There are several communities using OSI at present, the short list is:

Mneumoic	Community	TS-stack
`int-x25`	the International X.25	`x25`
`janet`	the JANET in the UK	`x25`
`internet`	the capital-I Internet	`tcp`
`localTCP`	the TCP loopback address	`tcp`
`IXI`	International X.25 Interconnect	`x25`

These are all termed *interim* communities as they do not use the real OSI network service. In order to facility communications between these communities (and others, e.g., private LANs), an Interim addressing scheme has been defined [Kil89b] (which is included in the ISODE documentation set, look in the directory *doc/interim/* in the source tree).

Thus, the first task when running the ISODE is to determine which communities your site belongs to. The `ts_communities` run-time tailoring variable keeps track of the communities which your host can (in)directly reach. This defaults to the four communities above, plus the `realNS` community. Whilst this is the most sensible default, this choice is probably wrong for most sites. So, if your host is connected to the International X.25 but not the JANET, you add the line

```
ts_communities: int-x25
```

to the *isotailor* file on that host.

Of course, if the host in question is connected not only to the International X.25 but also belongs to the capital-I Internet, then the line might read

```
ts_communities: int-x25 internet
```

instead. This raises an important question: if a host wishes to contact another host, and the two have multiple communities in common, what preference is given when connections are attempted? The answer is that the ordering of the communities in the `ts_communities` run-time tailoring variable tells the ISODE what the preference is. Thus, in the example above, if there were two hosts connected to both the International X.25 and the capital-I Internet, and one of the hosts wished to talk to the other, the International X.25 community would be tried first. If a connection could not be established, then the capital-I Internet community would be tried.

There are a few common site configurations. If the site has access to the capital-I Internet, then usually all hosts belong to the capital-I Internet community, e.g.,

```
ts_communities: internet
```

It is possible that one or more hosts at this site may have access to the International X.25, and exactly these hosts would have connectivity to this second community:

```
ts_communities: internet int-x25
```

8.2.1 Defining a new OSI community

Another common configuration is that a site has a single host that is connected to the International X.25. In addition, there is a private LAN, running the TCP/IP protocols, that all hosts at the site are connected to. In this case, you need to define a community name for your site. This is done by describing your community name as a macro in the *isomacros*(5) file and then enabling use of this community in the *isotailor*(5) file.

To describe your community, you create a macro of the form:

```
name    TELEX+value+stack+number+
```

where:

`name:` is the name of your community;

`value:` corresponds to the TELEX number at your site. (This is traditionally a three-digit international code followed by a national TELEX number.) The combination of `TELEX+value` defines an OSI address prefix which your site uniquely administers;

`stack:` is either `RFC-1006` or `X.25(80)` indicating the TS-stack used for the addresses; and,

`number:` by convention is a two-digit decimal number from 01 to 99 (this allows you to define up to 99 communities at your site).

Of course, there are other methods for generating unique OSI addresses, but this is the only method supported in the current implementation of the ISODE.

Defining a new TCP-based community

Consider the the case of an isolated LAN running the TCP/IP protocols.

First, edit the file *support/macros.local* in the source area and add a line line like this:

```
nott-ether TELEX+00738700+RFC-1006+01+
```

where `00738700` corresponds to your TELEX number. Next, type

```
# ./make macros
```

as the super-user. This will install a new version of the *isomacros*(5) file.

Second, in the *isotailor*(5) file, you would add this line:

```
ts_interim: nott-ether
```

which tells the ISODE about your new community. You would then add `nott-ether` to the beginning of the value for the run-time tailoring variable `ts_communities`. So, on the hosts with only TCP, you would have:

```
ts_interim:     nott-ether
ts_communities: nott-ether
```

and on the host which also had connectivity to the International X.25, you would have:

```
ts_interim:     nott-ether
ts_communities: nott-ether int-x25
```

Note that it is **critical** that the definition of `ts_interim` occur **before** the definition of `ts_communities`.

Defining a new X.25-based community

Consider the case of a site which has access to the International X.25 and belongs to a private X.25 network.

In this case, the macro definition might be:

```
psi-wan     TELEX+00738700+X.25(80)+02+
```

So, the tailoring variables would be

```
ts_interim: psi-wan
ts_communities: psi-wan
```

for those hosts which are only on the private X.25 network, whilst

```
ts_interim: psi-wan
ts_communities: psi-wan int-x25
```

would be used for hosts on both the International X.25 and the private X.25 network. Again, it is critical that the definition of `ts_interim` occur before the definition of `ts_communities`.

