Accredited Standards Committee
X3, INFORMATION PROCESSING SYSTEMS*

Doc. No.: **X3S3.3/90-342R**
Date: March 10, 1991
Project: JTC 1.6.41.4
Ref. Doc.: DIS 10589
Reply to: Dave Oran
Digital Equipment Corp
LKG 1-2/A19
550 King Street
Littleton, MA 01460
(508) 486-7377

To: Members and Alternates of X3S3.3

From: Dave Oran, Digital Equipment (DIS 10589 Editor)

Subject: Draft U.S.A. Comments on DIS 10589

The attached comments are to be submitted to X3S3 for approval along with a recommendation from X3S3.3 that the U.S. vote **YES with Comments** on the DIS 10589 ballot.

Comment 1: Bug

There is a security hole in the existing description of PDU acceptance tests. The current behaviour allows a misconfigured IS to transmit PDUs containing an Authentication Information field with an Authentication Type other than "Password" and have those PDUs processed even though the recipient IS does not implement the authentication type requested. As an example, see the following text in clause 8.2.4.1, item c2:

> "If the PDU contains the Authentication Information field, but the Authentication Type is not equal to "Cleartext Password", then the PDU shall be accepted unless the IS implements the authenticatiion procedure indicated by the Authentication Type. In this case whether the IS accepts or ignores the PDU is outside the scope of this International Standard."

Proposed Correction:

Replace the text in the following places:

a) Clause 7.3.15.1, items a3ii and a4ii
b) Clause 7.3.15.2, items a3ii and a4ii
c) Clause 8.2.4.1, item c2
d) Clause 8.4.1.1, item c2

with the following:

> "If the PDU contains the Authentication Information field, but the Authentication Type is not equal to "Password", then:
>
> a) If the IS implements the authenticatiion procedure indicated by the Authentication Type whether the IS accepts or ignores the PDU is outside the scope of this International Standard.
>
> b) If the IS does not implement the authentication procedure indicated by the Authentication Type then the IS shall ignore the PDU and generate an authenticationFailure notification."

Comment 2: Minor

DIS10589 makes extensive use of the abstract quantity systemID in its description of the behaviour of ISs. The meaning of this quantity is quite clear in the specification; it is the ID portion of the Network Entity Title assigned to the IS. However, the specification is less than helpful for how this quantity is initially established when the DIS10589 protocol machine is enabled. The following possibilities have occurred to us:

a) The value might be an explicit attribute of the Network Entity manged object, instantiated under the DIS10589 conditional package, and called systemID.

b) the value might be derived from the Network Entiry Title attribute of the Network entity Managed Object .

If solution (a) is adopted no attribute is needed to specify the idLength. If solution (b) is adopted, the standard must specify exactly *how* the systemID is derived (even if that is simply "a local matter"), and must specify some method by which the correct idLength is ascertained. On possibilitiy is to add yet another attribute of the Network Entity managed object which contains the routeing domain's idLength.

While the US has no strong opinion on the solution chosen, in the absence of another suggestion the US believes that explicit attributes for both systemID and idLength should be provided. Whether these attributes are read/write or read only should be an implementation option.

With the above in mind, we offer the following proposed text for the GDMO definitions:

Proposed Correction:

In the level1ISO10589Package, add the following attributes after isType in the middle of the left-hand column of page 69:

systemID **ATTRIBUTE**
WITH ATTRIBUTE SYNTAX ISO 10589-ISIS-SystemID;
MATCHES FOR Equality;
BEHAVIOUR systemID-B **BEHAVIOUR DEFINED AS**
The ID for the local system, to be appended to each of the system's area address(es) to form the Network Entity Titles valid for this IS. The derivation a value for systemID is a local matter.;;
REGISTERED AS {ISO10589-ISIS.aoi systemID (TBD)};

and in the ASN.1 sysntax module, add the following line:

systemID ::= **OCTETSTRING** (SIZE(1..8))

Comment 3: Bug

ISs have no need to listen to ISO9542 ISH PDUs. Clause 8.4.4, page 46, item c, and clause 10.3, item a, both say that they do.

