC 1334

Lloyd & Simpson, PPP Authentication Protocols

RFC1570

Simpson, PPP LCP Extensions

RFC 1598

Simpson, PPP in X.25

RFC 1618

Simpson, PPP over ISDN

RFC 1661

W. Simpson, The Point-to-Point Protocol (PPP)

RFC 1662

W. Simpson, PPP in HDLC-like Framing

RFC 1994

W. Simpson, PPP Challenge Handshake Authentication Protocol (CHAP)

RFC 2118

G. Pall, Microsoft Point-to-Point Compression (MPPC) Protocol

RFC 2153

W. Simpson, PPP Vendor Extensions

A description of these RFCs can be found on the Web under the address www.ietf.org/rfc.

3 Startup

3.1 Installing/Uninstalling

Installation requirements

CS-ROUTE is a component of the product CMX V5.1 for Solaris and is thus installed and started automatically on installing the CMX V5.1 product. For more details on the installation, refer to manual “CMX, Operation and Administration” [1] and the release notice for CMX V5.1.

3.2 Operating requirements

The following software is needed to run CS-ROUTE:

- The required CCPs must be installed and configured to use the subnetwork interfaces (see manuals “CMX/CCP, WAN Communication” [3] and “CMX/CCP, ISDN Communication” [4]).
- No additional software is needed to use TCP/IP via a WAN.

In addition, you will need to perform the following steps to run CS-ROUTE:

- Configure CS-ROUTE. This includes:
 - Configuring at least one IP interface (see the section “IP interface” on page 28) .
 - Configuring at least one subnetwork interface (see the section “Subnetwork interface” on page 32).
- Configuring the reachability of the CS-ROUTE partner system (see section “Partner systems and their reachability” on page 33).

4 Configuring

In order to use CS-ROUTE, you must:

- configure the local system
- make the partner systems and their reachability known to the local system

You can enter the data with commands, but this requires some detailed knowledge.

The commands are described in the chapter "CS-ROUTE commands" on page 51 and the chapter "FSS commands" on page 83.

FSS and the *fssadm* command are described in detail in the manual "CMX, Operation and Administration" [1].

Configuration examples are described in the chapter "Examples" on page 109.

4.1 Local system

The component CLW (Connectionless WAN Access) is a central part of CS-ROUTE. CLW provides the network access functions and those of LAN-WAN routing for TCP/IP.

CS-ROUTE provides one or more IP interfaces and subnetwork connections for this purpose.

The structure of CS-ROUTE and its embedding in the complete system can be shown in simplified form as follows:

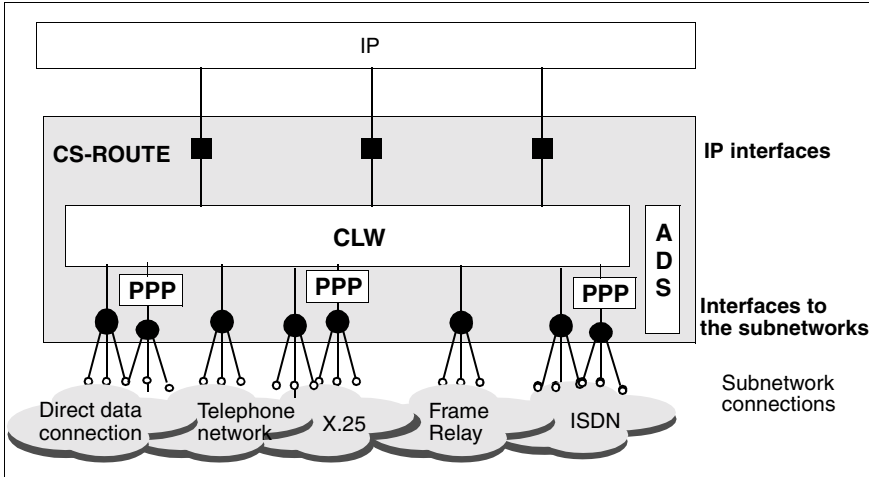


Figure 11: Structure of CS-ROUTE

Local system configuration comprises configuring (at least) one subnetwork interface and (at least) one IP interface.

CS-ROUTE stores the configured parameters in the file `/opt/lib/clw/clw.conf`. This data is loaded on starting CS-ROUTE and determines how CS-ROUTE works.

The configuration may be changed at any time during a session. Some changes are effective immediately, while others require a restart (see the chapter "Overview of administration and diagnosis" on page 47).

4.1.1 IP interface

CS-ROUTE provides the IP module with a network interface for each activated IP interface.

You have the following choice for exchanging IP data packets: you can configure one or more IP interfaces. The various options together with their advantages and disadvantages are described below in the section "Number of IP interfaces" on page 29.

You configure an IP interface as follows:

csr create if name=clwip<n> (see section "Object class if" on page 56)

Mandatory entry: interface name and address

An IP interface can only be used if it is activated.

Number of IP interfaces

CS-ROUTE can work with up to 10 IP interfaces. You can optionally operate all subnetwork interfaces (and subnetwork connections) in the same way via one IP interface or you can tune the assignment between the IP and subnetwork interfaces.

The following applies generally:

You should configure an IP interface if alternate routing is to be used without restrictions via all subnetwork connections. Alternate routing is limited to the subnetwork connections that can be reached via an IP interface.

If an IP interface is to be optimally tuned to a subnetwork type, you should configure an IP interface for each subnetwork type. IP interfaces can be configured optimally if they only have to serve the connections of one subnetwork type.

The default CS-ROUTE configuration provides **one** IP interface. This configuration with just one IP interface provides the greatest flexibility. Further advantages are:

- The local system must only be assigned one IP address.
- All active subnetwork interfaces are used for route selection, provided no restrictions are set by the assignment of a subnetwork ID (list).

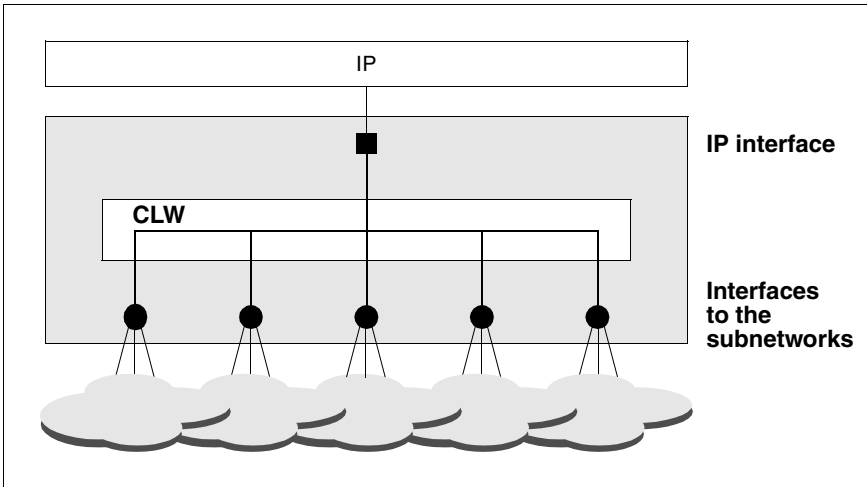


Figure 12: One IP interface

The IP interface is operated across all subnetwork connections, provided they are all operational and all subnetwork interfaces are activated.

Configuring:

You do not need to specify a subnetwork ID (list) to configure an IP interface. All activated subnetwork interfaces are included for route selection.

Example

An IP interface is to be configured.

- with the command:

```
csr create if name=clwip1 ipaddr=195.75.18.49
```

A configuration with just one IP interface can cause bottlenecks when slow subnetwork connections hinder operation of faster ones. It is advisable to configure **multiple** IP interfaces in this case.

If each IP interface is assigned to just one subnetwork interface, the IP interfaces can be configured on a subnetwork-specific basis.

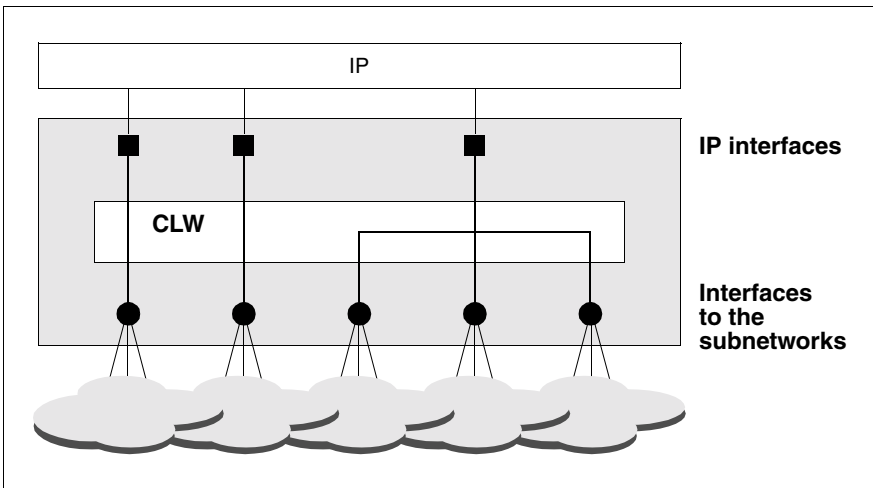


Figure 13: Multiple IP interfaces

Configuring:

The following must be noted when configuring IP interfaces:

- Each IP interface must be assigned a unique IP address.
- Each IP interface must be assigned a unique subnetwork ID or subnetwork ID list, where multiple subnetwork types are also allowed.

This assignment of subnetwork IDs determines the functioning of CS-ROUTE.

“Empty” subnetwork ID lists are permitted. Each subnetwork ID may only be assigned to one IP interface.

Example

One IP interface is to be configured for all existing ISDN connections with the subnetwork ID ISDN-1, and a further IP interface is to be configured for all existing X.25 connections with the subnetwork ID X25-1.

- With the commands:

```
csr create if name=clwip1 ipaddr=195.75.18.49
snid-list=ISDN-1
csr create if name=clwip2 ipaddr=195.75.20.50
snid-list=X25-1
```

4.1.2 Subnetwork interface

You must configure a subnet interface for each subnet type that is to be used by CS-ROUTE. Routes to WAN partner systems are not taken into account unless the corresponding subnet interfaces have been configured and activated. If a WAN partner can be accessed via different subnet types then all the corresponding subnetworks must be activated so that all the various routes can be used.

To configure a subnet interface:

- Use the command `csr create net` (see section "Object class net" on page 62)

Mandatory input: subnet type

4.1.3 Message filters

You can use a message filter to permit or prevent the routing of specific TCP/IP messages. To this end, every inbound or outbound message is checked against the active filters.

A filter is described by an access parameter and a number of filter criteria. The access parameter specifies whether messages which fulfill the filter criteria are forwarded (*access=permit*) or rejected (*access=deny*).

The following filter criteria are possible:

- Direction of the message: input or output
- Sender and/or receiver IP addresses
- Sender and/or receiver IP network masks
- Sender and/or receiver TCP port numbers
- Protocol type: icmp, udp, tcp or ip

The following rules apply whenever a message is checked:

- In the case of a message then only those filter entries that were configured for the corresponding message direction are considered.
- If there is no filter entry for this direction then there are no constraints concerning data transfer in this direction.

- If filter entries exist for this direction then the associated address and protocol parameters are compared with those of the message. When the first hit is encountered the message is processed as determined by the *Access* parameter.
- If there is no hit, then the message is only accepted if there are no *permit* filter entries. *permit* entries are more restrictive than *deny* entries.



Caution!

There is no check of the consistency of multiple filter entries. It is the responsibility of the administrator who makes these entries to ensure that there is no contradiction between the entries.

4.2 Partner systems and their reachability

The way partner systems can be reached is defined in the Forwarding Support Service (FSS). The FSS, which is a CMX component, is described in detail in the manual “CMX, Operation and Administration” [1].

The following applies generally:

With **IP systems**, only the WAN partners, configured as remote systems, and the subnetwork routes to the WAN are configured in the FSS. The IP routes to end systems are configured with the *route add* command (see the section “Configuring IP routes” on page 43).

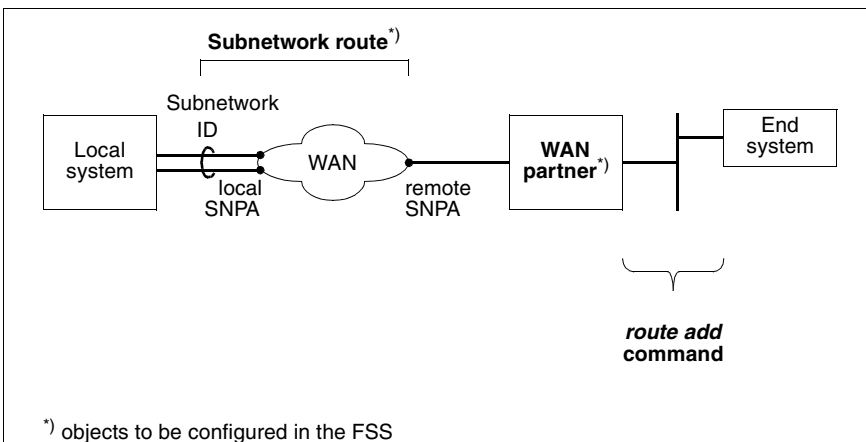


Figure 14: Configuration scheme for IP systems

In order to reach partner systems, you will need to configure the remote system (object name: NSAP), the subnetwork route (object name: SNPAROUTES) and, optionally, the facilities (object name: FACIL) in the FSS.

The order is important when configuring with commands. The syntax prescribes:

1. FACIL (facilities)
2. SNPAROUTES (subnetwork routes)
3. NSAP (remote system)

4.2.1 Facilities

You always configure facilities for specific routes to which these facilities are to be assigned. Facilities must be always be specified in the following cases:

- Van-Jacobsen header compression is to be set (attribute: *compress=TCP/IP*).
- The point-to-point protocol is to be used for the sake of interoperability with external routers (attribute: *ppp-profile=STANDARD* or *ppp-profile=GSM*).
- You wish to use access protection (attribute: *admit*). See also the section “Access protection for a subnetwork connection” on page 36.
- You use maximum integration, minimum integration or X.32 dialing with X.25 communication (attribute: *x25-description*).
- You use frame relay (attributes: *fr-cbs, fr-cir, fr-eps, fr-max-transit-delay, fr-encapsulation, fr-prio*). The values must be negotiated with the network provider and partner system.
- You use ISDN facilities such as the closed subscriber operating class (see the manual “CMX/CCP, ISDN Communication” [4] for details).
- You use X.25 facilities such as the fast select transfer (see the manuals “CMX/CCP, ISDN Communication” [4] and “CMX/CCP, WAN Communication” [3] for details).
- Use of the authentication protocols PAP/CHAP (with point-to-point protocol)
You wish to set the IDLE timer for individual routers (can also be set by a command).

You configure the facilities of partner systems as follows:

- with the command `fssadm create FACIL` (see section "Object class FACIL" on page 88). You must refer to this FACIL object in an SNPAROUTES object (attribute: `facil`).

4.2.1.1 Point-to-point protocol

The manual "CMX/CCP, ISDN Communication" [4] describes how you can configure your ISDN connection to use the point-to-point protocol (PPP). The use of PPP for your Solaris system with CS-ROUTE is defined for each subnetwork route by assigning a facility or FACIL object.

You configure PPP as follows:

- with the command `fssadm create FACIL name=... ppp-profile={STANDARDIGSM}`

PPP-specific facilities cannot be configured. The following default values apply:

Facility	Value
Maximum Receive Unit	1600
Quality Protocol	no
Magic Number	yes
Protocol Field Compression	yes
Address and Control Field Compression	yes
FCS-Alternatives	no
Self-Describing Padding	no
Numbered Mode	no
Multi-Link Procedure	no
Callback	no
Connect Time	no

Table 1: Default values for PPP

4.2.1.2 Access protection for a subnetwork connection

The *incoming-call* parameter in the object class SUBNET allows you to define access protection for a group of similar connections, which are identified by a subnetwork ID common to all the connections. The access protection is specified in the parameter *incoming-call*:

ALL All incoming calls for this subnetwork connection are permitted. Any *admit* attribute configured in the assigned FACIL object is ignored.

RESTRICTED

only incoming calls for which the attribute *admit = BOTH_IN_AND_OUT* or *admit=INCOMING_ONLY* has been configured in the assigned FACIL object are permitted for this subnetwork connection.

NONE

no incoming calls are permitted.

If there is no *incoming-call* attribute, then access control is not active. If you are using the product CS-Route then incoming calls will be permitted or refused depending on the *admit* parameter of the corresponding route.

You can define the use of each route with the *admit* attribute in the FACIL object:

- incoming and outgoing calls are permitted.
- Only incoming calls are permitted.
- Only outgoing calls are permitted.
- Neither incoming nor outgoing calls are permitted.

You configure the use of access protection:

- with one of the following commands

fssadm create FACIL name=... admit=BOTH_IN_AND_OUT, if incoming and outgoing calls are to be permitted,

fssadm create FACIL name=... admit=OUTGOING_ONLY, if only outgoing calls are to be permitted and

fssadm create FACIL name=... admit=INCOMING_ONLY, if only incoming calls are to be permitted.

4.2.1.3 Access protection at user level

When using PPP, you can configure password protection (PAP or CHAP) at the user level.

In both cases, you specify the name of the PPPAUTH object in `ppp-auth-params`.

- ▶ For authentication purposes, configure a PPPAUTH object containing an identification for the partner system and the password as `pap-peer-pwd` or `chap-peer-secret`. The password must be stored in a file, i.e. it cannot be specified directly in the `fsadm` command.

fsadm create PPPAUTH name=... peer-id=... pap-peer-pwd=<file>, if you want to use the simpler password variant with an unencrypted password,

fsadm create PPPAUTH name=... peer-id=..., loc_id=..., chap-peer-secret=<file>, if you want to use identification without the direct exchange of passwords.

- ▶ Configure a FACIL object containing the attributes `ppp-auth-protocol` and `ppp-auth-params`:

fsadm create FACIL name=... ppp-auth-protocol=PAP ppp-auth-params=..., if you want to use the simpler password variant with an unencrypted password,

fsadm create FACIL name=... ppp-auth-protocol=CHAP ppp-auth-params=..., if you want to use identification without the direct exchange of passwords.

4.2.1.4 Van-Jacobson header compression

Van-Jacobson header compression (VJHC) use is either configured statically for a subnetwork route by assigning a facility or FACIL object or negotiated via the WAN partner by using the point-to-point protocol.

If you use the point-to-point protocol, no special points have to be noted with respect to the VJHC. The WAN partners agree among themselves on whether, in which direction and with which slot size compression is used.

If VJHC is configured statically for a route, both WAN partners must be configured identically for using header compression and the slot size otherwise data may be lost.

For each TCP end-to-end connection that is multiplexed over a subnetwork connection, a storage area is required in which

- the TCP/IP header of the last message sent is stored before compression
- the TCP/IP header of the last message received is stored after compression.

Each of these storage areas is called a slot, and the complete area is called a slot array. Because of the limit on the number of headers which may be stored at any one time, a migration mechanism is used if the number of TCP connections to be multiplexed over a subnetwork connection exceeds this limit.

Correct slot array dimensioning ensures that

- no storage space is used if, e.g. very few TCP connections are used
- no unnecessary delays occur because of the migration mechanism when the number of multiplexed TCP connections exceeds the slot size.

You configure VJHC usage as follows:

- with the command `fssadm create FACIL name=... compress=TCP/IP`

You configure the slot array size as follows:

- with the command `csr set clw maxslot=...`

4.2.2 Subnetwork routes

A subnetwork route describes the route from the local host to a remote host within a subnetwork. If the remote host - as defined by the remote IP address - is located in a different subnetwork to the local host, the subnetwork route describes the route from the local host to the network gateway ("next hop"), where further routing to the remote host is then performed by the IP instance.

A subnetwork route is defined by its end points:

the subnetwork ID in the local system and the subnetwork address of the remote host (target host or next hop) in the same network. A subnetwork ID identifies a subnetwork connection or a group of similar subnetwork connections in the same subnetwork that the transport system can address using this ID.

4.2.2.1 Alternate routing

You must configure every alternative route to the partner system if you wish to use the alternate routing function. You create a list of these alternative routes via which the partner system is to be reached and assign it to the partner system, identified by an IP address. You can weight the routes within the list and thus define its selection priority.

Route selection

The following applies generally: all routes that are configured (FSS) to lead to the same WAN partner (identified by an IP address) are taken into account..

Restrictions can occur due to

- the assignment of subnetwork IDs to the IP interface
- the status of the subnetwork interface (DOWN)

You configure alternate routing as follows:

- with the commands:

```
fssadm create SNPAROUTES name=<route1> ...
```

```
fssadm create SNPAROUTES name=<route2> ...
```

```
fssadm create SNPAROUTES name=<route3> ...
```

```
fssadm create NSAP snpa-list=
```

```
"<route1>[/<weight>]+<route2>[/<weight>]+<route3>[/<weight>]"
```

You define an SNPAROUTES object for each route. You list all routes over which your partner system is to be reached in the appropriate NSAP object with the *snpa-list* attribute. Each route can be assigned a priority.

4.2.2.2 Minimum integration

The following figure shows the addressing scheme for minimum integration:

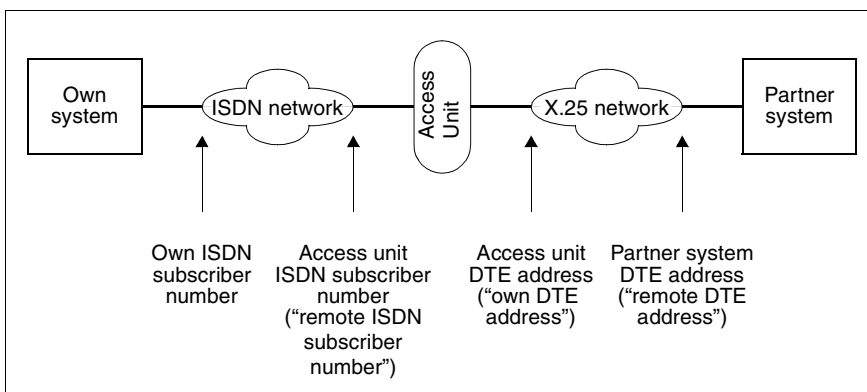


Figure 15: Addressing scheme of X.25 minimum integration

The manual “CMX/CCP, ISDN Communication” [4] describes how you must configure your ISDN connection for minimum integration. Your own ISDN subscriber number and the DTE address of the access unit (“own DTE address”) must be specified.