8.2.2 Heuristic Support

Finally, there are some cases when the ISODE must look at a simple string and derive an address. If the string encoding for presentation address defined in [Kil89a], then the macro defined earlier will help. But, if a simpler string is used, e.g.,

```
% ftam gonzo
```

or

```
% ftam 00000511160013
```

then it would be nice to have an intelligent default to use for the community associated with these. There are three run-time tailoring variables provided for this purpose:

if it looks like	**then use**
an OSI NSAP	`default_nsap_community`
a X.25 DTE	`default_x25_community`
a TCP address	`default_tcp_community`

To continue our two examples of private communities. for the private TCP/IP LAN, it would make sense to add

```
default_tcp_community: nott-ether
```

to the *isotailor* file on each host. For the private X.25–based community, it would make sense to add

```
default_x25_community: psi-wan
```

instead.

8.3 Transport-Service Bridges

There is one last question which must be considered: how can two hosts communicate if they do not have any communities in common? If there is a third host, which shares a community with the two other hosts, then a *TS-bridge* can be used to achieve connectivity on a per-connection basis.

8.3.1 Client Hosts

For each host, examine the value of the `ts_communities` run-time variable. For those communities which are not listed, but which you wish that host to be able to communicate with, you must add the definition of the appropriate TS-bridge to the `tsb_communities` run-time tailoring variable. A definition consists of two tokens, the name of the community that the TS-bridge can reach, and the transport address of the TS-bridge (using the string encoding defined in [Kil89a]), e.g.,

```
tsb_communities: internet Int-X25(80)=31344152401010+PID+03018000
```

This says that the capital-I Internet community can be reached by contacting the TS-bridge located on the International X.25 at the specified DTE and PID.

It is **critical** to observe that the TS-bridge must share a common community with the host containing this definition. Thus, a more instructive example is:

```
ts_communities: int-x25 internet
tsb_communities: internet Int-X25(80)=31344152401010+PID+03018000
```

which describes a host connected to the International X.25 that may wish to talk to hosts in the capital-I Internet community.

Similarly, one could imagine a host belonging to the capital-I Internet community using the following definitions:

```
ts_communities: internet int-x25
tsb_communities: int-x25 Internet=gonzo.twg.com+17004
```

8.3.2 Server Hosts

Typically, the entity responsible for a community also runs a TS-bridge which connects that community to the other Interim communities. Obviously, this host must support each of the Interim communities to be connected to the local community. The `tsb_default_address` run-time tailoring variable is used to define the transport address (usually containing multiple network addresses) where the TS-bridge listens.

So, to continue with the nott-ether example, the host on which the TS-bridge resides might have these definitions:

```
ts_interim:      nott-ether
ts_communities:      nott-ether int-x25
tsb_default_address: Nott-Ether=sheriff+17004"—Int-X25(80)=23426020017299+PID+03018000
```

The rule is simple, for each community named in the `ts_communities` run-time tailoring variable, a network address is present in the run-time tailoring variable `tsb_default_address`. By convention, for TCP-based addresses, TCP port 17004 is used, whilst for X.25–based addresses, PID 0301800 is used.

8.4 In Retrospect

What a mess! There's a lot to be said for the focused "one transport protocol, one network service" approach used by the Internet suite of protocols.

Bibliography

[CCI84a] Session Protocol Specification for Open Systems Interconnection (OSI) for CCITT Applications. International Telegraph and Telephone Consultative Committee, October 1984. Recommendation X.225.

[CCI84b] Session Service Definition for Open Systems Interconnection (OSI) for CCITT Applications. International Telegraph and Telephone Consultative Committee, October 1984. Recommendation X.224.

[CCI84c] Transport Service Definition for Open Systems Interconnection (OSI) for CCITT Applications. International Telegraph and Telephone Consultative Committee, October 1984. Recommendation X.214.

[ISO86] Information Processing Systems — Open Systems Interconnection — Transport Service Definition. International Organization for Standardization and International Electrotechnical Committee, 1986. International Standard 8072.

[ISO87a] Information Processing — Open Systems Interconnection — Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1). International Organization for Standardization and International Electrotechnical Committee, 1987. International Standard 8825.