The ISDN subscriber number of the access unit (“remote ISDN subscriber number”) and the DTE address of the partner system (“remote DTE address”) are specified in the FSS when the subnetwork route is defined.

You configure minimum integration as follows:

- with the commands

```
fssadm create FACIL name=... x25-description=...
```

```
fssadm create SNPAROUTES name=... x31-msa=... facil=...
```

The *x25-description* attribute specifies the name of the predefined description of the X.25 connection in the CC configuration file (DTE name).

The subscriber number of the access unit and the DTE address of the partner system are defined with the *x31-msa* (msa = multistage address) attribute.

4.2.2.3 Maximum integration

The following figure shows the addressing scheme for maximum integration:

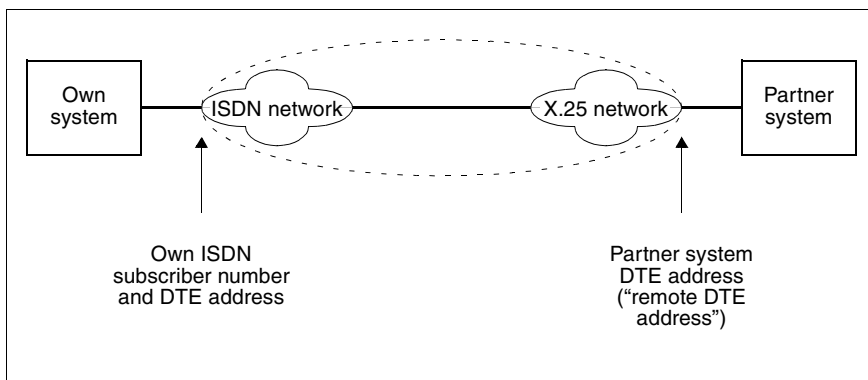


Figure 16: Addressing scheme of X.25 maximum integration

The manual “CMX/CCP, ISDN Communication” [4] describes how you can configure your ISDN connection for maximum integration. Your own ISDN subscriber number and DTE address of the access unit must be specified in the CC configuration file.

The DTE address of the partner system (“remote DTE address”) or the PVC number is specified in the FSS when the subnetwork route is defined.

You configure maximum integration as follows:

- with the following commands

```
fssadm create FACIL name=... x25-description=...
fssadm create SNPAROUTES name=... x31-dte-addr=... facil=...
```

or

```
fssadm create FACIL name=... x25-description=...
fssadm create SNPAROUTES name=... x31-pvc=... facil=...
```

The *x25-description* attribute specifies the name of the predefined description of the X.25 connection in the CC configuration file (DTE name).

The DTE address of the partner system is specified with the *x31-dte-addr* attribute, and the PVC number with the *x31-pvc* attribute.

4.2.2.4 X.32 dialing

The following figure shows the addressing scheme for X.32 dialing:

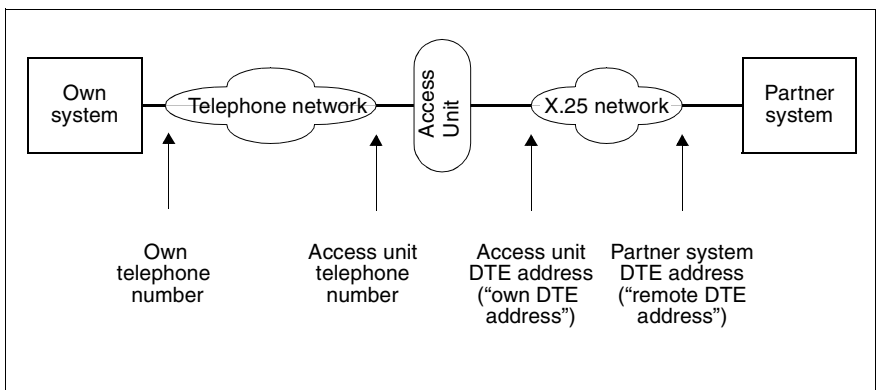


Figure 17: Addressing scheme for X.32 dialing

The manual “CMX/CCP, WAN Communication” [3] describes how you configure your telephone connection for X.32 dialing. You will need to specify your own telephone number and that of the access unit as well as the DTE address of the access unit.

The telephone number of the access unit and the DTE address of the partner system (“remote DTE address”) are specified in the FSS when defining the subnetwork route. You have to define an additional subnetwork route if the local system is to be both an active and passive partner at connection setup. This involves specifying X.32 dialing for the local telephone number.

You configure X.32 dialing as follows:

- with the commands:

```
fssadm create FACIL name=... x25-description=...
```

```
fssadm create SNPAROUTES name=... x32-phone-nr=... facil=...
```

```
fssadm create SUBNET subnet=... x25-description=...
```

You make the name of the KOGS *XZSTW* macro which describes the X.25 access known with the *x25-description* attribute.

You make the access unit telephone number (X.25 exchange) and partner system DTE address known with the *x32-phone-no* attribute.

You must make the subnetwork ID of your own telephone number known in the SUBNET object with the *subnet* attribute. You use the *x25-description* attribute to specify X.32 dialing for incoming calls and at the same time reference the description of the X.25 access if you also want to receive incoming calls.

4.2.3 Information on the partner system

You configure the WAN partner systems as follows:

- with the command *fssadm create NSAP* (see section “Object class NSAP” on page 93)

Mandatory entry: partner system name, network address.

Additionally, for WAN partners: routes via which the partner system is reached.

4.2.4 Configuring IP routes

This section describes the configuration steps required to reach partner systems from the IP viewpoint. To find the route, the system needs to know the local IP interface and possibly the routers via which the partner system can be reached. The number of routers via which the system is reached must be specified as metric.

Systems are generally reachable without additional route entries if they are in the same subnetwork as the configured IP interface. Note that the term “subnetwork” has a different meaning in connection with IP than it does in connection with WANs.

From the IP viewpoint, hosts are in the same subnetwork if the network and subnetwork numbers of the IP address match. The network number and optional subnetwork number are fields in the IP address. The network number is the first part of the IP address and can be 1, 2 or 3 bytes long. The remaining bits are split into the subnetwork number (optional) and host number.

You must make additional route entries with the command *route add* (see the online manual pages) for all hosts which, on the basis of their IP address, are in a different subnetwork than the configured IP interface.

In order to ensure that the required route commands are executed automatically whenever CS-ROUTE is started (especially when rebooting the system), you can place them in a file, for which the path name is specified on executing the command *csr create if name=clwip<n>* (see also the release notice for CMX).

Example

The local system with CS-ROUTE has the IP address 172.16.10.1. This is a class B address and the network number is therefore 2 bytes long. All systems with the same network number (172.16.xxx.xxx) are reachable via the configured IP interface from the IP viewpoint.

Route entries are required for all other systems.

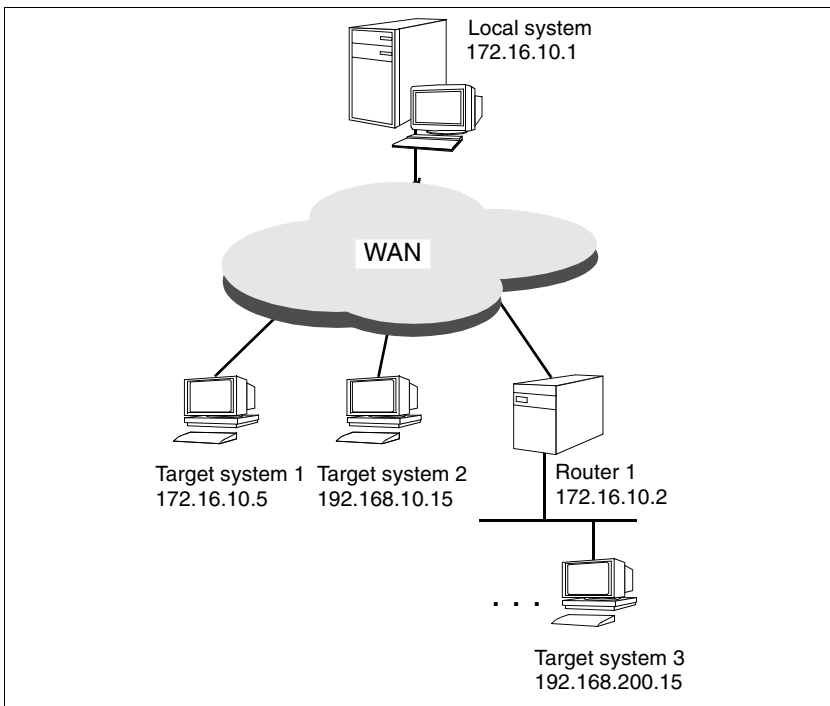


Figure 18: IP routes

Case 1:

The target system is in the same network/subnetwork.

Target system 1 with IP address 172.16.10.5 is in the same subnetwork as the local system with IP address 172.16.10.1.

Route entries are not required.

Case 2:

Target system 2 with IP address 192.168.10.15 is in a different network than the local system with IP address 172.16.10.1.

- ▶ Configure a route to target system 2.

Enter the IP address of your own interface as the gateway and always specify the value 0 as metric.

```
route add host 192.168.10.15 172.16.10.1 0
```

Case 3:

Target system 3 with IP address 192.168.200.15 is reachable via router 1 with IP address 172.16.10.2. From the IP viewpoint, router 1 is in the same network as the local system with IP address 172.16.10.1.

- ▶ Configure a route to target system 3 via router 1.

Enter the IP address of the directly adjacent system as the gateway and specify as metric a value > 0 (number of routers).

```
route add host 192.168.200.15 172.16.10.2 1
```

(If you want to address all hosts in the target network rather than just one target host, you can also specify

```
route add net 192.168.200 172.16.10.2 1.)
```

5 Overview of administration and diagnosis

Starting CS-ROUTE

Once you have installed the CS-ROUTE package, CS-ROUTE is automatically started whenever you start the system.

You can deactivate this autostart function with the command *csr autostop* and reactivate it with *csr autostart* (see chapter “CS-ROUTE commands” on page 51).

You can start CS-ROUTE explicitly with the command *csr start* (see chapter “CS-ROUTE commands” on page 51).

Stopping CS-ROUTE

You stop CS-ROUTE with the *csr stop* command (see section “Starting/stopping CS-ROUTE” on page 53).

TCP/IP is not affected by this command, so unrestricted communication is still possible on the LAN.

CS-ROUTE status information

You can call up general CS-ROUTE status information with the *csr command* (see section “CS-ROUTE” on page 53).

This displays the CS-ROUTE daemons with their process IDs and the status of the autostart function.

Activating/deactivating an IP interface

IP interfaces are automatically activated when CS-ROUTE is started up if they were active the last time CS-ROUTE was run.

If an interface is configured while CS-ROUTE is running then you must activate it so that CS-ROUTE can be operated via this route.

Sometimes it may be necessary to deactivate an interface without stopping CS-ROUTE, e.g. if you need to reset the interface parameters. CS-ROUTE continues to operate with the other interface. Any subnetwork connections that are not linked to this interface are retained.

To activate or deactivate an IP interface, use the command *csr set if... state=up/down* (see section “Object class if” on page 56).

Overview of administration and diagnosis

If an interface is deactivated then all the subnetwork connections attached to this interface are disconnected.

Modifying the configuration parameters for an IP interface

You change the configuration parameters for an IP interface with the command *csr set if* (see section “Object class if” on page 56).

Changes to the subnetwork ID list and the Maximum Transfer Unit do not take effect until CS-ROUTE is restarted. Changes to other operating parameters take effect immediately.

Activating/deactivating a subnetwork interface

Subnetwork interfaces are activated automatically on starting up CS-ROUTE provided they were active during the last CS-ROUTE run. If an interface is configured during a CS-ROUTE session, you will need to activate it in order to enable CS-ROUTE to operate via this interface.

It may be necessary to deactivate a subnetwork interface without stopping CS-ROUTE, e.g. if a subnetwork connection fails or the parameter settings for an interface have to be changed. CS-ROUTE then continues working with the remaining interfaces. Subnetwork connections over these other interfaces remain intact.

You can activate/deactivate a subnetwork interface with the *csr set net ... state=up|down* command (see section “Object class net” on page 62).

When an interface is deactivated, all subnetwork connections running over this interface are shut down.

Modifying the configuration parameters for a subnetwork interface

You modify the configuration parameters for a subnetwork interface with the *csr set net* command (see section “Object class net” on page 62).

Activating/deactivating message filters

Message filters are automatically activated when CS-ROUTE is started if they were active during the last CS-ROUTE run. If a filter is configured while CS_ROUTE is running then it must also be activated if it is to take effect. It may be necessary to deactivate a filter without stopping CS-ROUTE, e.g. if you need to change the parameter settings for the filter.

To activate or deactivate a message filter, you use the command *csr set if... state=up|down* (see section “Object class fi” on page 67).

Modifying the configuration parameters for a message filter

You can modify the configuration parameters for a filter using the command *csr set fi* (see section “Object class fi” on page 67).

All the changes take effect as soon as you set the filter to *active*.

Modifying the CS-ROUTE operating parameters

You modify the size of the slot array and the predefined maximum transfer unit with the command *csr set clw* (see section “Object class clw” on page 54).

Slot array size changes are effective immediately, while changes to the maximum transfer unit only take effect after a CS-ROUTE restart.

Shutting down subnetwork connections

You can shut subnetwork connections down explicitly using the command *csr set conn ref=<connection_reference>... state=down* (see section “Object class conn” on page 74).

Retrieving status/statistic data

You can only query the status of the CS-ROUTE object. However, in the case of the IP interfaces, the subnetwork interfaces and the message filters you can output the status or statistical values.

To do this, you use the commands *csr get clw*, *csr get if*, *csr get conn*, *csr get net* and *csr get fi*.

You can display information on the local subnetwork connections with the *csr get lsn* command (see section “Object class lsn” on page 73).

6 CS-ROUTE commands

The *csr* command is used for configuring CS-ROUTE in expert mode.

Description format

Please refer to section “Notational conventions” on page 3 in this manual for the description format used for the following *csr* command.

Actions

When you use the *csr* command you apply specific actions to CS-ROUTE or the object classes and their attributes.

The following actions can be specified for CS-ROUTE in the *csr* command:

start Starts CS-ROUTE.

stop Stops CS-ROUTE.

autostart

Activates the autostart function.

autostop

Deactivates the autostart function.

The following actions can be specified for object classes and their attributes in the *csr* command. Refer to the description of the object class concerned to find out which actions are permitted for a specific object class.

create

Creates an object with the specified attribute values.

delete

Deletes an object.

set Sets the attributes of an object to the specified values.

get Retrieves objects in the specified object class with their attributes.

Examples of *csr* commands

```
csr start
csr set if name=clwipl mtu=1200 snid-list="X25-1 X25-2"
```

CS-ROUTE commands

Help functions

You call help on command syntax with the following commands:

csr ? Displays the actions.

csr action ?

Displays the object classes for which an action is possible.

csr action object class ?

Displays the attributes which are suitable for the specified object class.

Overview of actions, object classes and attributes

The following table provides an overview of the object classes. It also shows which actions you can apply to a specific object class and which attributes you can assign to the object class.

Action	Object class	Attribute
get set	clw	mtu, maxslot
get set	conn	type, ref, state, statistics
create delete get set	if	name, ipaddr, netmask, snid-list, mtu, state, statistics
get	lsn	-
create delete get set	net	type, maxplcon, maxcon, inmax, outmax, state, statistics
create delete get set	fi	name, access, direction, protocol, srcaddress, destaddress, srcmask, destmask, srcport, destport state, statistic

Table 2: Actions, object classes and attributes of the csr commands

The sections below first describe the commands that can be used with CS-ROUTE as a whole. The object classes that are of relevance for the use of CS-ROUTE are then described.

6.1 CS-ROUTE

The following commands can be used on CS-ROUTE:

CS-ROUTE status information

csr

The CS-ROUTE daemons are displayed with their process IDs: *clwd* (clw daemon), *pppd* (PPP daemon) and the status of the autostart function.

6.1.1 Starting/stopping CS-ROUTE

csr {**_start**|**_stop**} [**ppp**]

start Starts CS-ROUTE. All configured IP or OSI-CLNP interfaces and subnetwork interfaces are activated.

stop Stops CS-ROUTE. The current configuration is terminated.

ppp Starts/stops the PPP daemon.

6.1.2 Autostart function of CS-ROUTE autostart function

csr {**_autostart**|**_autostop**} [**_snmp**]

autostart

Starts CS-ROUTE automatically on starting up the system.

autostop

Deactivates the automatic startup of CS-ROUTE on starting up the system.

6.2 Object class clw

You configure the CLW component with a clw object.

The following commands are permitted:

csr_g[et] [_clw]

csr_s[et] [_clw] { _mtu= | _maxslot=}

The *csr get clw* command displays CLW status information.

You can modify the maximum transfer unit and/or the slot array size with the *csr set clw* command.

mtu=

Maximum length of user data (maximum transfer unit) for the IP interface (in bytes).

This value is used if no interface-specific value was defined with the *csr create if* command.

Possible values: ≥ 41

Default value: 1600

Changes to the maximum transfer unit only take effect after CS-ROUTE is restarted.

maxslot=

Size of the slot array with Van-Jacobson header compression.

This parameter determines the number of end-to-end connections whose headers can be stored at a specific time. The parameter does not limit the number of connections which can be multiplexed over a subnetwork connection.

The slot array size should match the number of end-to-end connections which are active simultaneously. Please note that the value must be negotiated with the communication partner.

If Van-Jacobson header compression is used without using the point-to-point protocol, the slot array size must be configured the same in both WAN partner systems. Data may be lost if the two configurations are inconsistent.

The slot array size is negotiated between the WAN partners if the point-to-point protocol is used.

Possible values: 3...255

Default value: 10

Output:

if-list List of configured IP interfaces independently of their status

---: No IP interface is configured/activated

maxslot

Slot array size with Van-Jacobson header compression

mtu Maximum length of user data for the IP interface (in bytes)

net-list

List of configured subnetwork interfaces independently of their status.

---: No subnetwork interfaces are configured/activated

start/stop

Time at which the CLW daemon clwd was activated or deactivated

state Current state of the CLW component

up: Activated

down:

Deactivated

6.3 Object class if

An if object represents an IP interface.

The following commands are permitted:

```
csr_cr[reate]_if [_name=]clwipn [_ipaddr=] [_netmask=] [_snid-list=]
    [_mtu=][_s[tate]=]
```

```
csr_d[ele]te_if [_name=]clwipn
```

```
csr_g[et]_if [_name=]clwipn [_l={l|m|h}]
```

```
csr_s[et]_if [_name=]clwipn[_ipaddr=] [_netmask=] [_snid-list=]
    [_mtu=][_s[tate]=] [_stati[stics]=]
```

An interface description with the specified attributes is created with the *csr create if* command. The configuration data is stored if the command was successfully executed.

To prepare a configured interface for operation, you must activate it with the *state=up* parameter either directly during configuration (*csr create if* command) or subsequently (*csr set if* command).

You can delete an interface configuration with *csr delete if*. The interface is then deactivated and all subnetwork connections which are being served via the interface at this time are shut down.

You can display information on configured IP interfaces with *csr get if*. You can obtain detailed information with *csr get if-l={l|m|h}*.

name=

Interface name

The interface name is made up of the fixed name part `clw`, the name of the layer 3 protocol (`ip` or `clnp`) and the interface index (0 or n).

clwipn

For an IP interface

Possible values for n : 0...9

A range can be specified as the index with *get*: **clwip**[m - n]

Possible values for m and n : 0...9; $n \geq m$

name= is used as a filter for the *get* action.

ipaddr=

IP address of the interface in dotted notation (e.g. 172.16.10.1) or in hexadecimal notation with the prefix `0x` (e.g. `0xac100a01`)

Only required for IP interfaces.

Changes to the IP address only take effect after the interface is activated the next time.

netmask=

Network mask of the interface in dotted notation (e.g. 255.255.255.128) or in hexadecimal notation with the prefix `0x` (e.g. `0xfffff80`).

Default value: the following network masks are used, depending on the network class.

Network class	Network mask	
Class A network	0xff000000	255.0.0.0
Class B network	0xffff0000	255.255.0.0
Class C network	0xfffff000	255.255.255.0

Changes to the network mask only take effect after the interface is activated the next time.

snid-list=

One or more subnetwork IDs which are assigned to the interface.

Only one IP interface can be used if you do not specify *snid-list=*. These interfaces are assigned all known subnetwork IDs.

Note that each subnetwork ID may only be assigned to one IP interface.

FR-n

With frame relay accesses.

Possible values for *n*: 1...128

ISDN-n

With ISDN accesses.

Possible values for *n*: 1...32

PP-n

With direct data connections.

Possible values for *n*: 1...32

PT-n

With analog telephone connections.

Possible values for *n*: 1...32

X25-n

With X.25 accesses.

Possible values for *n*: 1...32

You can also specify several separate subnetwork IDs. Each entry must then be separated by a blank and enclosed in quotes (e.g. *snid-list= "ISDN-1_ ISDN-4_ ISDN-30"*).

Changes to the subnetwork ID only take effect after the interface is activated the next time.

mtu=

Maximum length of user data for the interface (in bytes)

Possible values: ≥ 41

Default value: the value that was defined for object class *chw*.

Changes to the maximum transfer unit only take effect after the interface is activated the next time.

state=

up The status of the interface is set to active. If CS-ROUTE is already started then the interface is activated immediately. Otherwise, the interface is not activated until CS-ROUTE is started

down

The status of the interface is set to inactive.

All subnetwork connections concerned are shut down. The interface remains inactive even when CS-ROUTE is (re)started.

statistics=0

Deletes the interface-specific statistical data.

-l= Controls the extent of the output.

Default value: l

l The most important attributes are output in tabular form ("level=low").

m The described attributes are output ("level=medium"), see table "Attribute output for if objects" on page 60.

h Statistical data is also output in addition to described attributes ("level=high").

Output:

The attributes output with *get* are listed in the following table:

Attribute	<i>get</i>		
	<i>-l=l</i>	<i>-l=m</i>	<i>-l=h</i>
ipaddr	x	x	x
name	x	x	x
netmask		x	x
mtu		x	x
snet	x	x	x
start/stop		x	x
state	x	x	x
statistical data			x

Table 3: Attribute output for if objects

ipaddr

Interface IP address in dotted notation

Only for IP interfaces

name

Interface name

netmask

Interface network mask in hexadecimal and dotted notation

Only for IP interfaces

mtu

Maximum length of user data for the interface (in bytes)

snet

Subnetworks and associated IDs that are assigned to the interface

all: All known subnetwork IDs are assigned to the interface (default setting if certain subnetwork IDs have not been explicitly assigned to the interface).

---: No subnetwork connection reported by the CCP's

cfg: Explicitly assigned subnetwork IDs



The layout for the listed subnetworks has been extended. The status of each subnetwork is also output.

start/stop

The time at which the interface was activated or deactivated.

state Current or configured status of the interface.

If CS-ROUTE and at least one of the associated subnetwork interfaces is active then the current status is output:

up: active

down:
inactive

If CS-ROUTE is not active or none of the associated subnetwork interfaces has been activated then the current status is output:

cfg: up:
Configured as active.

cfg: down:
Configured as inactive.

This status is set when CS-ROUTE is started or, if CS-ROUTE is already active, following the activation of at least one of the associated subnetwork interfaces.

6.4 Object class net

A net object represents an interface to a subnetwork type. Subnetwork routes to WAN partners can only be used after a subnetwork interface has been configured and activated.

The following commands are permitted:

```
csr_cr[reate]_n[et]_t[type] [=] [_maxplcon=] [_maxcon=] [_inmax=]
    [_outmax=][_s[tate]=]
```

```
csr_d[ele]t[e]_n[et]_t[type]=
```

```
csr_g[et]_n[et] [_t[type]=] [_-l={l|m|h}]
```

```
csr_s[et]_n[et]_t[type] [=] [_maxplcon=] [_maxcon=] [_inmax=]
    [_outmax=][_attributes=default] [_s[tate]=] [_stati[stics]=]
```

A subnetwork interface description is created with the specified attributes using the *csr create net* command. A subnetwork interface with default values is configured if you do not specify any further attributes. You can display the default values with the *csr get net type=... attributes=default* command.

If the *csr create net* command executes successfully then the configuration data is saved. To prepare a configured interface for operation, you must activate it with the *state=up* parameter, either directly during configuration (*csr create net* command) or subsequently (*csr set net* command).

You can delete the interface configuration with *csr delete net*. The interface is deactivated and all subnetwork connections that are currently served via this interface are also shut down.

You can use *csr get net* to display all the subnetwork interfaces, including those that have not yet been configured. Detailed information on the interfaces can be called up using the *csr get net-l={l|m|h}* command.

type=

Type of subnetwork interface

fr Frame relay

isdn ISDN

pp Direct data connection

pt Analog telephone network

x25 X.25

type= is used as a filter for the *get* action.

Changes to the subnetwork type only take effect after CS-ROUTE is restarted.

maxplcon=

n Maximum number of parallel connections to a WAN partner system.

If the attribute value deviates from the default configuration, you must note that if you set this value too high, you may not be able to configure connections to other WAN partners.

unl Unlimited number of parallel connections to a WAN partner system

maxcon=

n Maximum number of simultaneously active subnet connections of the type specified (see *type*)

unl Unlimited number of simultaneously active connections

inmax=

sec Permitted idle time for incoming connections after which the connections are to be shut down (in seconds)

Possible values: 0...32767

unl No idle time monitoring

outmax=

sec Permitted idle time for outgoing connections after which the connections are to be shut down (in seconds)

Possible values: 0...32767

unl No idle time monitoring

[attributes]=[default]

Displays the default values of all attributes (with *csr get net*) or the attributes are set to their default values (with *csr set net*).

state=

up The status of the interface is set to active. If CS-ROUTE is already started then the interface is activated immediately. Otherwise, the interface is not activated until CS-ROUTE is started.

down

Deactivates the subnetwork interface.

All subnetwork connections concerned are shut down. The interface remains inactive even when CS-ROUTE is (re)started.

statistics=0

Deletes the subnetwork-specific statistical data.

-l= Controls the extent of the output.

Default value: l

l The most important attributes are output in tabular form (“level=low”).

m The described attributes are output (“level=medium”), see table “Attribute outputs for net objects” on page 64.

h Statistical data is output in addition to the described attributes (“level=high”).

Output:

The attributes output with *get* are listed in the following table:

Attribute	get		
	-l=l	-l=m	-l=h
actcon		x	x
inmax		x	x
inmin		x	x
maxcon		x	x
maxconp		x	x
maxplcon		x	x
maxplconp	x	x	x
outmax		x	x

Table 4: Attribute outputs for net objects

Attribute	get		
	-l=l	-l=m	-l=h
outmin		x	x
snid-list	x	x	x
start/stop		x	x
state	x	x	x
type	x	x	x
statistic data			x

Table 4: Attribute outputs for net objects

actcon

Number of currently active connections

inmax

Permitted idle time for incoming connections after which the connections are to be shut down (in seconds)

unlimited:

No idle time monitoring

inmin

Minimum idle time for incoming connections after which the connections may be shut down (in seconds)

unlimited:

No idle time monitoring

maxcon

Maximum number of simultaneously active connections

maxconp

Maximum number of simultaneously active connections since the interface was activated (peak until now)

maxplcon

Maximum number of parallel connections to a WAN partner

maxplconp

Maximum number of parallel active connections to a WAN partner since the interface was activated (peak until now)

outmax

Permitted idle time for outgoing connections after which the connections are to be shut down (in seconds)

unlimited:

No idle time monitoring

outmin

Minimum idle time for outgoing connections after which the connections are shut down (in seconds)

unlimited:

No idle time monitoring

snid-list

Subnetwork IDs of the subnetwork accesses to the corresponding subnetwork type reported at this time by the CCPs

---: No subnetwork access

start/stop

Time at which CS-ROUTE operation via the subnetwork was permitted or was no longer permitted

state Current or configured status of the subnetwork interface

If the interface is configured then the following statuses may be output dependent on the status of CS-ROUTE:

If CS-ROUTE is active:

up: Active

down: Inactive

If CS-ROUTE is inactive:

cfg: up:

Configured as active

cfg: down:

Configured as inactive

The displayed status is set after CS-ROUTE is started.

default:

No subnetwork interface description created using the *csr create net* command

type Type of subnetwork

6.5 Object class fi

An fi object represents a filter with certain filter criteria. You have to activate a configured filter in order for it to become effective.

Each incoming or outgoing message is checked against the active filters. If the filter criteria correspond to the message's address and protocol parameters then the message is either routed on or rejected depending on the action assigned to the filter.

The following commands are permitted:

```

csr <action> fi [name=] <filtername>
    [[access=] <filter-action>]
    [[d[ir]=]<direction>]
    [[p[rot]=]<protocol>]
    [[srcaddr= <address>] [mask=<address-mask>]]
    [[destaddr=<address>] [mask=<address-mask>]]
    [[srcport= <port nr>] [[op=]<operator for port nr>]]
    [[destport= <port nr>] [[op=]<operator for port nr>]]
    [state= <state>]
    [stat[istic]]
  
```

You do not need to enter any keywords if the parameters associated with them are only able to accept defined values. You can also omit the keyword *name*. In this case, the first parameter after the keyword *fi[lter]* is interpreted as the filter name.

<action>

Describes the action to be performed at the filter:

create

Creates a filter file

delete

Deletes a filter file

get Reads a filter file (or files)

set Modifies a filter file (or files)

name=

Name of the filter; user-defined string of up to 32 characters.

access=

Specifies whether the messages that meet the filter criteria are routed onwards or rejected:

permit

The messages are to be routed.

deny The messages are to be rejected.

If you do not enter a value for this parameter, *deny* is set by default.

prot= Specifies the protocol type of the messages to which the filter criteria apply:

tcp TCP data packets

udp UDP data packets

icmp ICMP data packets

ip Incorporates all the protocols: TCP, UDP and ICMP

If you do not specify the parameter then *ip* is set by default.

srcaddr=

Sender IP address in dot notation.

destaddr=

Destination IP address in dot notation.

mask=

Mask for the corresponding IP address in dot notation:

Bits that are set to *1* are ignored in the corresponding message address when the comparison is performed.

Specify this parameter immediately after the IP address.

srcport=

Sender port number.

destport=

Destination port number.

op= Operation that is to be applied to the specified port number:

eq Only applies to this port number

ne Applies to all port numbers apart from this one

gt Applies to all port numbers larger than this one

lt Applies to all port numbers smaller than this one

If you do not specify an operation then the value is set to *eq* by default.

You must enter this parameter directly after the port number.

state=

Status of the filter:

up The filter is to become active.

down The filter is to become/remain inactive.

If you do not specify a status then the value is set by default to *down* if the filter is recreated. Otherwise the previous status is retained.

statistic

Resets the statistics counters (only in the event of *csr set fi [<name>]*)

Call-related validity of parameters:

– **csr cr[eate]fi[lter]**

You must enter the filter name. All the other parameters are possible. The *statistic* parameter is of no relevance.

– **csr g[et] fi [[[name=]<filter-name>] [[access=]<filter-action>]**

[[d[ir]=<direction>] [[p[rot]=<protocol>][state=<state>]][-l={l|m|h}]

If you specify selection parameters then the filters are displayed with the corresponding parameter values.

Example:

If you specify *dir=inp state=up* then you can specify the input direction for all the active filters.

The display type is defined using the *-l* switch which acts in the same way as for the other CS-ROUTE objects

-l=l Only outputs the filter names (default setting)

-l=m Outputs the filters together with all their attributes.

`-l=h` In addition to the attributes, also outputs the number of packets that have so far matched the filter criteria (statistics).

If you specify neither a name nor any parameters then all the filters are displayed. If you additionally specify the `-l=h` switch then all the statistics for all the filters are displayed. The layout of the output is described in section “Partner systems and their reachability” on page 33.

– `csr s[et] fi[lter]`

All the parameters are valid (similar to `csr cr fi`).

You must specify the filter name unless you only specify the `state` parameter in which case the statistics counters for all the filters are reset.

The status of the filters remains unchanged unless you initiate a change by specifying the `state` parameter.



You can modify, but not delete, addressing parameters (IP addresses and port numbers) using the `set` command. If you want to delete addressing parameters, you must delete and then recreate the filter.

– `csr d[ele]te fi[lter]`

You may only enter the filter name as a parameter. You are not permitted to specify any further parameters.

Output:

Command `csr get fi <filter-name>`

You use the command `csr get fi <filter-name>` to read a specific filter together with its attributes.

If the `-l=h` switch is not set then the output has the following format:

`name= <filter-name>`

`filterid= <number>`

Internally assigned identifier for filtering

`access= {permit | deny}`

`direction= {input | output}`

protocol= {ip | tcp | udp | icmp}

srcaddr= <IP-address> [with mask <netmask>]

Lines containing an IP address and network mask are not output unless this is explicitly specified.

destaddr= <IP-address> [with mask <netmask>]

Lines containing an IP address and network mask are not output unless this is explicitly specified.

srcport= <port-number> with operation= {eq | ne | gt | lt}

Lines containing a port number are not output unless this is explicitly specified.

destport= <port-number> with operation= {eq | ne | gt | lt}

Lines containing a port number are not output unless this is explicitly specified.

state= {configured as inactive | configured as active}

This text is output if the CS-ROUTE daemon is not active.

state= {inactive | active} [since <date>]

The status *active* or *inactive* is output when CS-ROUTE is started.

If you specify the *-l=h* switch then the statistic counter below is also output:

statistic sampled since <date>

hits of packets= <number>

Command `csr get fi` [<selection parameters>]

You use the command `csr get fi` [<selection-parameters>] to display **all** the filters whose parameters correspond to the specified selection parameters.

The *-l=* switch determines the type of display:

-l=l (or no switch specification)

– <filter-name1>

– <filter-name2>

– etc.

-l=m All the relevant filters are displayed. The filters are displayed as described in “Command `csr get fi` <filter-name>” on page 70 (*-l=h* not set).

-l=h All the relevant filters are displayed. The filters are displayed as described in “Command `csr get fi <filter-name>`” on page 70 (-l=h). This is followed by global statistics concerning all the filters:

global statistic for all filters, sampled since <date>

input direction

output direction

total nr of packets=
<number>

total nr of packets=
<number>

packets denied= <number>

packets denied= <number>

packets permitted=
<number>

packets permitted= <number>

6.6 Object class lsn

The lsn object represents the local subnetwork connections.

The following command is permitted:

csr_get_lsn

Information on the local subnetwork connections is output with the *csr get lsn* command.

Output:

address type

Type of address

CC#/L#

Board and line number in the following format:

W board number / L line number

interface state

Current state of the subnetwork interface

up: Activated

down:

Deactivated

local SNPA

Your own subnetwork address

subnetid

Subnetwork ID

subnet type

Type of subnetwork

6.7 Object class conn

A conn object represents a subnetwork connection.

The following commands are permitted:

csr_g[et]_c[onn] [**t[ype]=|_ref=**] [**-l={|l|m|h}**]

csr_s[et]_c[onn] [**t[ype]=|_ref=**] [**s[tate]=**] [**stati[stics]=**]

type=

Connections of the specified subnetwork type

fr Frame relay

isdn ISDN

pp Direct data connection

pt Analog telephone network

x25 X.25

type= is used as a filter for the *get* action.

ref= Reference of the connection.

ref= is used as a filter for the *get* action.

state=down

The connection is shut down.

statistics=0

The statistical data of the specified connection(s) is deleted.

-l= Controls the extent of output.

Default value: l

l The most important attributes are output in tabular form (“level=low”).

m The described attributes are output (“level=medium”), see table “Attribute outputs for conn objects” on page 75.

h Statistical data is output in addition to the described attributes (“level=high”).

Output:

The attributes output with *get* are listed in the following table:

Attribute	get		
	-l=l	-l=m	-l=h
A/P	x	x	x
CC		x	x
idle		x	x
inmax		x	x
maxslot (l->r)		x	x
maxslot (r->l)		x	x
nlpid		x	x
NSAP (local)	x	x	x
NSAP (remote)	x	x	x
outmax		x	x
ppp		x	x
ref	x	x	x
SNPA (local)		x	x
SNPA (remote)	x	x	x
start/stop		x	x
state	x	x	x
type		x	x
vjhc (l->r)		x	x
vjhc (r->l)		x	x
statistic data			x

Table 5: Attribute outputs for conn objects

A/P Specifies connection creation

A: Active connection setup, i.e. connection setup initiative comes from the local system

P: Passive connection setup, i.e. connection setup initiative comes from the remote system

CC Board and line number in the following format:

W <board_number> / L <line_number>

idle (sec)

Idle time since the last activity on this connection

inmax

Permitted idle time for incoming connections (passive connection setup) after which the connections are to be shut down (in seconds)

unlimited:

No idle time monitoring

maxslot (l->r)

Slot array size with Van-Jacobson header compression in the local → remote direction.

---: No Van-Jacobson header compression

maxslot (r->l)

Slot array size with Van-Jacobson header compression in the remote → local direction.

---: No Van-Jacobson header compression

nlpid Protocol identification (“network layer protocol identification”).

IP: IP protocol

CLNP:

OSI-CLNP protocol

NSAP (local)

Network address of your own system.

NSAP (remote)

Network address of the remote system

outmax

Permitted idle time for outgoing connections (active connection setup) after which the connections are to be shut down (in seconds)

unlimited:

No idle time monitoring

- ppp Point-to-point protocol
- yes: Communication is via the point-to-point protocol.
 - no: Communication is not via the point-to-point protocol.
- ref Reference of the connection
- SNPA (local)
Own subnetwork address
- SNPA (remote)
Remote subnetwork address
- start/stop
Time at which the connection was shut down
- state Current state of the connection
- DATA:
The connection is active with send and receive rights.
- DSTOP:
The connection is active but locked for sending.
- CONRQ:
Wait for connection acceptance by the WAN partner (active setup).
- CONIN:
Wait for connection acceptance by the local system (passive setup).
- type Type of subnetwork
- vjhc (l->r)
- yes: Van-Jacobson header compression is used in the local → remote direction.
 - no: Van-Jacobson header compression is not used in the local → remote direction.
- vjhc (r->l)
- yes: Van-Jacobson header compression is used in the remote → local direction.
 - no: Van-Jacobson header compression is not used in the remote → local direction.

6.8 Diagnostic commands

You can trace the CLW kernel components, the DLPI and NPI interfaces and the internal ADS component for diagnostic purposes.

The trace command *comtr* is described in detail in the manual “CMX, Operation and Administration” [1].

6.8.1 ADS trace

You use the *comtr -mads* command to control the trace for the ADS components. The trace can be started and stopped at any time, and its contents can be output at any time. The following trace levels are supported.

- 2 complete trace
- 4 trace for the interface between the ADS user and the ADS components
- 8 trace for the interface between the ADS components and the RADIUS client daemon (not relevant at present)
- 16 trace for the interface between the ADS components and the FSS

The interface traces can be combined by adding the trace levels together.

Example

Trace level 12 activates the trace between the ADS user and the ADS component

6.8.2 CLW trace

The command *comtr -mclw* can be used to control the CXLW component trace. A trace can be started and stopped at any time, and its contents can be output at any time. The following trace levels are supported. Any higher-level trace level automatically includes the supported trace level.

- 1 Trace of all unusual events
- 2 Trace of connection/disconnection without user data
- 3 Trace of connection/disconnection with user data
- 4 Trace of data transfer without user data
- 5 Trace of data transfer with user data, short output
- 6 Trace of data transfer with user data, long output
- 10 Trace of filter actions

6.8.3 PPP trace

You control the NPI interface trace of the PPP driver with the *comtr -m ppx* command. A trace can be started and stopped at any time and its contents can be also output at any time.

6.8.4 CS-ROUTE traces

The following commands are available for activating, deactivating and querying the status of CS-ROUTE traces (ADS, CLW and PPP traces):

csrtron

activates the CS-ROUTE traces via the *comtr* command with the default trace levels ADS: 2, CLW: 5, PPP: 6.

csrtroff

deactivates the traces.

csrtrstat

provides information on the status of the traces.

You can output the parameters of the commands with *csrtron ?* and *csrtroff ?*.

6.8.4.1 csrtron

csrtron [**-m**<mod>] [**-l**<level>] [**-u**<size>] [**-d**<dir>] [**-f**<file>]

mod Driver(s) for which the trace is to be activated (default : ads clw ppx).

ads, clw, ppx

Default: ads + clw + ppx

level Trace level to be used. If no trace level is specified, the following levels are activated:

ads level 2

clw level 5

ppx level 6

size Size of the kernel memory(Kb) needed by *comtr* to store binary traces.

Default:

CMX < 5.1E : 1000Kb

CMX >= 5.1E : depending on driver init

dir Directory in which the trace files are stored (binary format as well as evaluated text files in ASCII format). A directory is created where necessary/possible.

If no directory is specified, the working directory ('pwd') is used.

file File name of the binary trace file(s).

comtr appends the index '.0[12]'.

Default: comtr.bin_<>.0[12]

Example

```
csrtron -mads -l8 -mppx -l3 -d /var/...
```

This command activates the traces for the ads driver with the trace level set to 8 and the ppx driver with the trace level set to 3. The binary files and ASCII files for the traces are written to the directory */var/...*

6.8.4.2 csrtroff

csrtroff [_-m<mod>] [_-c[_-d<dir>]] [_-e]

mod Driver(s) for which the trace is to be deactivated.

ads, clw, ppx

Default: ads + clw + ppx

-c Deactivate and reactivate trace.

ads, clw, ppx

Default: ads + clw + ppx

There is a short break between deactivation and reactivation.

dir Directory in which the trace files are to be stored (binary format).

This option can only be used in conjunction with *-s*.

-e Edit evaluated text files.

7 FSS commands

The description of an FSS configuration is stored in the Forwarding Support Information Base (FSB) database. The FSB is an object-oriented database and defines a series of object classes with their attributes. When you make an entry in the FSB, you create an object which belongs to a specific object class and assign it attribute values according to this object class.

With standard configurations, you can make the FSB entries via the character-oriented user interface CMXCUI. Configuration examples can be found in the chapter “Examples” on page 109.

The *fssadm* command is used for configuring the Forwarding Support Service (FSS) in expert mode. The *fssadm* command should only be used for special configurations and by persons with comprehensive knowledge of CMX, ISDN and WANs.

FSS and the *fssadm* command are described in detail in the manual “CMX, Operation and Administration” [1]. Only the parameters and values of the object classes which are relevant for CS-ROUTE are described in this chapter.

Syntax

The syntax of the *fssadm* commands shown below for various object classes is described in detail in the section “Notational conventions” on page 3 of this manual.

Actions

When you use the *fssadm* command, you apply specific actions to the object classes and their attributes. Refer to the description of the object class concerned to determine which actions are permitted for a specific object class.

The following actions can be specified in the *fssadm* command:

create

Creates an object with the specified attribute values.

delete

Deletes an object.

set

Sets the attributes of an object to the specified values.

FSS commands

get Retrieves the objects of the specified object class together with their attributes.

If attribute values are specified, *fssadm* only selects objects with these attribute values. *fssadm* rejects the input of multiple attributes that already uniquely identify one object each.

Examples of fssadm commands

```
fssadm create SNPAROUTES name=RX25_CS1 subnet=X25-1\  
           dteaddr=12345  
fssadm create NSAP name=NSIP_CS1 internet-addr=205.75.2.10\  
           snpa-list=RX25_CS1
```

Refer to the man pages for a detailed description of the *fssadm* command.

Uppercase and lowercase with actions, object classes and attributes

fssadm does not differentiate between uppercase and lowercase or between a dash (-) and underscore (_) when naming actions, object classes, attributes and attribute values which are symbolic constants.

Examples of using uppercase and lowercase with fssadm commands

The following three entries are permitted and are equivalent:

```
SubNet=isdN-1  
SUBNET=ISDN_1  
subnet=ISDN-1
```

Uppercase and lowercase with the name attribute

Uppercase and lowercase are differentiated for names which you have to define yourself (object class FACIL, SNPAROUTES, NSAP with the name attribute).

Examples of using uppercase and lowercase with the name attribute

The following three entries designate three different names:

```
name=HOST-1  
name=Host-1  
name=host_1
```

Uppercase and lowercase with `fssadm` information output

In the `fssadm` outputs, the names of actions and attributes are written in lowercase, and the names of object classes and attribute values that are symbolic constants are written in uppercase.