[ISO87b] Information Processing Systems — Open Systems Interconnection — Basic Connection Oriented Session Protocol Specification. International Organization for Standardization and International

Electrotechnical Committee, August 1987. International Standard 8327.

[ISO87c] Information Processing Systems — Open Systems Interconnection — Basic Connection Oriented Session Service Definition — Addendum 2: Incorporation of Unlimited User Data. International Organization for Standardization and International Electrotechnical Committee, August 1987. Draft Addendum 8326/DAD 2.

[ISO87d] Information Processing Systems — Open Systems Interconnection — Basic Connection Oriented Session Protocol Specification — Addendum 2: Incorporation of Unlimited User Data. International Organization for Standardization and International Electrotechnical Committee, August 1987. Draft Addendum 8327/DAD 2.

[ISO87e] Information Processing Systems — Open Systems Interconnection — Basic Connection Oriented Session Service Definition. International Organization for Standardization and International Electrotechnical Committee, August 1987. International Standard 8326.

[ISO88a] Information Processing Systems — Open Systems Interconnection — Connection Oriented Presentation Service Definition. International Organization for Standardization and International Electrotechnical Committee, April 1988. Final Text of Draft International Standard 8822.

[ISO88b] Information Processing Systems — Open Systems Interconnection — Connection Oriented Presentation Protocol Specification. International Organization for Standardization and International Electrotechnical Committee, April 1988. Final Text of Draft International Standard 8823.

[Kil89a] Stephen E. Kille. A string encoding of Presentation Address. Research Note RN/89/14, Department of Computer Science, University College London, February 1989.

[Kil89b] Stephen E. Kille. An interim approach to use of Network Addresses. Research Note RN/89/13, Department of Computer Science, University College London, February 1989.

[KR78] Brian W. Kernighan and Dennis M. Ritchie. *The C Programming Language*. Software Series. Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1978.

[Pad85] Michael A. Padlipksy. Gateways, Architectures, and Heffalumps. In *The Elements of Networking Style and Other Essays and Animadversions on the Art of Intercomputer Networking*, chapter 10, pages 167–176. Prentice-hall, Englewood Cliffs, New Jersey, 1985. Also available as Internet Request for Comments 875.

[Pos81] Jon B. Postel. Transmission Control Protocol. Request for Comments 793, DDN Network Information Center, SRI International, September 1981. See also MIL-STD 1778.

[RC86] Marshall T. Rose and Dwight E. Cass. OSI Transport Services on top of the TCP. *Computer Networks and ISDN Systems*, 12(3), 1986. Also available as NRTC Technical Paper #700.

[RC87] Marshall T. Rose and Dwight E. Cass. ISO Transport Services on top of the TCP. Request for Comments 1006, DDN Network Information Center, SRI International, May 1987.

[Ros90] Marshall T. Rose. *The Open Book: A Practical Perspective on Open Systems Interconnection.* Prentice-hall, 1990. ISBN 0–13–643016–3.

Index

A
addr2ref, 22
aetbuild, 10
Aziz, Ashar, xix

B
Betzler, Uli, xx
Braun, Hans-Werner, xviii
Brezak, John, xix

C
Cass, Dwight E., xvi
Chirieleison, Don, xix
Cowin, Godfrey, xvii

D
dish, 9
DLOG, 162
dotoken, 48
dotokens, 48
DR_FATAL, 115
DR_OFFICIAL, 115
dsabuild, 10
Dubous, Olivier, xviii

E
Easterbrook, Stephen, xvii
EEC, xvii
endisoservent, 143
ESPRIT, xvii
execve, 123, 142

F
Finni, Olli, xviii

G
getisoservent, 143
getisoserventbyname, 143
getisoserventbyport, 144
getisoserventbyselector, 144
getpid, 148
Gosling, James, xx

H
Heinänen, Juha, xix
Horton, Mark R., xviii
Horvath, Nandor, xix

I
is2paddr, 16
is2saddr, 42
is2taddr, 102
_isodefile, 155
isodesetailor, 154
isodesetvar, 154
isodetailor, 154
isodexport, 155
isoentities, 16, 42, 102
isomacros, 170, 171
isoservent, 142
isoservices, 133
isotailor, 167, 170, 171

J
Jacobsen, Ole-Jorgen, xix
Jelfs, Philip B., xix