Help functions

You can use the following commands to request help on the `fssadm` syntax:

`fssadm_?`

Displays a general description of the `fssadm` syntax and information on the help function.

`fssadm_action_?`

Displays the object class to which an action may be applied.

`fssadm_action_object class_[[attribute name=]attribute value ...]_?`

Completes the command with the attributes which are suitable for the specified context. The context is only taken into account for the attributes that follow the context in the output.

`fssadm_action_object class_[[attribute name=]attribute value ...]_attribute name=_?`

Shows the syntax of the specified attribute in the given context. Only those attributes in the specified context which precede the queried attribute are taken into account.

Example

The `fssadm create snparoutes type=isdn-nc ?` command returns the following output:

```
fssadm create SNPAROUTES <name> [<subnet>] type=ISDN-NC\
{<remsnpa> <nailed-up-isdn> } (min=0,max=1) [<facil>]
```

The `fssadm create snparoutes subnet=isdn-1 type=?` command returns the following output:

```
<type>: ISDN | ISDN-[ENC] | X31-M[SA] | X31-S[VC] | X31-P[VC]
```

When entering the question mark (?), bear in mind that this character has a special meaning for the shell and may need to be escaped with a backslash (\).

Abbreviated notation

Abbreviated keywords:

You can abbreviate the keywords as long as the commands, actions, object classes and attributes remain unique.

Example

```
fssadm create snparoutes name=XY subnet=isdn-1 isdn-nr=132345
```

Abbreviated notation:

```
fssadm cr sn nam=XY su=isdn-1 is=132345
```

Abbreviations with positional parameters:

Attribute keywords may be omitted, provided the values can be identified uniquely by their setting, format or context.

Example

```
fssadm create snparoutes name=XY subnet=isdn-1 \  
type=isdn isdn-nr=132345
```

Abbreviated notation:

```
fssadm cr sn XY isdn-1 132345
```

Keyword and positional parameters can be combined as required.



Avoid abbreviations and positional parameter notation in shell scripts that are to be used over extended periods. Such commands may be rendered invalid if new object classes, attributes or attribute values are introduced in a product update.

Overview of actions, object classes and attributes

The following table provides an overview of the object classes that may be in the configuration file. You can also use the table to see which actions may be used in the *fssadm* command on a specific object class and which attributes are assigned to the object class.

Actions	Object Class	Attributes
create delete get set	NSAP	name, address, net, type, access, snpa-list, subnet
create delete get set	FACIL	name, short-id, facil, admit, ppp-profile, compress, npid, ppp-accm, ppp-auth- protocol, ppp-auth-params
create delete get set	SNPAROUTES	name, address, subnet, type, short-id, facil
create delete get set	PPPAUTH	name, short-id, loc-id, peer-id, pap-loc-pwd, pap-peer-pwd, chap-loc-secret, chap-peer-secret
create delete get set	SUBNET	subnet, incoming-call, x25-description

Table 6: Actions, object classes and attributes of the *fssadm* command

The object classes that are relevant for CS-ROUTE are described in the following sections.

7.1 Object class FACIL

You can assign facilities to each route with a FACIL object

The following commands are permitted:

fssadm **create** **FACIL**

```
name = [facil=] [in-max-idle=] [out-max-idle=] [admit=]
[ppp-profile=] [compress=] [npid=] [ppp-accm=]
[ppp-auth-protocol=] [ppp-auth-params=]
```

fssadm **delete** **FACIL**

```
{name=|short-id}
```

fssadm **get** **FACIL**

```
[name=] [short-id=] [facil=] [in-max-idle=] [out-max-idle=]
[admit=] [ppp-profile=] [compress=] [npid=] [ppp-accm=]
[ppp-auth-protocol=][ppp-auth-params=]
```

fssadm **set** **FACIL**

```
{name=|short-id=|name= short-id=} [facil=]
[in-max-idle=] [out-max-idle=][admit=] [ppp-profile=]
[compress=] [npid=] [ppp-accm=] [ppp-auth-protocol=]
[ppp-auth-params=]
```

name=

Name of the FACIL object. Mandatory for the *create* action.

The name is used for the SNPAROUTES reference to the FACIL object and as an identifier in the *fssadm* command or *fssadm* outputs.

1-15 characters: letters, digits and the special characters `_` (underscore) and `#` (number sign).

A distinction is made between uppercase and lowercase. The first character must not be a digit or an underscore (`_`).

in-max-idle

Monitoring time, in seconds, of the IDLE status of a connection set up by an incoming call. Once this IDLE time has expired, the connection is closed down.

out-max-idle

Monitoring time, in seconds, of the IDLE status of a connection set up by an outgoing call. Once this IDLE time has expired, the connection is closed down.

admit=

Defines whether or not incoming and/or outgoing calls are permitted. This access protection is desirable at the subnetwork level. A FACIL object with this attribute is only linked with SNPAROUTES objects which represent dial-up connections.

B[OTH_IN_AND_OUT]

Incoming and outgoing calls are permitted (corresponds to the earlier value *YES*).

O[UTGOING_ONLY]

Only outgoing calls are permitted (corresponds to the earlier value *NO* or no entry).

I[NCOMING_ONLY]

Only incoming calls are permitted.

N[EITHER_IN_NOR_OUT]

Neither incoming nor outgoing calls are permitted.

In the case of incoming calls, the setting for the corresponding subnetwork connection is evaluated first (see the attribute *incoming-call* in section "SUBNET object class" on page 104).

compress=

Specifies whether Van-Jacobson header compression is to be used.

T[CP/IP]

Compression is used.

N[O] Compression is not used.

facil=

Name of a defined FACIL object that is referenced.

You can use this referenced FACIL object to define default values or values which are common to several FACIL objects. Values of the referenced FACIL object are overwritten by the values of the referencing FACIL object.

npid=

ID of the network protocol. This attribute defines the Layer 3c entity to which an incoming call is to be forwarded if this is not already defined by the protocol data (e.g. X.25 call user data or ISDN signaling). If the Layer 3c entity is not defined by either protocol data or the FACIL attribute npid, the definition in the CCP configuration applies.

INTERNET

The incoming call is forwarded to the Layer 3c entity INTERNET.

ppp-profile=

Point-to-point protocol usage

If *ppp-accm* is specified, the value GSM is assigned automatically. If another *ppp** attribute is specified without specifying *ppp-accm*, the *ppp-profile* is automatically assigned the value STANDARD.

N[O] The point-to-point protocol is not used.

STANDARD

The point-to-point protocol is used (corresponds to YES in earlier CMX versions).

GSM Use of the point-to-point protocol with asynchronous transmission via ISDN when dialing in via mobile telephone.

ppp-accm=

Asynchronous Control Character Mapping.

Control characters which are not to be interpreted as control characters in the mobile telephone network are masked using Asynchronous Control Character Mapping (ACCM).

You use this parameter to specify which control characters are to be masked when a connection is set up via a mobile telephone. You can specify the control characters using an abbreviation or as a pair of hexadecimal numbers.

ALL_CNTRL_CHARS

All characters are mapped.

NO_MAPPING

No characters are mapped.

The table below provides you with an overview of the control characters and the corresponding hexadecimal values:

Control character	Hex	Meaning
NUL	00	nil (no operation)
SOH	01	start of header
STX	02	start of text
ETX	03	end of text

Control character	Hex	Meaning
EOT	04	end of text
ENQ	05	inquiry
ACK	06	acknowledge
BEL	07	bell
BS	08	backspace
HT	09	horizontal tabulation
LF	0A	line feed
VT	0B	vertical tabulation
FF	0C	form feed
CR	0D	carriage return
SO	0E	shift-out
SI	0F	shift-in
DLE	10	data link escape
DC1	11	device control 1, (XON)
DC2	12	device control 2
DC3	13	device control 3, (XOFF)
DC4	14	device control 4
NAK	15	negative acknowledgment
SYN	16	synchronous idle
ETB	17	end of transmission block
CAN	18	cancel
EM	19	end-of-medium (signal 3)
SUB	1A	substitute character
ESC	1B	ESCAPE
FS	1C	file separator
GS	1D	group separator
RS	1E	record separator
US	1F	unit separator

If you specify more than one value, you must separate the values with a plus (+) or minus (-) sign and enclose them in quotation marks (").

+ in front of a value means: mask control characters

- in front of a value means: do not mask control characters

To make things easier to read, blanks and newline control characters may be used before and/or after the plus or minus signs (+ or -). In this case, the entire expression must be enclosed in quotation marks ("), e.g. `ppp-accm="0x11 + 0x12"` or `ppp-accm="ALL_CNTRL_CHARS - FF"`.

`ppp-accm` needs to be specified only if the user dials in via a mobile telephone. In the mobile telephone network of Deutsche Telekom, you must specify `ppp-accm=DC1+DC3`.

ppp-auth-protocol=

Authentication protocol

N[O] No authentication protocol is requested actively.

PAP Authentication procedure in which an unencrypted password is transmitted.

CHAP

Authentication procedure in which the password is known only to the two partners and is not transmitted over the public network.

ppp-auth-params=

Name of a PPPAUTH object. This parameter is used to reference an authentication parameter (PAP/CHAP parameter) in the PPPAUTH object.

short-id=

Automatically assigned short ID (decimal number) of the FACIL object. It can be used with the *set*, *get* and *delete* commands to identify the object in diagnostics, i.e. in trace entries.

7.2 Object class NSAP

Every end or transmission system for which transport connections are to be set up or via which data is to be transferred is represented by an NSAP object.

The following commands are permitted:

fssadm_create_NSAP

name= internet-addr= [net=] [access=] snpa-list=

fssadm_delete_NSAP

{name=|internet-addr=}

fssadm_get_NSAP

[name=] [internet-addr=] [net=] [access=] [subnet=] [type=]
[snpa-list=]

fssadm_set_NSAP

{name=|internet-addr=|name= internet-addr=} [net=]
[access=][snpa-list=]

name=

Name of the NSAP object

1-32 printable, visible characters

internet-addr=

Internet address of the end system or the transmitting system in the form:

number.number.number.number

number

Possible value: 0...255

Only syntactically valid Internet addresses are permitted, i.e. not 25.0.0.0 or 139.11.255.255.

subnet=

Subnetwork ID. This parameter can be used as a filter with the *get* action and refers to SUBNID in the KOGS XSNID macro or SNID in *clw.cc.conf* file.

This ID consists of a prefix, which defines the subnetwork type, and a subsequent number.

Only the NSAP objects that are assigned routes with the specified subnetwork ID are displayed, and only the corresponding routes are included in the *snpa-list* attribute. The presence of additional routes with other subnetwork IDs is indicated with “+”.

X25-n n=1...32

For X.25 subnetwork connections via dedicated (leased) lines or ISDN permanent connections

PT-n n=1...32

For analog telephone lines (also with X.32 dialing)

FR-n n=1...128

For frame relay subnetwork access

PP-n n=1...32

For point-to-point permanent connections (leased lines) without X.25

ISDN-n n=1...32

For subnetwork access via ISDN dial-up connections (including X.25 via ISDN dial-up connections) or ISDN permanent connections without X.25

type=

SNPA address type

This parameter can be used as a filter with the *get* action and refers to the NSAP objects that are assigned routes with the specified SNPA address type.

FR-PVC

Frame relay PVC

PP Leased line**PT** Analog telephone line**PVC** X.25 PVC

X25 X.25 SVC

ISDN

ISDN dial-up connection

ISDN-N[C]

ISDN permanent connection

X31-M[SA]

X.25 connection with minimum integration according to X.31case A

X31-S[VC]

Switched virtual call X.25 connection (SVC) with maximum integration according to X.31case B

X31-P[VC]

Permanent virtual circuit X.25 connection (PVC) with maximum integration according to X.31case B

X32-PTMSA

X.25 connection with X.32 dialing

net= Network protocol used by the local system to reach the NSAP.

Only needs to be specified if the *get* action has been defined as a filter criterion.

I[INTERNET]

The Internet protocol is used.

access=

Specifies how the subnetwork addresses required for the route to the partner system are to be determined.

This entry is only meaningful as a filtering criterion with the *get* action.

DIRECT

The *snpa-list* attribute exists.

snpa-list=

List of SNPAROUTES objects that can alternatively be used to reach the NSAP. The objects are entered in the following format:

snpa [/weight] [+snpa [/weight]]...

snpa Name of an SNPAROUTES object

1-15 characters: letters, digits, and the special characters _ (underscore) and # (number sign).

A distinction is made between uppercase and lowercase. The first character must not be a digit or underscore (_).

weight

Specifies a priority for the routes in the list.

Possible value: 1...20

20 is the highest priority. The SNPAROUTES object specified in the list with *snpa/20* is used as the first alternative route.

Blanks and newline characters may be added before and/or after the plus character (+) to improve legibility. The complete expression must then be enclosed in quotes, e.g. `snpa-list="route1 + route2"`.

The list can contain a maximum of 20 entries.

+snpa [/weight]

Only for the *set* action:

The specified SNPAROUTES object is inserted into the list or the priority of an existing object is modified,

e.g. `fssadm set NSAP name=N1 snpa-list=+R3/5`.

-snpa

Only for the *set* action:

The specified SNPAROUTES object is deleted from the list, e.g.

`fssadm set NSAP name=N1 snpa-list=+R3/5-R4+R5/4`.

snpa-list may not be specified together with *hop-nsap*.

7.3 Object class SNPAROUTES

An SNPAROUTES object represents a group of similar routes. Similar routes have a common end point, the remote subnetwork connection. In addition, the starting points of similar routes, the local subnetwork connections, have a common subnetwork ID.

Such a group of similar routes is itself referred to in simplified form as a route.

The parameters and parameter values described in the following section are important to CS-ROUTE.

The following commands are permitted:

fssadm_create_SNPAROUTES

name= subnet= [**type=**] **address** [**facil=**]

fssadm_delete_SNPAROUTES

{**name=**|**short-id=**|**subnet=** [**type=**] [**address**]}

fssadm_get_SNPAROUTES

[**name=**] [**short-id=**] [**subnet=**] [**type=**]
[**address**] [**facil=**]

fssadm_set_SNPAROUTES

{**name=**|**short-id=**} [**subnet=**] [**type=**]
[**address**] [**facil=**]

name=

Name of the SNPAROUTES object. The name is required for NSAP object reference to this SNPAROUTES object and as an identifier in *fssadm* commands or *fssadm* outputs.

1-15 characters: letters, digits and the special characters _ (underscore) and # (number sign).

A distinction is made between uppercase and lowercase. The first character must not be a digit or an underscore (_).

subnet=

subnetwork ID of the subnetwork connections belonging to this route (or group of routes). The ID consists of a prefix specifying the subnetwork type and a subsequent number.

- X25-n** n=1...32
For X.25 subnetwork connections via leased lines or ISDN permanent connections
- PT-n** n=1...32
For analog telephone lines (also with X.32 dialing)
- FR-n** n=1...128
For frame relay subnetwork access
- PP-n** n=1...32
For point-to-point permanent connections (leased lines) without X.25
- ISDN-n** n=1...32
For subnetwork access via ISDN dial-up connections (including X.25 via ISDN dial-up connections) or ISDN permanent connections

type=

SNPA address type

The value is determined implicitly from the subsequent address and can therefore be omitted with the *create* or *set* actions.

For the address types *PP* and *ISDN-NC*, the address attribute is not evaluated for addressing and is therefore optional. It is only meaningful to create such an SNPAROUTES object if address type-specific default FACIL attributes are to be defined.

FR-PVC

Frame relay PVC

(corresponding address attribute: *fr-pvc=*)

PP Leased line

(corresponding address attribute: *line-nr=*)

PT Analog telephone line

(corresponding address attribute: *phone-nr=*)

PVC X.25 PVC

(corresponding address attribute: *pvc-nr=*)

X25 X.25 dial-up line

(corresponding address attribute: *dte-addr=*)

ISDN ISDN dial-up connection

(corresponding address attribute: *isdn-nr=*)

ISDN-N[C]

ISDN permanent connection
(corresponding address attribute: *nailed-up-isdn=*)

X31-M[SA]

X.25 connection with minimum integration according to
X.31 case A
(corresponding address attribute: *x31-msa=*)

X31-S[VC]

Switched virtual call X.25 connection (SVC) with maximum
integration according to X.31 case B
(corresponding address attribute: *x31-dte-adr=*)

X31-P[VC]

Permanent virtual circuit X.25 connection (PVC) with maximum
integration according to X.31 case B
(corresponding address attribute: *x31-pvc-nr=*)

X32-PTMSA

X.25 connection with X.32 dialing
(corresponding address attribute: *x32-phone-nr=*)

address

Address of the remote subnetwork access. One of the following parameters must be specified for *address*:

phone-nr=

Telephone number of the partner.

1-24 printable characters, enclosed within single quotes (')

phone-nr is evaluated only if AUTO/ABG or AUTO/ANK is specified for the RUF AUTO parameter in the KOGS XLTNG macro.

line-nr=

CC and line number for a leased line, in the form:

[cc number/] line number

Entries correspond to the TNS entry

“WAN [cc number:]line number“.

cc number

Number of the CC: 1...256

line number

Single-digit line number: 1...4

The *line-nr* attribute is optional. It is not used for CC or line selection for outgoing data. Only the subnetwork ID is relevant for this selection.

dte-addr=

DTE address of the partner: 1-17 digits

pvc-nr=

X.25 PVC number and optionally your own DTE address, in the form:

pvc number [/dte address]

pvc number

Possible value: 0...4095

dte address

1-17 digits

fr-pvc=

CC number, line number and frame relay PVC number, in the form:

cc number/line number/pvc number

cc number

Possible value: 1...256

line number

The line number must correspond to the decimal value of the KOGS LPUFADR parameter. The value must be specified in decimal in the FSB and in hexadecimal in the KOGS.

Possible value: 1...4

pvc number

Possible value: 16...1007

x31-msa=

Two-step dialing address with minimum integration according to X.31 case A, in the form:

isdn number/x25 dte address

isdn number

ISDN number of the access unit (first step): 1-20 digits

x25 dte addr

DTE address of the partner (second step): 1-17 digits

x31-dte-addr=

Address with maximum integration according to X.31 case B, if the partner is reached via X.25 SVC, in the form:

x25 dte address rem [/x25 dte address local]

x25 dte address rem

DTE address of the partner: 1-17 digits

x25 dte address local

Own DTE address: 1-17 digits (optional)

x31-pvc-nr=

Address with maximum integration according to X.31 case B, if the partner is reached via X.25 PVC, in the form:

number [/x25 dte address local]

number

PVC number of the partner

Possible value: 0...4095

x25 dte address local

Own DTE address: 1-17 digits

x32-phone-nr=

Two-step dialing address with X.32 dialing, in the form:

telephone number/dte address

telephone number

Telephone number of the access unit (X.25 exchange)
(first step)

1-24 printable characters, enclosed within single quotes (')

dte address

DTE address of the partner (second step): 1-17 digits

isdn-nr=

ISDN subscriber number of the partner (including the area code):
1-20 digits

nailed-up-isdn=

CC and line number for an ISDN permanent connection, in the
form:

cc number/line number

cc number

Number of the communication controller: 1...32

line number

Line number: 0, 1, 2, 32, 33, 34.

The *nailed-up-isdn* attribute is optional. It is not used for CC or line selection for outgoing data. Only the subnetwork ID is relevant for this selection.

Meaning:

0	1st. ISDN connection	D channel
1	1st. ISDN connection	1st. B channel
2	1st. ISDN connection	2nd. B channel
32	2nd. ISDN connection	D channel
33	2nd. ISDN connection	1st B channel
34	2nd. ISDN connection	2nd B channel

facil=

Refers to a FACIL object that defines the facilities which apply to the route represented by this SNPAROUTES object.

Name of a defined FACIL object

1-15 characters: letters, digits and the special characters _ (underscore) and # (number sign).

A distinction is made between uppercase and lowercase. The first character must not be a digit or an underscore (_).

Example

```
FACIL name=VJHC compress=TCP/IP
FACIL name=IN admit=INCOMING-ONLY facil=VJHC
FACIL name=OUT admit=OUTGOING-ONLY facil=VJHC
SNPAROUTES name=R1 subnet=ISDN-1 isdn-nr=6301997 facil=IN
SNPAROUTES name=R2 subnet=ISDN-2 isdn-nr=6301997 facil=OUT
```

short-id=

Automatically assigned short ID (decimal number) of the SNPAROUTES object. It can be used with the *set*, *get* and *delete* commands and is used to identify the object in diagnostics, i.e. in trace entries.

7.4 SUBNET object class

Objects of the SUBNET class represent a local subnetwork connection, which is uniquely identified by a subnetwork ID, or a group of similar local subnetwork connections, which is identified by the subnetwork ID that is common to these connections (subnet attribute).

The object is assigned values which are required to set X.25 minimum integration for calls with unknown ISDN partners or X.32 dialing for telephone calls, and to activate and deactivate access protection.

fssadm create SUBNET

```
subnet=[incoming-call=][x25-description=]  
[osi-nsap-address=]
```

subnet=

Subnetwork ID

ISDN-i

i=1...32

For subnetwork access via ISDN dial-up connections (with or without the X.25 protocol) or via ISDN permanent connections (without the X.25 protocol)

X25-i

i=1...32

For subnetwork access via ISDN permanent connections with the X.25 protocol

PT-i

i=1...32

For subnetwork access via telephone connections

X21-i

i=1...32

For subnetwork access via X.21 dial-up connections

[incoming-call=]

Together with the *admit* attribute, this attribute provides the configuration function for access protection in CCP-ISDN. It acts as a switch for activating and deactivating access checks (temporarily or permanently).

This is only relevant to dial-up connections (ISDN dial-up connections and X.25-SVC), i.e. for the connection types ISDN, X31-MSA, X.25, and X31-SVC.

NONE

All incoming connection requests are rejected. Any *admit* attribute configured for the calling address is ignored. The subnetwork address test is deactivated.

RESTRICTED

Incoming connection requests are accepted only if an incoming call is configured as permissible for the calling address, i.e. if the corresponding SNPAROUTES object is assigned a FACIL object that has the attribute *admit=BOTH_IN_AND_OUT* or *admit=INCOMING_ONLY*. The subnetwork address test is activated.

ALL All incoming connection requests are processed. Any *admit* attribute configured for the calling address is ignored. The subnet address test is deactivated.

If there is no *incoming-call* attribute, then access control is not activated. If you are using the product CS-ROUTE then incoming calls may or may not be admitted depending on the *admit* parameter of the corresponding route.

[x25-description=]

The *x25-description* attribute is permitted only for ISDN and PT subnetwork IDs.

It refers to the predefined description of the X.25 access in the CC configuration file (DTE name).

This attribute selects the specified description of the X.25 access for ISDN calls from unknown partners or incoming telephone calls.

The value must match that of the *DTE_Name* operand in the corresponding CC configuration file.

Possible values:

1...8 characters: letters, digits and the special characters \$, # and @. The first character must not be a digit. No distinction is made between uppercase and lowercase.



For ISDN calls from known partners, X.25 minimum integration is set by configuring an SNPAROUTES object of the type X31-MSA which corresponds to the calling ISDN number. The selection of the X.25 access description is configured only by assigning a FACIL object with the attribute *x25-description* to this SNPAROUTES object. The SUBNET attribute *x25-description* therefore has no effect on known partners.

[osi-nsap-address=]

OSI address in Reference Publication Format in accordance with IS 8348 Add2

The attribute is only permissible in the case of objects with the subnetwork X25-n (n = 1, 2, ..., 32) and can only be specified with the *fssadm* commands *create*, *get* and *set*.

All syntactically correct OSI-NSAP addresses are accepted. The addresses do not have to be unique: the same OSI-NSAP address may be repeated for different SUBNET objects and/or NSAP objects and/or the LOCNSAP object.

7.5 Object class PPPAUTH

A PPPAUTH object is required if you want to configure an authentication check for communication via TCP/IP using PPP (point-to-point protocol). This is done by specifying the appropriate details for access protection via PAP (Password Authentication Protocol) and CHAP (Challenged Handshake Authentication Protocol).

The following commands are permitted:

fssadm create PPPAUTH

```

_name=[ loc-id=] [_peer-id=] [_pap-loc-pwd=]
[_pap-peer-pwd=] [_chap-loc-secret=] [_chap-peer-secret=]

```

fssadm delete PPPAUTH

```

_name=|_short-id=

```

fssadm get PPPAUTH

```

[_name=][_short-id=] [_loc-id=] [_peer-id=] [_pap-loc-pwd=]
[_pap-peer-pwd=] [_chap-loc-secret=] [_chap-peer-secret=]

```

fssadm set PPPAUTH

```

{[_name=|_short-id=] [_name=] [_loc-id=] [_peer-id=]
[_pap-loc-pwd=] [_pap-peer-pwd=] [_chap-loc-secret=]
[_chap-peer-secret=]}

```

name=

Name of the PPPAUTH object. Mandatory for the *create* action.

The name is required for FACIL object reference to the PPPAUTH object and as an identifier in *fssadm* commands or *fssadm* outputs.

1-15 characters: letters, digits and the special characters `_` (underscore) and `#` (number sign).

No distinction is made between uppercase and lowercase. The first character must not be a digit or an underscore (`_`).

short-id=

Automatically assigned short ID (decimal number) of the PPPAUTH object. It can be used with the *set*, *get* and *delete* commands and to identify the object in diagnostics, i.e. in trace entries.

loc-id=

Local ID of the PPPAUTH object.

1- 32 printable visible characters.

A distinction is made between uppercase and lowercase letters.

peer-id=

Partner ID.

1- 32 printable visible characters.

A distinction is made between uppercase and lowercase letters.

pap-loc-pwd=

PAP password for the local system.

1- 32 printable visible characters.

A distinction is made between uppercase and lowercase letters.

The password must be stored in a file as plain text, and the name of this file must be specified as a parameter.

pap-peer-pwd=

PAP password for the partner system. Name of a file in which the password is stored in plaintext.

1- 32 printable visible characters.

A distinction is made between uppercase and lowercase letters.

The password must be stored in a file as plain text, and the name of this file must be specified as a parameter.

chap-loc-secret=

CHAP password for the local system.

1- 255 printable visible characters.

A distinction is made between uppercase and lowercase letters.

The password must be stored in a file as plain text, and the name of this file must be specified as a parameter.

chap-peer-secret

CHAP password for the partner system. Name of a file in which the CHAP password is stored in plain text.

1- 255 printable visible characters.

A distinction is made between uppercase and lowercase letters.



The file in which a CHAP or PAP password is stored is only needed temporarily, i.e. only at the time when an *fssadm* command that references it is issued. The information is then encrypted and stored in the FSB. The file should be deleted after the *fssadm* command is executed.

8 Examples

This chapter starts with two examples that illustrate the basic principles for configurations using TCP/IP.

These detailed descriptions relate to ISDN. You will then find further examples showing configuration extensions for use with different subnetwork types and network connections.

The relevant commands are shown for each example for the sake of clarity.

8.1 TCP/IP via ISDN

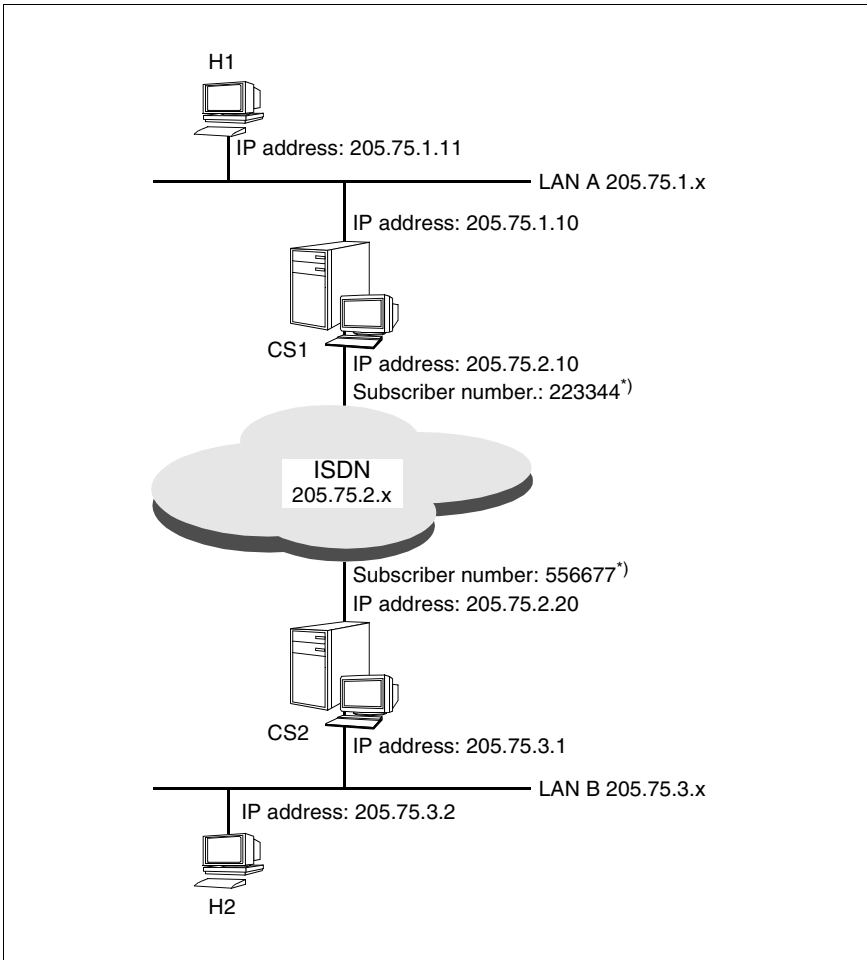


Figure 19: TCP/IP via ISDN (dial-up connection)

*) Please note that the general conventions must be observed when specifying the subscriber numbers in the FSS.

For partner systems in the same local network, only the call number needs to be specified (e.g. 223344); for partner systems outside the local area, you will also need to enter the appropriate country and area codes (e.g. 089223344 for partners in Munich and 00431223344 for partners in Vienna, Austria.).

H1 and H2 are Solaris systems in TCP/IP LANs. CS1 and CS2 are Solaris systems with CS-ROUTE. The LANs are connected via an ISDN dial-up line. The point-to-point protocol is used.

For H1 and H2, you must only configure IP routes to all partners not in the same network/subnetwork. Note that the described procedure for Solaris systems applies.

For C1 and C2, you must configure IP routes to the partner systems. You must also configure the WAN partner systems and their reachability in the FSS.

Configuration data required for H1:

- IP address of the LAN connection: 205.75.1.11
- Route entry for WAN partners:
Enter CS1 as the default router with the command

```
route add net 205.75.2.0 205.75.1.10 1
```
- Route entry for LAN B partners:
Enter CS1 as the default router with the command

```
route add net 205.75.3.0 205.75.1.10 1
```

Configuration data required for CS1:

- IP address of the LAN connection: 205.75.1.10
- IP interface (IP address of the WAN connection):

```
csr create if name=clwipl ipaddr=205.75.2.10  
snid-list=ISDN-1
```
- Route command for LAN B partners in the file specified with the *csr create if...* command

```
route add net 205.75.3.0 205.75.2.20 1
```
- Subnetwork interface:

```
csr create net type=ISDN
```

- FSS configuration:

```
fssadm create FACIL name=cs_fac ppp_profile=YES
fssadm create SNPAROUTES name=r_cs1_cs2 subnet=ISDN-1 \
    isdn-nr=556677 facil=cs_fac
fssadm create NSAP name=CS2 internet-addr=205.75.2.20 \
    snpa-list=r_cs1_cs2
```

Configuration data required for CS2:

- IP address of the LAN connection: 205.75.3.1

- IP interface (IP address of the WAN connection):

```
csr create if name=clwip1 ipaddr=205.75.2.20
    snid-list=ISDN-1
```

- Route command for LAN A partners in the file specified with the *csr create if...* command

```
route add net 205.75.1.0 205.75.2.10 1
```

- Subnetwork interface:

```
csr create net type=ISDN
```

- FSS configuration:

```
fssadm create FACIL name=cs_fac ppp_profile=YES
fssadm create SNPAROUTES name=r_cs2_cs1 subnet=ISDN-1 \
    isdn-nr=223344 facil=cs_fac
fssadm create NSAP CS1 internet-addr=205.75.2.10 \
    snpa-list=r_cs2_cs1
```

Configuration data required for H2:

- IP address of the LAN connection: 205.75.3.2

- Route entry for a WAN partner:

Enter CS2 as the default router with the command:

```
route add net 205.75.2.0 205.75.3.1 1
```

- Route entry for a LAN A partner:

Enter CS2 as the default router with the command:

```
route add net 205.75.1.0 205.75.3.1 1
```


8.1.1 Configuring an ISDN permanent connection

Apart from the configuration of SNPAROUTES objects for CS1 and CS2, the configuration for ISDN permanent connections is the same as for ISDN dial-up connections (see figure 19 on page 110). However, if the point-to-point protocol is not used, you must configure a FACIL object with the attribute *npid=INTERNET*.

In the case of an ISDN permanent connection, the subnetwork ID is sufficient for identification purposes.

SNPAROUTES object for CS1:

```
fssadm create SNPAROUTES name=r_cs1_cs2 type=ISDN-NC \  
                subnet=ISDN-1
```

SNPAROUTES object for CS2:

```
fssadm create SNPAROUTES name=r_cs2_cs1 type=ISDN-NC \  
                subnet=ISDN-1
```

8.1.2 Van-Jacobson header compression

The CS1 and CS2 routers use Van-Jacobson header compression as well as the point-to-point protocol.

The use of Van-Jacobson header compression is configured by assigning a FACIL object to the route concerned with the *compress=TCP/IP* attribute.

Add the *compress* attribute to the FACIL object for CS1.

FACIL object for CS1:

```
fssadm create FACIL name=cs_fac ppp_profile=YES compress=TCP/IP
```

8.1.3 Access protection at user level

PAP as the authentication procedure

In the following example, the system CS1 requests a PAP identification from the system CS2.

Configuration of CS1:

```
fssadm create PPPAUTH name=authpap peer-id=CS2 \
    pap-peer-pwd=/var/tmp/remote
```

where the file */var/tmp/remote* contains the password.

```
fssadm create FACIL name=FAC_PAP ppp-auth-protocol=PAP \
    ppp-auth-params=authpap
fssadm create SNPAROUTES name=r_cs2_cs1 subnet=ISDN-1 \
    type=ISDN isdn-nr=556677 facil=FAC_PAP
fssadm create NSAP name=test internet-addr=201.75.2.20 \
    snpa-list=r_cs2_cs1
```

Configuration of the partner system CS2:

```
fssadm create PPPAUTH name=authpap loc-id=CS2 \
    pap-loc-pwd=/var/tmp/local
```

where the file */var/tmp/local* contains the password. The contents of this file must match the contents of the file */var/tmp/remote* on system CS1.

```
fssadm create FACIL name=FAC_PAP ppp-auth-params=authpap
fssadm create SNPAROUTES name=r_cs2_cs1 subnet=ISDN-1 \
    type=ISDN isdn-nr=223344 facil=FAC_PAP
fssadm create NSAP name=test internet-addr=201.75.2.10 \
    snpa-list=r_cs2_cs1
```

CHAP as a one-sided authentication procedure

In the following example, the system CS1 requests a CHAP identification from the system CS2.

Configuration of CS1:

```
fssadm create PPPAUTH name=authchap loc-id=CS1 peer-id=CS2 \
    chap-peer-secret=/var/tmp/secret
```

where the file */var/tmp/secret* contains the password.

```
fssadm create FACIL name=FAC_CHAP ppp-auth-protocol=CHAP \
  ppp-auth-params=authchap
fssadm create SNPAROUTES name=r_cs2_cs1 subnet=ISDN-1 type=ISDN \
  isdn-nr=556677 facil=FAC_CHAP
fssadm create NSAP name=test internet-addr=201.75.2.20 \
  snpa-list=r_cs2_cs1
```

Configuration of the partner system CS2:

```
fssadm create PPPAUTH name=authchap loc-id=CS2 peer-id=CS1 \
  chap-loc-secret=/var/tmp/secret
```

where the file */var/tmp/secret* contains the password. The contents of the file */var/tmp/secret* must match the contents of the password file on the system CS1.

```
fssadm create FACIL name=FAC_CHAP ppp-auth-params=authchap
fssadm create SNPAROUTES name=r_cs2_cs1 subnet=ISDN-1 \
  type=ISDN isdn-nr=223344 facil=FAC_CHAP
fssadm create NSAP name=test internet-addr=201.75.2.10 \
  snpa-list=r_cs2_cs1
```

CHAP as a mutual authentication procedure

In the following example, the systems CS1 and CS2 request a CHAP identification from each other.

Configuration of CS1:

```
fssadm create PPPAUTH name=authchap loc-id=CS1 \
  peer-id=CS2 chap-loc-secret=/var/tmp/secret1 \
  chap-peer-secret=/var/tmp/secret2
```

where the file */var/tmp/secret1* contains the password for system CS1, and the file */var/tmp/secret2* contains the password for system CS2.

```
fssadm create FACIL name=FAC_CHAP ppp-auth-protocol=CHAP \
  ppp-auth-params=authchap
fssadm create SNPAROUTES name=r_cs2_cs1 subnet=ISDN-1 \
  type=ISDN isdn-nr=556677 facil=FAC_CHAP
fssadm create NSAP name=test internet-addr=201.75.2.20 \
  snpa-list=r_cs2_cs1
```

Configuration of the partner system CS2:

```
fssadm create PPPAUTH name=authchap loc-id=CS2 peer-id=CS1 \  
    chap-loc-secret=/var/tmp/secret2 \  
    chap-peer-secret=/var/tmp/secret1
```

where the file */var/tmp/secret1* contains the password for system CS1, and the file */var/tmp/secret2* contains the password for system CS2. The contents of the files must match the contents of the files on system CS1. It is not, however, necessary to specify the same path and file names.

```
fssadm create FACIL name=FAC_CHAP ppp-auth-protocol=CHAP \  
    ppp-auth-params=authchap  
fssadm create SNPAROUTES name=r_cs2_cs1 subnet=ISDN-1 \  
    type=ISDN isdn-nr=223344 facil=FAC_CHAP  
fssadm create NSAP name=test internet-addr=201.75.2.10 \  
    snpa-list=r_cs2_cs1
```



It is also possible for incoming and outgoing calls to use different CHAP passwords. However, this only functions if the partner system also supports RFC1334.

Do not forget to remove the password files */var/tmp* after configuration.

8.1.4 Configuration for PBXs

If the Solaris system with CS-ROUTE is connected to the ISDN network via a PBX, you must take the following into account during configuration: Example of “0 for obtaining an outside line”:

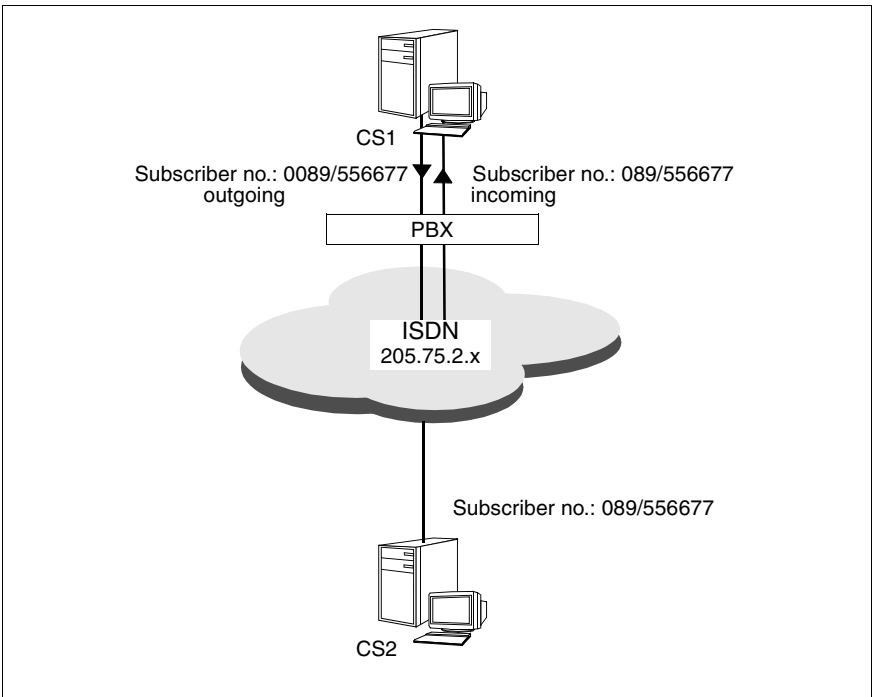


Figure 20: Subscriber numbers with PBXs

If the PBX is configured asymmetrically, the digits for obtaining an outside line (e.g. 0) are not included in the subscriber number of the incoming call.

This asymmetry must be taken into account by configuring the FSS so that different SNPAROUTES objects are configured for each outgoing and incoming call.

You must therefore configure one SNPAROUTES object each for incoming calls (without leading zero) and one SNPAROUTES object each for outgoing calls (with leading zero) for CS1. You must also enter a reference to the two objects in the *snpa-list* attribute of the NSAP object. You define the access protection in the assigned FACIL objects.

FSS configuration for CS1:

```
fssadm create FACIL name=fac_out admit=OUTGOING_ONLY
fssadm create FACIL name=fac_in admit=INCOMING_ONLY
fssadm create SNPAROUTES name=r_cs1_cs2out subnet=ISDN-1 \
    isdn-nr=0089556677 facil=fac_out
fssadm create SNPAROUTES name=r_cs2_cs1in subnet=ISDN-1 \
    isdn-nr=089556677 facil=fac_in
fssadm create NSAP name=CS2 internet-addr=205.75.2.20 \
    snpa-list="r_cs1_cs2out + r_cs2_cs1in"
```

8.1.5 Configuration for asynchronous PPP

You configure asynchronous PPP by specifying the *ppp-profile=GSM* parameter. The *ppp-accm* parameter allows you to specify special characters which are to be transmitted transparently. If this parameter is omitted, it is assumed that all special characters are to be transmitted transparently. We recommend that you specify the characters DC1+DC3, since these are used by the ISDN - D1/D2 gateway for flow control.

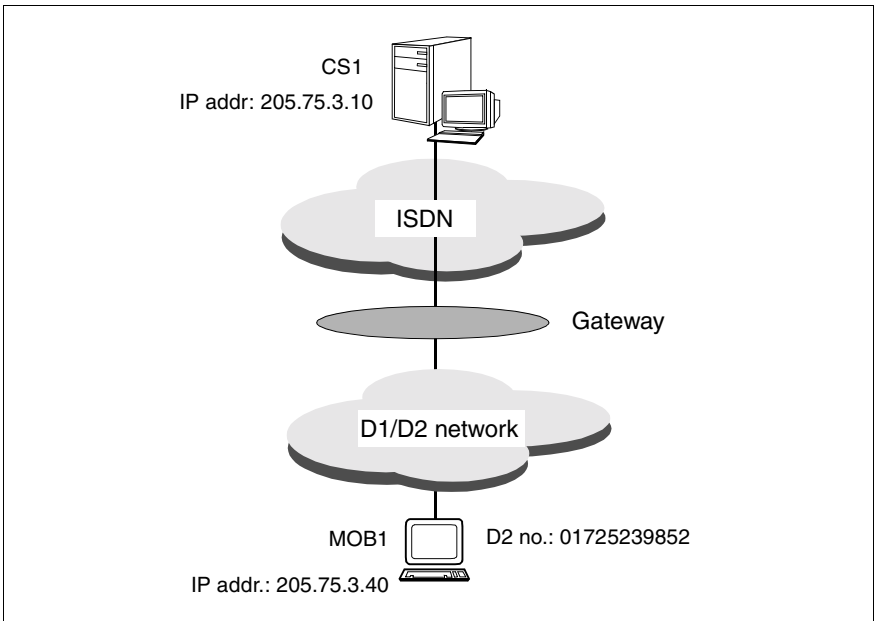


Figure 21: Connection to mobile subscribers (GSM)

FSS configuration for CS1:

```
fssadm create FACIL name=FAC_PPP_ASYNC ppp-profile=GSM \
  ppp-accm=DC1+DC3
fssadm create SNPAROUTES name=con subnet=ISDN-1 \
  isdn-nr=01725239850 facil=FAC_PPP_ASYNC
fssadm create NSAP name=MOB1 internet-addr=205.75.3.40 \
  snpa-list=con
```

8.1.6 Configuring PPP in X.25

It is possible to link CS-ROUTE via X.25. In order to do this, PPP must be supported in X.25 in accordance with RFC1598.

You configure PPP in X.25 by assigning a FACIL object to an X.25 route (e.g. SNPAROUTES ... *subnet=x25-1* ...) with the entry *ppp-profil=STANDARD*. The entry *ppp-profil=GSM* is not allowed for PPP in X.25.

For details on entries in the Forwarding Support Service (FSS), see the manual "CMX/CCP, WAN Communication" [3].

Entry for authentication:

```
fssadm create PPPAUTH name=PNT loc-id=SolarisRechner \  
    peer-id=mueller chap-peer-secret=/tmp/passwd-file \  
    chap-loc-secret=/tmp/passwd-file
```

The local and remote systems use the same password.

Entry for facilities:

```
fssadm create FACIL name=FNT ppp-profil=STANDARD \  
    ppp-auth-params=PNT compress=TCP/IP \  
    npid=INTERNET ppp-auth-protocol=CHAP
```

This sets Van Jacobsen header compression, the network protocol, PPP protocol and authentication using CHAP with the MD5 encryption algorithm.

Entry for the routes:

```
fssadm create SNPAROUTES name=WNT subnet=x25-32 \  
    dte-addr=123456789 facil=FNT
```

Entry for remote host (contains the IP address of the partner system):

```
fssadm create NSAP name=Ntrechner internetaddr=129.186.22.1 \  
    net=internet access=DIRECT snpa-list=WNT
```


8.2 TCP/IP via X.25

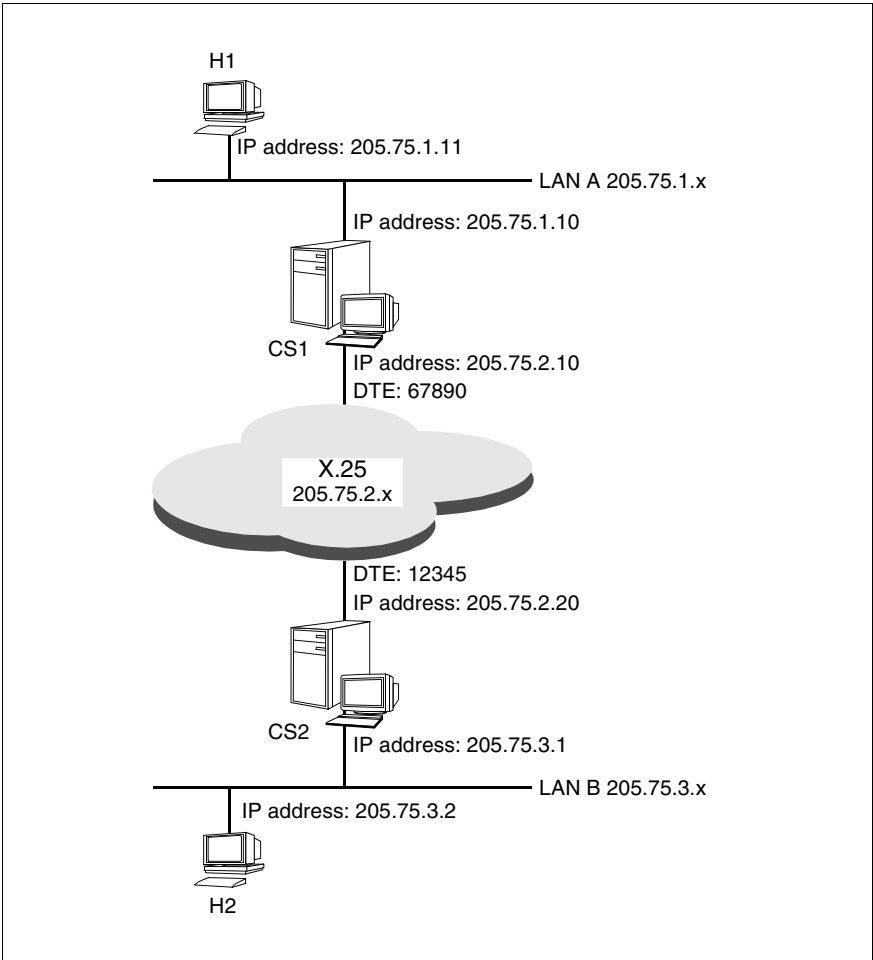


Figure 22: TCP/IP via X.25 (SVC)

H1 and H2 are end systems in TCP/IP LANs. CS1 and CS2 are Solaris systems with CS-ROUTE. The LANs are connected via SVC.

The IP route entries are the same as in the example “TCP/IP via ISDN” on page 110. You must note the following differences for CS1 and CS2 when configuring the IP interface, subnetwork interface and FSS.

Configuration data for CS1:

- IP interface (IP address of the WAN connection):

```
csr create if name=clwip1 ipaddr=205.75.2.10
          snid-list=X25-1
```

- Subnetwork interface:

```
csr create net type=X25
```

- FSS configuration:

The *npid* attribute in the FACIL object is not required.

```
fssadm create SNPAROUTES name=r_cs1_cs2 subnet=X25-1 \
             dte-addr=12345
fssadm create NSAP name=CS2 internet-addr=205.75.2.20 \
             snpa-list=r_cs1_cs2
```

Configuration data for CS2:

- IP interface (IP address of the WAN connection):

```
csr create if name=clwip1 ipaddr=205.75.2.20
          snid-list=X25-1
```

- Subnetwork interface:

```
csr create net type=X25
```

- FSS configuration:

The *npid* attribute in the FACIL object is not required.

```
fssadm create SNPAROUTES name=r_cs2_cs1 subnet=X25-1 \
             dte-addr=67890
fssadm create NSAP name=CS1 internet-addr=205.75.2.10 \
             snpa-list=r_cs2_cs1
```

8.2.1 Configuring an X.25 PVC

Apart from the configuration of FACIL and SNPAROUTES objects for CS1 and CS2, the configuration for X.25 PVCs is the same as for X.25 SVCs (see figure “TCP/IP via X.25 (SVC)” on page 121).

An X.25 PVC is defined via the PVC number and relevant local DTE address. With X.25 PVCs, you must specify the *pvc-nr* attribute in the SNPAROUTES objects for CS1 and CS2 with PVC number and local DTE address instead of the *dte-addr* attribute.

In the FACIL object you must use the *npid* attribute to specify the network protocol used, since there is no connection setup phase for a PVC.

SNPAROUTES object for CS1:

```
fssadm create FACIL name=cs_fac npid=INTERNET
fssadm create SNPAROUTES name=r_cs1_cs2 subnet=X25-1 \
    pvc-nr=8/67890 facil=cs_fac
```

SNPAROUTES object for CS2:

```
fssadm create FACIL name=cs_fac npid=INTERNET
fssadm create SNPAROUTES name=r_cs2_cs1 subnet=X25-1 \
    pvc-nr=243/12345 facil=cs_fac
```

8.3 TCP/IP via frame relay

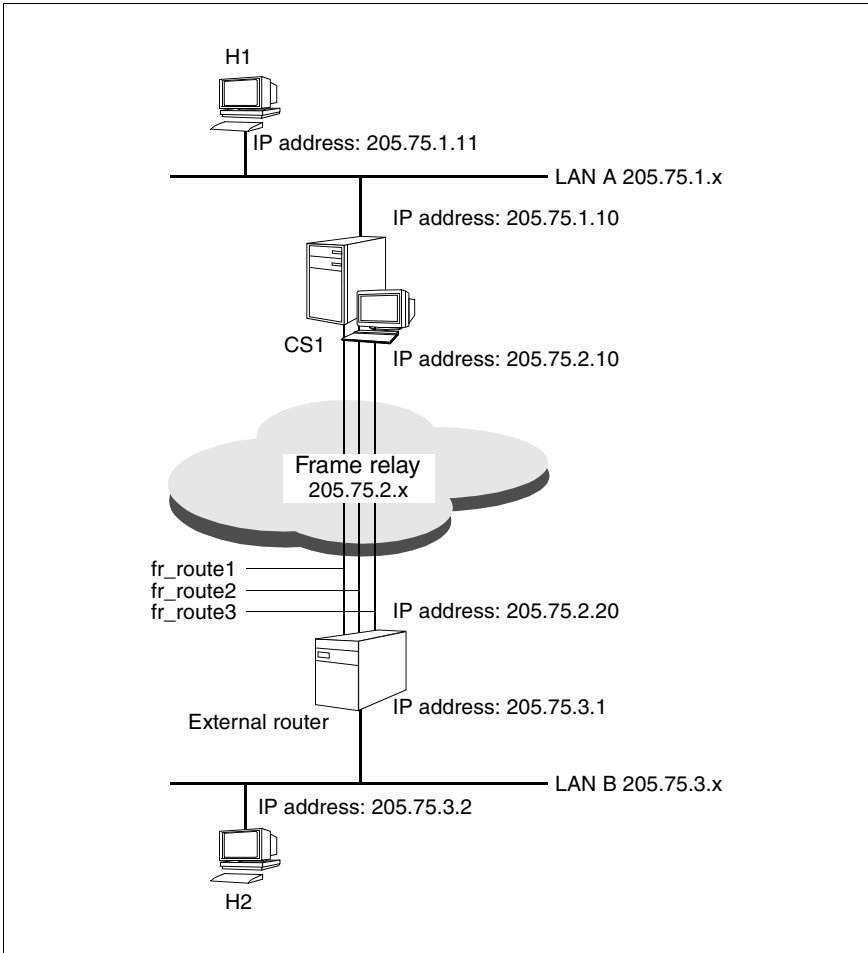


Figure 23: TCP/IP via frame relay (3 PVCs)

H1 and H2 are end systems in TCP/IP LANs. CS1 is a Solaris system with CS-ROUTE. The connection to the external router fr_router is via three frame relay PVCs (fr_route1, fr_route2 and fr_route3).

The IP route entries are the same as in the example “TCP/IP via ISDN” on page 110. You must note the following differences for CS1 when configuring the IP interface, subnetwork interface and FSS.

Configuration data for CS1:

- IP interface (IP address of the WAN connection):

```
csr create if name=clwipl ipaddr=205.75.2.10
          snid-list=FR-1
```

- Subnetwork interface:

```
csr create net type=FR
```

- FSS configuration:

You must define an SNPAROUTES object with the *fr-pvc* and *facil* attributes for each frame relay PVC. The PVC number, which you must specify together with the CC and line numbers for the *fr-pvc* attribute, must be negotiated with the network provider.

The values for the *fr-cbs*, *fr-cir*, *fr-eps* and *fr-max-transit-delay* attributes must also be negotiated with the network provider and those of the *fr-cbs*, *fr-cir*, *fr-eps* and *fr-encapsulation* attributes with the partner system. The attributes of the FACIL objects are described in the manual “CMX/CCP, WAN Communication” [3].

```
fssadm create FACIL name=fr_facil fr-encapsulation=YES \
              fr-prio=1 fr-cbs=64 fr-cir=64 fr-eps=0 \
              fr-max-transit-delay=10
fssadm create SNPAROUTES name=fr_route1 subnet=FR-1 \
              fr_pvc=2/1/41 facil=fr_facil
fssadm create SNPAROUTES name=fr_route2 subnet=FR-1 \
              fr_pvc=2/1/42 facil=fr_facil
fssadm create SNPAROUTES name=fr_route3 subnet=FR-1 \
              fr_pvc=2/1/43 facil=fr_facil
fssadm create NSAP name=fr_router internet-addr=205.75.2.20 \
              snpa-list="fr_route1 + fr_route2 + fr_route3"
```

8.4 TCP/IP via a direct data connection

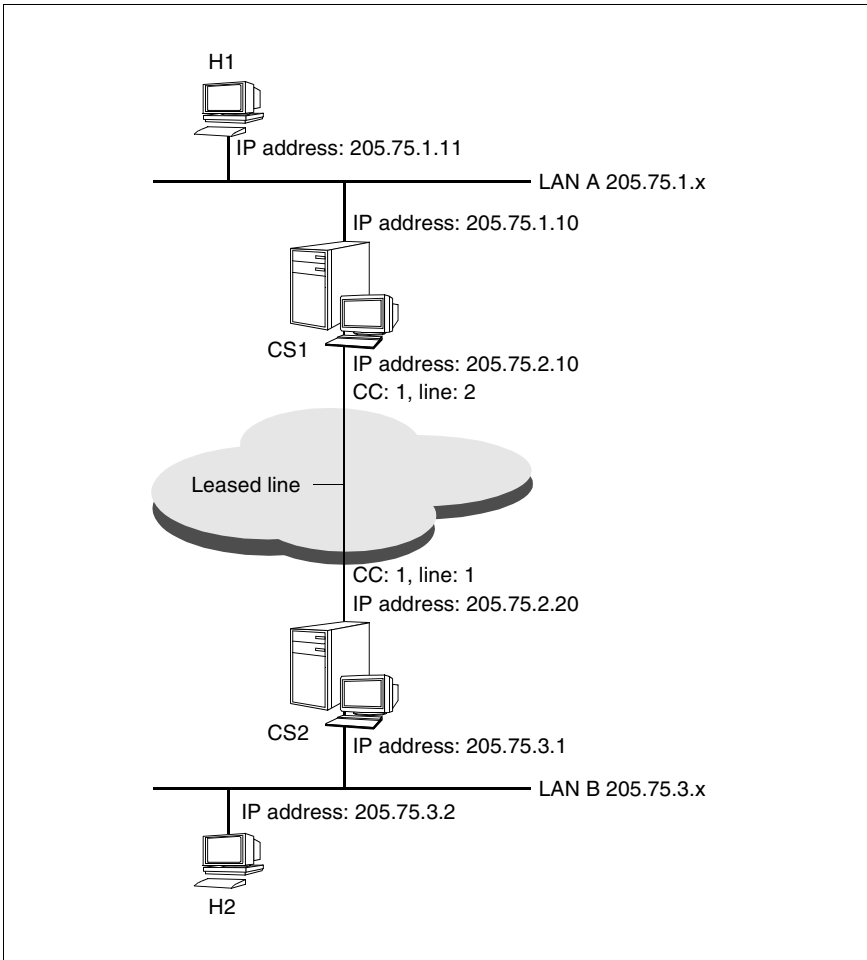


Figure 24: TCP/IP via a direct data connection

H1 and H2 are end systems in TCP/IP LANs. CS1 and CS2 are Solaris/Reliant UNIX systems with CS-ROUTE. The LANs are connected via a direct data connection.

The IP route entries are the same as in the example “TCP/IP via ISDN” on page 110. You must note the following differences for CS1 and CS2 when configuring the IP interface, subnetwork interface and FSS.

In the FACIL object you must use the *npid* attribute to specify the network protocol used, as there is no connection setup phase for a direct data connection.

Note that no profile entry may be defined in the KOGS XLTNG macro for direct data connections.

```
XLTNG  PROFIL  = -.
      ...
```

Configuration data for CS1:

- IP interface (IP address of the WAN connection):

```
csr create if name=clwipl ipaddr=205.75.2.10
          snid-list=PP-1
```

- Subnetwork interface:

```
csr create net type=PP
```

- FSS configuration:

```
fssadm create FACIL name=cs_fac npid=INTERNET
fssadm create SNPAROUTES name=r_cs1_cs2 subnet=PP-1 \
          line-nr=1/2 facil=cs_fac
fssadm create NSAP name=CS2 internet-addr=205.75.2.20 \
          snpa-list=r_cs1_cs2
```

Configuration data for CS2:

- IP interface (IP address of the WAN connection):

```
csr create if name=clwipl ipaddr=205.75.2.20
          snid-list=PP-1
```

- Subnetwork interface:

```
csr create net type=PP
```

- FSS configuration:

```
fssadm create FACIL name=cs_fac npid=INTERNET
fssadm create SNPAROUTES name=r_cs2_cs1 subnet=PP-1 \
          line-nr=1/1 facil=cs_fac
fssadm create NSAP name=CS1 internet-addr=205.75.2.10 \
          snpa-list=r_cs2_cs1
```

8.5 Alternate routing

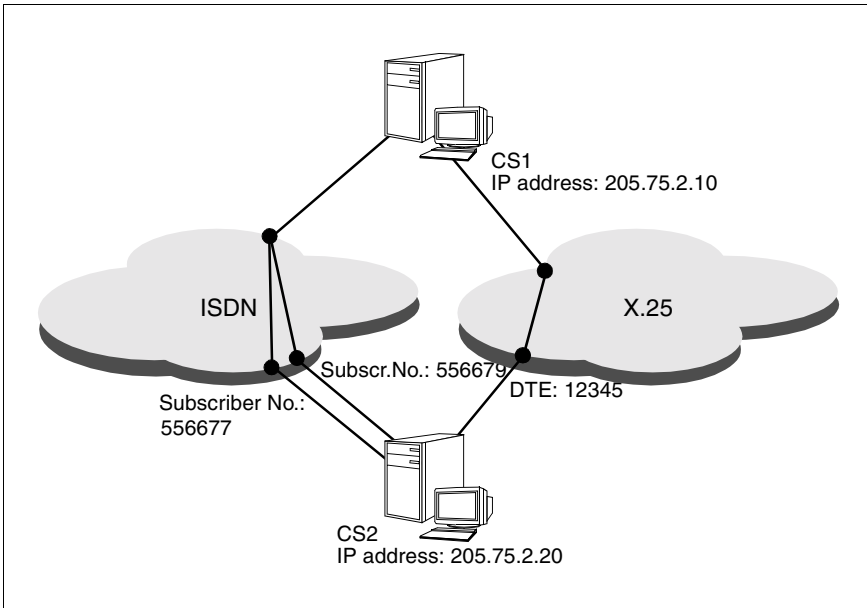


Figure 25: Alternate routing via ISDN and X.25

CS2 can be reached from CS1 via three routes:

- Route 1 via ISDN with the ISDN subscriber number 556677
- Route 2 via ISDN with the ISDN subscriber number 556679
- Route 3 via X.25 with the appropriate DTE address 12345

Route 1 is to have the highest priority (20), route 2 a medium priority (10) and route 3 the lowest priority (1).

The IP route entries are the same as in the example “TCP/IP via ISDN” on page 110. You must note the following differences for CS1 when configuring the IP interface, subnetwork interface and FSS.

Configuration data for CS1:

- IP interface (IP address of the WAN connection):

```
csr create if name=clwipl ipaddr=205.75.2.10
          snid-list="ISDN-1 X25-1"
```

- Subnetwork interface:

```
csr create net type=ISDN
csr create net type=X25
```

- FSS configuration:

You define an **SNPAROUTES** object for each route. In the corresponding **NSAP** object you list all routes via which the partner system is to be reached with the *snpa-list* attribute. Each route is assigned a priority.

```
fssadm create FACIL name=cs_fac ppp_profile=YES
fssadm create SNPAROUTES name=route_1 subnet=ISDN-1 \
             isdn-nr=556677 facil=cs_fac
fssadm create SNPAROUTES name=route_2 subnet=ISDN-1 \
             isdn-nr=556679 facil=cs_fac
fssadm create SNPAROUTES name=route_3 subnet=X25-1 \
             dte-addr=12345
fssadm create NSAP name=CS2 internet-addr=205.75.2.20 \
             snpa-list="route_1/20 + route_2/10 + route_3/1"
```

8.6 X.31 minimum integration

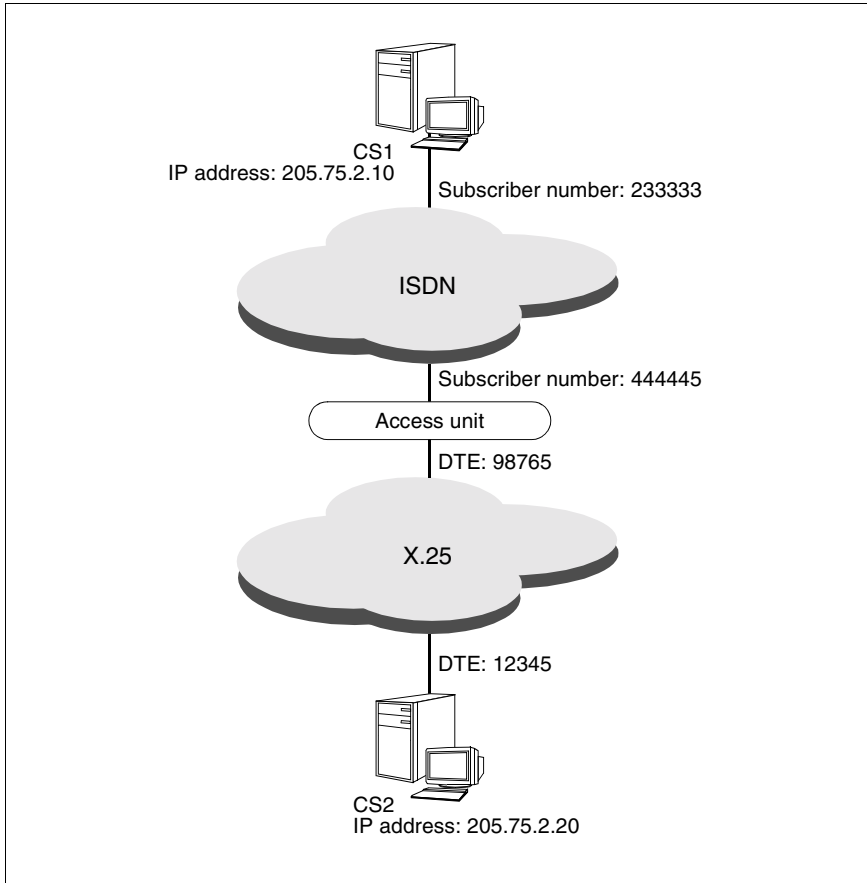


Figure 26: Minimum integration

CS2 can be reached from CS1 using two-step dialing via ISDN and X.25 (minimum integration).

The IP route entries are the same as in the example “TCP/IP via ISDN” on page 110. You must note the following differences for CS1 when configuring the IP interface, subnetwork interface and FSS.

Configuration data for CS1:

The configuration of an ISDN connection for minimum integration is described in detail in the manual “CMX/CCP, ISDN Communication” [4].

The main entries needed to set up your own ISDN connection (CC configuration file) are your own ISDN subscriber number and the description of your own X25 connection.

- IP interface (IP address of the WAN connection):

```
csr create if name=clwip1 ipaddr=205.75.2.10
          snid-list=ISDN-1
```

- Subnetwork interface:

```
csr create net type=ISDN
```

- FSS configuration:

Use the *x25-description* attribute in the FACIL object to specify the predefined description of the X25 connection in the CC configuration file (DTE name).

In the SNPAROUTES object, assign the access unit subscriber number and partner system DTE address to the *x31-msa* attribute.

```
fssadm create FACIL name=min_facil x25-description=x25ACC
fssadm create SNPAROUTES name=min_r subnet=ISDN-1 \
              x31-msa=444445/12345 facil=min_facil
fssadm create NSAP name=CS2 internet-addr=205.75.2.20 \
              snpa-list=min_r
```

8.7 X.31 maximum integration

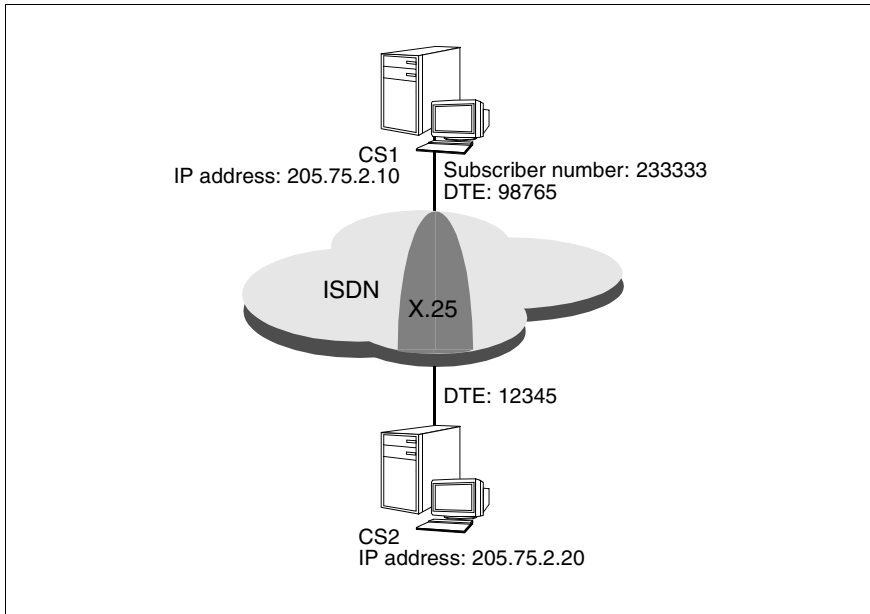


Figure 27: Maximum integration

The ISDN network provides a packet switching service according to X.25. CS2 can be reached from CS1 via maximum integration.

The IP route entries are the same as in the example “TCP/IP via ISDN” on page 110. You must note the following differences for CS1 when configuring the IP interface, subnetwork interface and FSS.

Configuration data for CS1:

The configuration of an ISDN connection for maximum integration is described in detail in the manual “CMX/CCP, ISDN Communication” [4].

The main entries needed to set up your own ISDN connection (CC configuration file) are your own ISDN subscriber number and the description of your own X25 connection.

- IP interface (IP address of the WAN connection):

```
csr create if name=clwip1 ipaddr=205.75.2.10
          snid-list=ISDN-1
```

- Subnetwork interface:

```
csr create net type=ISDN
```

- FSS configuration:

Use the *x25-description* attribute in the FACIL object to specify the predefined description of the X25 connection in the CC configuration file (DTE name).

In the SNPAROUTES object, assign the partner system DTE address at the X.25 SVC to the *x31-dte-addr* attribute.

```
fssadm create FACIL name=max_facil x25-description=x25ACC
fssadm create SNPAROUTES name=max_r subnet=ISDN-1 \
              x31-dte-addr=12345 facil=max_facil
fssadm create NSAP name=CS2 internet-addr=205.75.2.20 \
              snpa-list=max_r
```

8.8 X.32 dialing

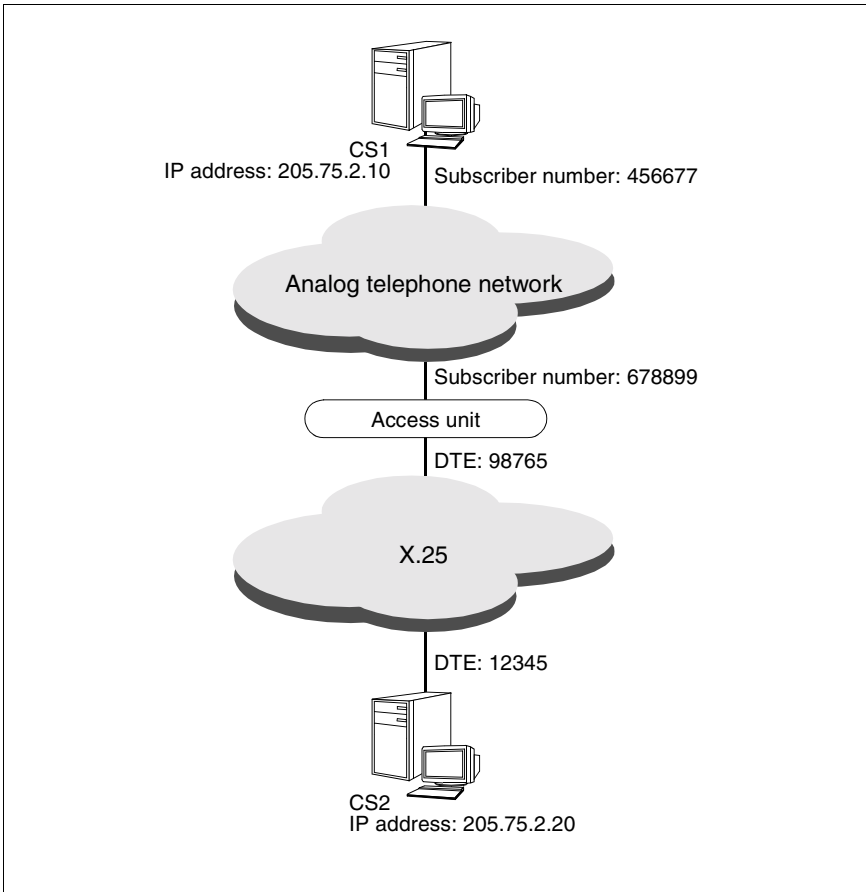


Figure 28: X.32 dialing

CS2 can be reached from CS1 using two-step dialing via the analog telephone network and X.25 (X.32 dialing).

The IP route entries are the same as in the example “TCP/IP via ISDN” on page 110. You must note the following differences for CS1 when configuring the IP interface, subnetwork interface and FSS.

Configuration data for CS1:

The configuration of a telephone connection for X.32 dialing is described in detail in the manual “CMX/CCP, WAN Communication” [3].

– KOGS XLTNG macro:

Your own telephone number is specified as follows:

```
XLTNG RUFNUM = 456677.
    ...
```

– KOGS XZSTW macro:

The telephone number and DTE address of the access unit are specified as follows:

```
XZSTW SKANALN = 1-10,
      NAME     = x25ACC,
      RUFNUM   = 678899
      DTEADR   = 98765,
      NETZTYP  = X25/TYP9
```

– IP interface (IP address of the WAN connection):

```
csr create if name=clwipl ipaddr=205.75.2.10
          snid-list=PT-1
```

– Subnetwork interface:

```
csr create net type=PT
```

– FSS configuration:

Use the *x25-description* attribute in the FACIL object to specify the name of the KOGS XZSTW macro, which describes the X25 connection.

In the SNPAROUTES object, assign the access unit telephone number (X.25 exchange) and the partner system DTE address at the X.25 SVC to the *x32-phone-nr* attribute.

```
fssadm create FACIL name=x32_facil x25-description=x25ACC
fssadm create SNPAROUTES name=x32_r subnet=PT-1 \
          x32-phone-nr=678899/12345 facil=x32_facil
fssadm create NSAP name=CS2 internet-addr=205.75.2.20 \
          snpa-list=x32_r
```

An additional SUBNET object is required if CS1 is to be both an active and passive partner at connection setup.

This SUBNET object must be used to assign the name of the KOGS macro XZSTW, which describes the X.25 connection, to the local telephone number of CS1.

```
fssadm create FACIL name=x32_facil x25-description=x25ACC
fssadm create SNPAROUTES name=x32_r subnet=PT-1 \
    x32-phone-nr=678899/12345 facil=x32_facil
fssadm create SUBNET subnet=PT-1 \
    x25-description=X25ACC
fssadm create NSAP name=CS2 internet-addr=205.75.2.20 \
    snpa-list=x32_r
```


8.9 Setting up message filters

Example 1

```
csr create filter name=filter1 access=permit dir=input
      srcaddr=195.75.18.50 destport=102
```

Effect:

Only incoming TCP/IP messages with the sender IP address 195.75.18.50 and the destination port number 102 are routed onwards. All other incoming messages are rejected.

Outgoing messages are unaffected. The filter is not yet active.

Example 2

```
csr create filter name=filter2 access=permit dir=input
      prot=tcp srcaddr=192.168.10.0 mask=0.0.0.255
      destaddr=192.168.20.10 destport=102 op=ne state=up
```

Effect:

All incoming messages with the sender address 192.168.x.x and the destination address 192.168.20.11 are accepted if they are not intended for the port rfc1006. The filter becomes active immediately.

Example 3

```
cr fi filter3 destport=23
```

Fully expanded, this command is as follows:

```
csr create filter name=filter3 access=deny dir=input
      prot=ip destport=23 op=eq state=down
```

Effect:

All incoming messages addressed to the port *telnet* are to be rejected. The filter is not yet active.

9 What to do if...

This chapter describes the most common problems encountered during configuration and in the state of the local Solaris system. The actions listed under “Solution:” generally eliminate the error and thus enable the configuration to be run successfully.

9.1 CS-ROUTE is not started automatically after a system start

Possible causes:	Solution:
The autostart function is not activated.	<ul style="list-style-type: none"><li data-bbox="468 619 947 671">▶ Check whether the autostart function is activated with the <i>csr</i> command.<li data-bbox="468 692 938 745">▶ Activate the autostart function with the <i>csr autostart</i> command.

9.2 The IP interface cannot be configured/started

Possible causes:	Solution:
<p>No local access with corresponding subnetwork ID.</p>	<ul style="list-style-type: none"> ▶ Display the subnetwork IDs assigned to the local subnetwork accesses with the <i>csr get lsn</i> command. ▶ Deactivate the IP interface with the <i>csr set if name=... state=down</i> command. ▶ If CS-ROUTE runs, customize the subnetwork ID (list) assigned to the IP or OSI-CLNP interface, to the CCP configuration with the <i>csr set if name=... snid-list=... state=up</i> command. <p>If CS-ROUTE is not running, start it with the <i>csr start</i> command.</p>
<p>The subnetwork ID (list) assigned to the IP interface is not unique. (Each subnetwork ID may only be assigned to one IP interface.)</p>	<ul style="list-style-type: none"> ▶ Display the interface configuration data with the <i>csr get if</i> command. ▶ Deactivate the IP interface with the <i>csr set if name=... state=down</i> command. ▶ Correct the configuration data and activate the interface with the <i>csr set if name=... snid-list=... state=up</i> command.
<p>An IP interface with an empty subnetwork ID list already exists. (All known subnetwork IDs are assigned to this interface.)</p>	<ul style="list-style-type: none"> ▶ Deactivate the IP interface with the <i>csr set if name=... state=down</i> command. ▶ Assign a subnetwork ID (list) to this IP interface and activate the interface with the <i>csr set if name=... snid-list=... state=up</i> command. ▶ Create a new IP interface with a different subnetwork ID (list) with the <i>csr create if name=... snid-list=... state=up</i> command.

9.3 The subnetwork interface cannot be activated

Possible causes:	Solution:
The corresponding subnetwork interface is not configured.	<ul style="list-style-type: none"><li data-bbox="468 347 1029 654">▶ Display the status of the subnetwork interface with the <i>csr get net</i> command. Status indication “default“: The subnetwork interface was not explicitly configured. All parameters are set to default. Status indication “down“: The subnetwork interface was configured but not activated.<li data-bbox="468 675 1029 751">▶ Configure the subnetwork interface and activate it with the <i>csr create net type=... state=up</i> command.

9.4 The WAN partner does not respond to the *ping* command

The following refers to communication between a Solaris system with CS-ROUTE and a WAN partner or a system that can be reached via a WAN.

Possible causes:	Solution:
No configured/ activated subnetwork interface.	<ul style="list-style-type: none"><li data-bbox="393 424 948 735">▶ Check the configured subnetwork interfaces with the <i>csr get net</i> command. Status indication “default”: The subnetwork interface was not explicitly configured. All parameters are set to default. Status indication “down”: The subnetwork interface was configured but not activated.<li data-bbox="393 751 948 871">▶ Configure the desired subnetwork interface with the <i>csr create net type=...</i> command or activate the configured subnetwork interface with the <i>csr set net type=... state=up</i> command.
No configured/ activated IP interface .	<ul style="list-style-type: none"><li data-bbox="393 887 948 951">▶ Check the configured interfaces with the <i>csr get if</i> command.<li data-bbox="393 967 948 1090">▶ Configure the desired interface with the <i>csr create if name=...</i> command or activate the configured interface with the <i>csr set if name=... state=up</i> command.

Possible causes:	Solution:
<p>The IP module data output is not via the CS-ROUTE IP interface.</p>	<ul style="list-style-type: none"> ▶ Check the IP routes with the <i>netstat -r[n]</i> command. ▶ If required, configure the necessary interface-specific IP routes in the <i>/opt/lib/clw/clw.routes/clwip<n></i> file. ▶ Deactivate the IP interface with the <i>csr set if name=clwip<n> state=down</i> command. ▶ Activate the IP interface with the <i>csr set if name=clwip<n> state=up</i> command. ▶ Check whether the loopback interface is active with the <i>netstat -i</i> command. ▶ If necessary, activate the loopback interface.
<p>The IP module passes the wrong NSAP address of the “next hop”.</p>	<p>If the WAN partner is on a different IP subnetwork from the IP viewpoint, the value 0 must be specified as metric.</p> <ul style="list-style-type: none"> ▶ Check the IP routes with the <i>netstat -r[n]</i> command. ▶ Enter the correct command for defining an IP route in the <i>/opt/lib/clw/clw.routes.clwip<n></i> file. ▶ Deactivate the IP interface with the <i>csr set if name=clwip<n> state=down</i> command. ▶ Activate the IP interface with the <i>csr set if name=clwip<n> state=up</i> command.
<p>The IP module does not have the information for forwarding the message.</p>	<ul style="list-style-type: none"> ▶ Check the IP routes with the <i>netstat -r[n]</i> command. ▶ Check whether the local interface responds to the <i>ping</i> command by specifying the IP address of the local interface in the <i>ping</i> command. ▶ If the local interface does not respond to the <i>ping</i> command, check whether the loopback interface is active with the <i>netstat -i</i> command.

What to do if...

Possible causes:	Solution:
The NSAP object of the WAN partner is not configured in the FSS.	<ul style="list-style-type: none">▶ Check the configured NSAP object with the <i>fssadm get NSAP</i> command.▶ Configure the missing NSAP object with the <i>fssadm create NSAP name=... snpa-list=</i> command.
No route is configured in the FSS.	<ul style="list-style-type: none">▶ Check the route(s) assigned to the NSAP object with the <i>fssadm get NSAP name=...</i> command.▶ Configure an SNPAROUTES object in the FSS with the <i>fssadm create SNPAROUTES name=...</i> command.▶ Refer to this SNPAROUTES object in the NSAP object with the <i>fssadm set NSAP name=... snpa-list=...</i> command.
The route is configured in the FSS but the corresponding subnetwork interface is not active.	<ul style="list-style-type: none">▶ Check the configured route with the <i>fssadm get NSAP</i> and <i>fssadm get SNPAROUTES</i> commands.▶ Check the subnetwork interfaces with the <i>csr get net</i> command.▶ Configure the desired subnetwork interface and activate it with the <i>csr create net type=... state=up</i> command or activate the configured subnetwork interface with the <i>csr set net type=... state=up</i> command.

Possible causes:	Solution:
<p>Inconsistent subnetwork IDs in the route configuration .</p>	<ul style="list-style-type: none"> ▶ Check the configuration of the SNPAROUTES objects with the <i>fssadm get SNPAROUTES name=...</i> command. ▶ Check the configuration of the local accesses with the <i>csr get lsn</i> command. ▶ Update the FSS configuration of the subnetwork route(s) to the description of the local accesses/the local accesses. ▶ Check the interface configuration with the <i>csr get if</i> command. ▶ Customize the interface configuration and subnetwork routes configuration to each other with the <i>fssadm set SNPAROUTES name=... subnet=...</i> and <i>csr set if name=...</i> commands. ▶ Deactivate the interface with the <i>csr set if name=... state=down</i> command. ▶ Activate the interface with the <i>csr set if name=... state=up</i> command.
<p>Your own connection is not ready.</p>	<ul style="list-style-type: none"> ▶ Check the state. (Is the line connected?). ▶ Activate the line.
<p>The subnetwork connection succeeds but data can only be transferred in one direction.</p>	<ul style="list-style-type: none"> ▶ Check (KOGS and FACIL object in the FSS) whether the subnetwork profile is configured identically on both sides (e.g. with/without T.70-3).
<p>Inconsistent VJHC in the WAN partner configuration (without PPP).</p>	<ul style="list-style-type: none"> ▶ Check the slot array size with the <i>csr get clw</i> command. ▶ Update the slot array size with the <i>csr set clw maxslot=...</i> command.
<p>Message filter prevents data transfer</p>	<ul style="list-style-type: none"> ▶ Check the configured, active message filters.

9.5 The WAN partner can be reached actively but no passive connection can be made

Possible causes:	Solution:
The local system is connected via a PABX. Different subscriber numbers must be considered depending on the connection setup direction (with and without leading zero).	<ul style="list-style-type: none"><li data-bbox="393 347 956 411">▶ Configure a second subnetwork route with the <code>fssadm create SNPAROUTES name=...</code> command.<li data-bbox="393 427 956 515">▶ Extend the list of assigned subnetwork routes in the NSAP object with the <code>fssadm set NSAP name=... snpa-list=...</code> command.

9.6 The local system does not respond to the *ping* command from the WAN partner

Possible causes:	Solution:
<p>The local system rejects the subnetwork connection.</p>	<ul style="list-style-type: none"> ▶ Check whether the WAN partner sender subscriber number is configured with the <i>fssadm get SNPAROUTES</i> command. ▶ If required, configure the necessary subnetwork route and assign it to the NSAP object with the <i>fssadm create SNPAROUTES name=...</i> and <i>fssadm set NSAP name=... snpa-list=...</i> commands. ▶ Check whether access protection was configured for the route with the <i>fssadm get FACIL</i> command. <p>Note that two subnetwork routes to a partner system may be required in some cases for access via a PABX (see the section “Configuration for PBXs” on page 117).</p>
<p>The connection is accepted but no link is made to the NSAP object.</p>	<ul style="list-style-type: none"> ▶ Check the NSAP object with the <i>fssadm get NSAP name=...</i> command ▶ Make the link to the subnetwork route with the <i>fssadm set NSAP name=... snpa-list=...</i> command.
<p>Message filter prevents data transfer</p>	<ul style="list-style-type: none"> ▶ Check the configured, active message filters.

Glossary

active partner

Communication partner that initiates connection setup (outgoing) in the case of connection-oriented protocols

alternate routing

If a line is occupied or fails, an attempt is made to set the connection up via an alternative route in the same or a different subnetwork.

American Standard Code for Information Interchange (ASCII)

International code for DP systems.

area (OSPF)

With OSPF, routers can be divided into logical groups known as areas. The structure of an area is hidden from the rest of the autonomous system and leads to a considerable reduction of routing traffic in the network. The routing within an area is determined by its topology, and is protected against incorrect routing data.

area border router (OSPF)

Area Border Routers are routers which are connected to a number of areas.

AS boundary router (OSPF)

Autonomous System Boundary Routers are routers which exchange information with routers in another AS. The AS Boundary Routers distribute the information throughout the entire AS via external routes.

ASCII

See *American Standard Code for Information Interchange*.

Autonomous System (AS) (OSPF)

Group of routers using a uniform routing protocol.

broadcast networks

In a broadcast network, all stations are connected to one transmission medium.

CC

See *Communication Controller*.

CCP

See *Communication Control Program*.

CCP profile

A CCP profile defines a specific protocol for each of the four lower layers (*transport system*) in the *OSI Reference Model*, thus determining specific network characteristics. A CCP profile comprises a *Transport Service Provider* and a *subnetwork profile*.

CHAP

Challenge-Handshake Authentication Protocol: procedure for access protection at user level which uses encrypted passwords.

CMX

Communications Manager in Solaris.

communication controller (CC)

Communication hardware in Solaris for connecting a Solaris system to a network. A CC is controlled by a *subnetwork profile*.

communication control program (CCP)

Software product which implements the lower four layers (*transport system*) of the *OSI Reference Model*. Access to the services of a CCP is provided via *CMX* with the *ICMX*, *XTI* or *TLI* program interfaces.

configuration-oriented generator language (KOGS)

Special language for configuring the operating parameters of a *subnetwork profile*.

designated router (OSPF)

Each multi-access network with at least two routers has a designated router. The designated router maintains the "original" topology database of the network. It plays an important role in the exchange of network status information, as all routers in the network coordinate their topology database through the sending and receiving of advertisements to the designated router.

The designated router concept allows a reduction in the number of partner relationships in a multi-access network. This reduces the routing traffic load in the network and the size of the topology database.

direct data connection

Leased line with one end equipment at each end (point-to-point connection).

end system

Host on which *TS applications* run.

Forwarding Support Information Base (FSB)

Database of the *Forwarding Support Service*.

Forwarding Support Service (FSS)

Service in *CMX* which provides the communication components in the Solaris kernel with information required in the layers of the *transport system* for establishing connections and forwarding data.

FSB

See *Forwarding Support Information Base*.

FSS

See *Forwarding Support Service*.

host

Computer on which *TS applications* run.

interior gateway protocol

Protocol for the distribution of routing information in an autonomous system.

Intermediate System (IS)

System which provides a relay function between *subnetworks* in layer 3 of the *OSI Reference Model*. Corresponds to a router.

internal router

Routers which are only connected to networks within one area are called internal routers.

Internet address

The Internet address is a four-byte number. It is represented by four fields which are separated by periods, e.g. 129.144.50.56. Each field represents one byte in decimal notation.

The Internet address is split into the following fields: one network number, one host number and one subnetwork number (optional).

The network number is the first part of the Internet address. It may be 1, 2 or 3 bytes long. The remaining bits are split into the subnetwork number (optional) and host number.

The host number uniquely identifies a specific host in a network with a common network and subnetwork number.

Internet protocol (IP)

IP is a protocol which is responsible for routing in a computer network. The protocol uses a four-byte Internet address for the destination and source specification. This address indicates the network and the system within the network which are to be addressed.

IP address

See *Internet address*.

KOGS

See *configuration-oriented generator language*.

maximum integration

Transmission of X.25 data via ISDN connections.

minimum integration

Two-step dialing to X.25 networks via ISDN according to X.31 case A.

multi-access networks (OSPF)

Networks which support more than two routers are called multi-access networks. Each pair of routers in this network can communicate directly with one another.

Network Service Access Point (NSAP)

Access point in a *end system*, via which the transport layer can enlist the services of the network layer. The *NSAP address* uniquely identifies a *end system* in the *network*.

network

Combination of systems that interact using a uniform protocol. A network protocol is a protocol between layer 3 entities in the context of the OSI Reference Model, e.g. Internet protocol, OSI-CLNP.

network address

Unique identification of an *NSAP* (NSAP address) or a layer 3 entity (network entity title) in a *network*.

NSAP

See *Network Service Access Point*.

NSAP address

Unique identification of an *NSAP* throughout the network, which allows unique addressing of a *end system* in the *network*.

OSI Reference Model

The Open Systems Interconnection Reference Model for communication between "open systems" is described in the IS 7498 standard.

PAP

Password Authentication Protocol: procedure for access protection at user level.

partner router

Special relationship configured between adjacent routers. Partner routers are routers which exchange their routing information.

passive partner

Communication partner that receives a connection setup request (incoming) passively in the case of connection-oriented protocols.

point-to-point protocol

Standardized method with which datagrams are routed via TCP/IP (RFCs 1171, 1172 and 1661).

protocol entity

Entity which implements a protocol belonging to a specific layer of the *transport system*.

router

System which provides a relay function between *subnetworks* in layer 3 of the *OSI Reference Model*.

routing

Routing is the forwarding of packets in a network without protocol conversion.

routing protocol

Protocol used by routers to exchange information concerning topologies, modifications and route costs with one another or with the connected end systems.

routing table

Each router has access to a routing table which contains the information required for routing messages. On the basis of the routing table, the router decides via which route, i.e. via which partners in the network, messages are forwarded.

SNID

See *subnetwork ID*.

SNPA

See *Subnetwork Point of Attachment*.

SNPA address

Unique identification of an *SNPA* within a *subnetwork*, e.g. subscriber number in an X.25 network.

stub area (OSPF)

OSPF allows specific areas to be configured as "dead ends", or what are known as "stub areas". AS external advertisements are not routed into or through stub areas. Routing to AS external destinations is based on a default mechanism in the stub areas. Configuration of the stub areas reduces the size of the topology database and memory requirements for an internal router in a stub area.

Subnetwork Point of Attachment (SNPA)

Access point of a system to a *subnetwork*. The SNPA is uniquely identified in the *subnetwork* by its *SNPA address*.

subnetwork

The sum of all transmission configurations and physical media which, as an independent unit, allow the interconnection of systems for the purpose of communication. Examples of subnetworks are local area networks (LAN) or public packet switching networks.

subnetwork access

A *subnetwork* access comprises all the attributes of a *subnetwork* connection. The definition of the term subnetwork access used here is general enough to describe both switched connections and leased lines. A fixed generated B channel of an ISDN access can also be considered as a subnetwork access under this definition. A subnetwork access can be managed locally.

subnetwork ID (SNID)

A subnetwork ID identifies a *subnetwork access* or a group of similar *subnetwork accesses* in the same subnetwork, which can address the *transport system* under this identifier.

subnetwork profile

The subnetwork profile contains all the components (*protocol entities*) of a *CCP profile* which control the *communication controller*.

TCP

Transmission Control Protocol RFC 793.

Type of Service (TOS)

Under TOS, the types of service available for forwarding data packets include connection with less delay, high throughput, high reliability or low monetary cost (RFC 1349).

Van-Jacobson header compression

The headers of TCP/IP data packets are compressed using Van-Jacobson header compression.

X.32 dialing

Two-step dialing to X.25 partners via the analog telephone network according to X.32.

Abbreviations

ABR

Area Border Router

AS

Autonomous System

ASBR

AS Boundary Router

CC

Communication Controller

CCP

Communication Control Program

CHAP

Challenge-Handshake Authentication Protocol

CLNP

Connectionless Network Protocol (IS 8473)

CLW

Connectionless WAN Access

CMX

Communications Manager in UNIX

CS

Communication Services

DR

Designated Router

DTE

Data Terminal Equipment

DT PDU

Data Protocol Data Unit

Abbreviations

EGP

Exterior Gateway Protocol

ER PDU

Error Report Protocol Data Unit

FSB

Forwarding Support Information Base

FSS

Forwarding Support Service

FTP

File Transfer Protocol

ICMP

Internet Control Message Protocol

IGP

Interior Gateway Protocol

IP

Internet Protocol

ISDN

Integrated Services Digital Network

IR

Interner Router

LAN

Local Area Network

LS

Link State

LSA

Link State Advertisement

NSAP

Network Service Access Point

NFS

Network File System

OSI

Open Systems Interconnection

OSPF

Open Shortest Path First

OSPF

Open Shortest Path First

PAP

Password Authentication Protocol

PPP

Point-to-Point-Protokoll

PVC

Permanent Virtual Circuit

RFC

Request For Comments

RIP

Routing Information Protocol

SNMP

Simple Network Management Protocol

SNPA

Subnet Point of Access

SPF

Shortest Path First

SVC

Switched Virtual Call

TCP

Transmission Control Protocol

Abbreviations

TIDU

Transport Interface Data Unit

TOS

Type Of Service

TP4

OSI Transport Protocol Class 4 (IS 8073)

TS

Transportsystem

VJHC

Van-Jacobson-Header-Compression

WAN

Wide Area Network

Related publications

- [1] **CMX V5.1** (Solaris)
Communications Manager UNIX
Operation and Administration
User Guide

Target group

System administrators

Contents

The manual describes the function of CMX as mediator between applications and the transport system. It contains basic information on configuration and administration of systems in network environments.

- [2] **CMX V5.1**
Communications Manager UNIX
Programming Applications
Programmers Reference Guide

Target group

Programmers

Contents

The manual describes the program interface of CMX, i.e. all tools that you can use for developing TS applications.

- [3] **CMX/CCP V5.1** (Solaris)
WAN Communication
User Guide

Target group

Network and system administrators

Contents

The manual describes the computer-to-computer connection via WAN (Wide Area Network) allowing communication in the remote area (Wide Area Network, WAN).

Related publications

- [4] **CMX/CCP V5.1 (Solaris)**
ISDN Communication
User Guide

Target group

Network administrators

Contents

The manual describes computer-to-computer connection via ISDN (Integrated Services Digital Network).

- [5] **openFT (UNIX)**
Enterprise File Transfer in the open world
User Guide

Target group

The manual is aimed at users and programmers who wish to transfer or manage files using *openFT* (UNIX).

Contents

This manual describes the facilities of *openFT* (UNIX) including the use of the FTAC functions and the support of FTAM. It contains the command interfaces, the program interface, and the messages.

- [6] **Advanced Server UNIX**
Overview and Installation
Manual for system and network administrators

Ordering manuals

If you want to order manuals, please apply to your local office.

Index

? (help function) 52

A

access

to the subnetwork 12

access (fssadm attribute) 95

access protection

at user level 22

configuring 37

access unit 14

action

(csr command) 52

(fssadm command) 83

activate

IP interface 59

OSI-CLNP interface 59

subnetwork interface 64

active IP interface 55

active OSI-CLNP interface 55

active subnetwork interface 55, 66

address (fssadm attribute) 99

admit (fssadm attribute) 89

ADS trace 78

alternate routing 19

configuring 38, 128

route selection 39

via ISDN and X.25 128

alternative route 19

analog telephone network 1, 7, 12

attribute (csr command) 52

attribute (fssadm command) 87

autostart (csr action) 51

autostart function 53

deactivate 53

autostop (csr action) 51

B

board number 73, 76

C

calls

permit 36

CCP 25

CHAP 22, 37

chap-loc-secret (fssadm attribute)

108

chap-peer-secret (fssadm attribute)

108

client

access options to WAN 9

CLW (Connectionless WAN Access)

54

clw (object class) 52, 54

CLW trace 79

commands for CS-ROUTE 51

component CLW 54

compress (fssadm attribute) 89

compression

Van-Jacobson header 22, 54

configure

IP interface 56

OSI-CLNP interface 56

subnetwork interface 62

configured IP interface 55

configured OSI-CLNP interface 55

configured subnetwork interface 55

configuring

access protection 37

alternate routing 38, 128

facilities 34

IP system 33

local system 27

maximum integration 40, 132

minimum integration 39, 130

partner system 42

point-to-point protocol 35

subnetwork route 38

Van-Jacobson header

compression 37, 113

X.32 dialing 41, 134

Index

- conn (object class) 52, 74
 - connection
 - dial-up (ISDN) 14
 - direct data 12
 - permanent (ISDN) 14
 - point-to-point 12
 - shut down 74
 - with failure 19
 - connection failure 19
 - connection reference 74
 - connection setup
 - active 75
 - passive 75
 - with maximum integration 16
 - with minimum integration 14
 - with X.32 dialing 12
 - create
 - csr object 51
 - FSS object 83
 - create (csr action) 51
 - create (FSS action) 83
 - csr fi 67
 - csr action
 - autostart 51
 - autostop 53
 - create 51
 - delete 51
 - get 51
 - set 51
 - start 51
 - stop 51
 - csr autostart 53
 - csr autostop 53
 - csr command
 - action 52
 - object class 52
 - csr create if 56
 - csr create net 62
 - csr delete if 56
 - csr delete net 62
 - csr get clw 54
 - csr get conn 74
 - csr get if 56
 - csr get lsn 73
 - csr get net 62
 - csr set clw 54
 - csr set conn 74
 - csr set if 56
 - csr set net 62
 - csr start 53
 - csr stop 53
 - CS-ROUTE
 - automatically start 53
 - start 53
 - starting 47
 - stop 53
 - stopping 47
 - CS-ROUTE commands 51
 - CS-ROUTE traces 79
 - csrtroff 81
 - csrtron 80
- ## D
- daemon
 - routing 10
 - data terminal equipment, see DTE
 - deactivate
 - IP interface 59
 - OSI-CLNP interface 59
 - subnetwork interface 64
 - default values
 - point-to-point protocol 35
 - deinstalling 25
 - delete
 - configuration data of a subnetwork interface 62
 - configuration data of an IP interface 56
 - configuration data of an OSI-CLNP interface 56
 - connection-specific statistic data 74
 - csr object 51
 - FSS object 83
 - interface-specific statistic data 59
 - subnetwork-specific statistic data 64
 - delete (csr action) 51

delete (FSS action) 83
diagnostic command 78
dial-up connection 14
direct data connection 1, 7, 12, 126
DTE 12, 14
dte-addr (fssadm attribute) 100

F

facil (fssadm attribute) 89, 103
FACIL (object class) 88
FACIL object 35
facilities
 configuring 34
fi (object class) 52, 67
forwarding support information base
 83
forwarding support service 33
frame relay
 PVC 124
frame relay network 1, 7, 17
fr-pvc (fssadm attribute) 100
FSB, see forwarding support information base
FSS action
 create 83
 delete 83
 for CS-ROUTE 87
 get 84
 set 83
FSS attribute
 access 95
 address 99
 admit 89
 chap-loc-secret 108
 chap-peer-secret 108
 compress 89
 dte-addr 100
 facil 89, 103
 for CS-ROUTE 87
 fr-pvc 100
 internet-addr 93
 isdn-nr 102
 line-nr 100
 loc-id 107

FSS attribute (cont.)
 nailed-up-isdn 102
 name 88, 93, 97, 107
 net 95
 npid 89
 pap-loc-pwd 107
 pap-peer-pwd 108
 peer-id 107
 phone-nr 99
 ppp-accm 90
 ppp-auth-params 92
 ppp-auth-protocol 92
 ppp-profile 90
 pvc-nr 100
 short-id 92, 103
 snpa-list 96
 subnet 94, 97, 98
 type 94
 x31-dte-addr 101
 x31-msa 101
 x31-pvc-nr 101
 x32-phone-nr 102
FSS configuration 83
FSS object class
 for CS-ROUTE 87
fssadm 83
functions 9

G

get (csr action) 51
get (FSS action) 84
GSM 16, 119

H

header compression, see Van-
 Jacobson header compression
help function (csr command) 52

I

identification
 protocol 76
 routes 97
 subnetwork, see subnetwork ID
Idle Time 88

Index

- idle time 65, 76
- if (object class) 52, 56
- information
 - CS-ROUTE status 53
 - retrieve 78
- installation
 - requirements 25
- installing
 - CS-ROUTE 25
- interface
 - IP 56
 - OSI-CLNP 56
 - subnetwork 62
- interface name 57, 60
- internet-addr (fssadm attribute) 93
- IP address
 - of the interface 57, 60
- IP interface 28, 56
 - activate 59
 - active 55
 - address 57, 60
 - configure 56
 - configured 55
 - current state 73
 - current status 61
 - deactivate 59
 - delete configuration data 56
 - modify configuration data 56
 - network mask 57, 60
 - retrieve configuration data 56
- IP system
 - configuring 33
- ISDN
 - dial-up connection 110
 - permanent connection 113
- ISDN network 1, 7, 14
- isdn-nr (fssadm attribute) 102
- L**
- LAN-WAN routing 7, 9
- leased line 126
- line number 73, 76
- line-nr (fssadm attribute) 100
- linking
 - LANs via multiple WANs 11
 - LANs via one WAN 10
- linking LAN islands 9
- list
 - subnetwork ID 58, 60, 66
- local subnetwork access 73
- loc-id (fssadm attribute) 107
- lsn (object class) 52, 73
- M**
- maximum integration 14, 15
 - configuring 40, 132
- maximum transfer unit, see mtu
- message filter
 - setting up 137
- message filters 23, 32
- minimum integration 14
 - configuring 39, 130
- mobile telephone 16
- modify
 - configuration data of a subnetwork interface 62
 - configuration data of an IP interface 56
 - configuration data of an OSI-CLNP interface 56
- mtu 49, 54, 55, 58, 60
- N**
- nailed-up-isdn (fssadm attribute) 102
- name
 - IP interface 57, 60
 - OSI-CLNP interface 57, 60
- name (fssadm attribute) 88, 93, 97, 107
- net (fssadm attribute) 95
- net (object class) 52, 62
- network access 7, 9
- network address
 - of own system 76
 - of remote system 76
- network layer protocol identification 76

network mask
 of the IP interface 57, 60
notational conventions 3
npid (fssadm attribute) 89
NSAP (object class) 93
NSAP object 42

O

object
 create (csr) 51
 create (FSS) 83
 delete (csr) 51
 delete (FSS) 83
 retrieve (csr) 51
 retrieve (FSS) 84
 set (csr) 51
 set (FSS) 83
object class
 clw 52
 conn 52
 FACIL 88
 fi 52
 if 52
 lsn 52
 net 52
 NSAP 93
 PPPAUTH 106
 SNPAROUTES 97
object class (csr command) 52
object class (FSS command) 87
operating
 requirements 25
OSI TP4/CLNP 1
OSI-CLNP interface 56
 activate 59
 active 55
 configure 56
 configured 55
 current state 73
 deactivate 59
 delete configuration data 56
 modify configuration data 56
 retrieve configuration data 56
own subnetwork address 77

P

PAP 22, 37
pap-loc-pwd (fssadm attribute) 107
pap-peer-pwd (fssadm attribute) 108
partner system
 configuring 42
 reachability 33
peer-id (fssadm attribute) 107
permanent connection 14
permanent line 12
permanent virtual call, see PVC
phone-nr (fssadm attribute) 99
point-to-point connection 12
point-to-point protocol 20, 77
 configuring 35
 default values 35
PPP daemon
 start 53
 stop 53
PPP driver 8
PPP trace 79
PPP, see point-to-point protocol
PPP, synchronous 16
ppp-accm (fssadm attribute) 90
PPPAUTH (object class) 106
ppp-auth-params (fssadm attribute)
 92
ppp-auth-protocol (fssadm attribute)
 92
ppp-profile (fssadm attribute) 90
preparation for use 25
preparing
 for operation 25
protocol
 point-to-point- 20
protocol identification 76
PVC 14, 17
pvc-nr (fssadm attribute) 100

R

reference of the connection 74, 77
remote subnetwork address 77
request for comment, see RFC

- retrieve
 - active subnetwork connections 74
 - configuration data of a subnetwork interface 62
 - configuration data of an IP interface 56
 - configuration data of an OSI-CLNP interface 56
 - csr object 51
 - FSS object 84
 - information 78
 - local subnetwork accesses 73
 - RFC 23
 - route
 - configuring 97
 - route selection
 - with alternate routing 39
 - router network 9
 - routing 9
 - routing daemon 10
 - routing service 7
- S**
- set
 - csr object 51
 - FSS object 83
 - set (csr action) 51
 - set (FSS action) 83
 - setup
 - message filter 137
 - short-id (fssadm attribute) 92, 103
 - shut down
 - subnetwork connection 74
 - slot array 38, 54, 55, 76
 - with point-to-point protocol 54
 - SNPA address type 94, 98
 - snpa-list 96
 - snpa-list (fssadm attribute) 96
 - SNPAROUTES (object class) 97
 - start
 - CS-ROUTE 53
 - CS-ROUTE automatically 53
 - PPP daemon 53
 - start (csr action) 51
 - state
 - of the IP interface 73
 - of the OSI-CLNP interface 73
 - of the subnetwork connection 77
 - statistic
 - delete interface-specific 59
 - delete subnetwork-specific 64
 - delete, connection-specific 74
 - status
 - subnetwork interface 66
 - status information 53
 - status of the IP interface 61
 - stop
 - CS-ROUTE 53
 - PPP daemon 53
 - stop (csr action) 51
 - subnet (fssadm attribute) 94, 97, 98
 - subnetwork access
 - address 99
 - local 73
 - subnetwork address
 - own 73, 77
 - remote 77
 - subnetwork connection
 - active 65
 - current state 77
 - shut down 74
 - subnetwork connections
 - retrieve 74
 - subnetwork ID 58, 60, 66, 97
 - subnetwork interface 32, 62
 - activate 64
 - active 55, 66
 - configure 62
 - configured 55
 - current status 66
 - deactivate 64
 - delete configuration data 62
 - modify configuration data 62
 - retrieve configuration data 62
 - subnetwork route
 - configuring 38
 - subnetwork type 62, 66, 77
 - SVC 14

switched virtual call, see SVC

T

TCP/IP 1, 7, 25

TCP/IP via direct data connection
126

TCP/IP via frame relay
PVC 124

TCP/IP via ISDN
dial-up connection 110
permanent connection 113

TCP/IP via X.25
PVC 123
SVC 121

telephone network 12

trace

point-to-point protocol 79
type (fssadm attribute) 94

U

user level

access protection 22

V

Van-Jacobson header compression
22, 54, 77, 89, 113
configuring 37

VJHC, see Van-Jacobson header
compression

W

weight (fssadm attribute value),
priority of routes 96

X

X.25 communication
via ISDN 14, 15
via the analog telephone network
12

X.25 network 1, 7, 14

X.25 PVC 123

X.25 SVC 121

X.31 case A 14

X.31 case B 14

X.32 dialing 12

configuring 41, 134

x31-dte-addr (fssadm attribute) 101

x31-msa (fssadm attribute) 101

x31-pvc-nr (fssadm attribute) 101

x32-phone-nr (fssadm attribute) 102

Fujitsu Siemens Computers GmbH
User Documentation
81730 Munich
Germany

Comments
Suggestions
Corrections

Fax: 0 700 / 372 00000

e-mail: manuals@fujitsu-siemens.com
<http://manuals.fujitsu-siemens.com>

Submitted by

Comments on CMX V5.1

Communications Manager UNIX TCP/IP via WAN/ISDN



Fujitsu Siemens Computers GmbH
User Documentation
81730 Munich
Germany

Comments
Suggestions
Corrections

Fax: 0 700 / 372 00000

e-mail: manuals@fujitsu-siemens.com
<http://manuals.fujitsu-siemens.com>

Submitted by

Comments on CMX V5.1

Communications Manager UNIX TCP/IP via WAN/ISDN





Information on this document

On April 1, 2009, Fujitsu became the sole owner of Fujitsu Siemens Computers. This new subsidiary of Fujitsu has been renamed Fujitsu Technology Solutions.

This document from the document archive refers to a product version which was released a considerable time ago or which is no longer marketed.

Please note that all company references and copyrights in this document have been legally transferred to Fujitsu Technology Solutions.

Contact and support addresses will now be offered by Fujitsu Technology Solutions and have the format ...@ts.fujitsu.com.

The Internet pages of Fujitsu Technology Solutions are available at [http://ts.fujitsu.com/...](http://ts.fujitsu.com/) and the user documentation at <http://manuals.ts.fujitsu.com>.

Copyright Fujitsu Technology Solutions, 2009

Hinweise zum vorliegenden Dokument

Zum 1. April 2009 ist Fujitsu Siemens Computers in den alleinigen Besitz von Fujitsu übergegangen. Diese neue Tochtergesellschaft von Fujitsu trägt seitdem den Namen Fujitsu Technology Solutions.

Das vorliegende Dokument aus dem Dokumentenarchiv bezieht sich auf eine bereits vor längerer Zeit freigegebene oder nicht mehr im Vertrieb befindliche Produktversion.

Bitte beachten Sie, dass alle Firmenbezüge und Copyrights im vorliegenden Dokument rechtlich auf Fujitsu Technology Solutions übergegangen sind.

Kontakt- und Supportadressen werden nun von Fujitsu Technology Solutions angeboten und haben die Form ...@ts.fujitsu.com.

Die Internetseiten von Fujitsu Technology Solutions finden Sie unter [http://de.ts.fujitsu.com/...](http://de.ts.fujitsu.com/), und unter <http://manuals.ts.fujitsu.com> finden Sie die Benutzerdokumentation.

Copyright Fujitsu Technology Solutions, 2009