Proposed Correction:

Reword clause 8.4.4, page 46, item c, as follows:

"Start listening for ISO 9542 ESH PDUs and acquire adjacencies as appropriate. Do not run the Designated Intermediate System election process."

Reword clause 10.3, item a, on page 66 as follows:

"The IS shall operate the Configuration Information functions on all types of subnetworks supported by the IS. This includes the reception of ESH PDUs, and the transmission of ISH PDUs."

Comment 4: Editorial

The material on what an IS does when enabling a broadcast circuit is "buried" in the middle of the material on LAN Designated ISs on page 46. We suggest moving most of the paragraph beginning "When the broadcast circuit is enabled" into a separate subclause as the first subclause of 8.4.

Proposed Correction:

Create a new subclause 8.4.1 from the material on page 46, as follows and renumber the subsequent sub-clauses:

8.4.1 Enabling of Broadcast Circuits

When a broadcast circuit is enabled on an Intermediate system the IS shall perform the following actions.

a) Commence sending IIH PDUs with the LAN ID field set to the concatenation of its own systemID and its locally assigned one octet Local Circuit ID.

b) Solicit the End system configuration as described in 8.4.5.

c) Start listening for ISO 9542 ESH PDUs and acquire adjacencies as appropriate. Do not run the Designated Intermediate System election process.

d) After waiting iSISHelloTimer * 2 seconds, run the Level 1 and or the Level 2 Designated Intermediate System election process depending on the Intermediate system type.

After deleting the above text from clause 8.4.4, move item (d) out of the list and reword the beginning of this item as follows:

> "Run the Level 1 and or the Level 2 Designated Intermediate System election process (depending on the Intermediate system type) whenever an IIH PDU is received or transmitted as described in . (For these purposes, the transmission of the system's own IIH PDU is equivalent to receiving it). If there has been no change to the information on which the election is performed since the last time it was run, the previous result can be assumed. The relevant information is:
>
> a) the set of Intermediate system adjacency states;
>
> b) the set of Intermediate System priorities (including this system's); and
>
> c) the existence (or otherwise) of at least one "Up" End system (not including Manual Adjacencies) or Intermediate system adjacency on the circuit.

Comment 5: Bug

Various places in Annex C were not updated after the decision to permit variable size ID fields. For example, the definition of lspId is **ARRAY** [0..7] **OF** Octet, when it should be **ARRAY** [0..idLength–1] **OF** Octet. All instances of these need to be fixed.

Comment 6: Editorial

The datastructure NETEntry in the adjacency database is not necessary and should be removed, along with the reference to it in the text above. This was a holdover from a version of the specification which described various aspects of End system routing.

Comment 7: Editorial

|The description of Manual Area Addresses in the first paragraph of clause 7.1.3 is misleading. It leads the reader to believe that one of the ISs addresses is "more equal" than the others, having been derived from its assigned NET. In fact any and all of the IS's area addresses can be combined with its ID to form valid NETs for the IS. This means, for example, that any of these addresses can appear in ISH or RD PDUs sent to End systems. None of this will confuse an End system, but the lack of a restriction limiting an IS to a single NET when it has multiple area addresses should be highlighted.

Proposed Correction:

Replace the first paragraph of clause 7.1.3 with the following:

> "The use of several synonymous area addresses by an IS is accommodated through the use of the management parameter manualAreaAddresses. This parameter is set locally for each level 1 IS by system management; it contains a list of all synonymous area addresses associated with the IS. All of the IS's manualAreaAddresses, when combined with the IS's systemID, are valid network entity titles for the IS."

Comment 8: Minor

As pointed out in comment 7 above, an IS may have more than one NET as a consequence of it being assigned more than one manualAreaAddress. Given that this is the case, there should be some guidance in the specification on the generation of ISO 9542 Redirects. Otherwise, NETs might be chosen differently by different ISs, resulting in unnecessary redirect timeouts and increased memory utilisation in End systems.

We suggest that DIS 10589 specify that the IS always specify the *numerically lowest* NET for the target IS. The requirement should be included in clause 10.3, where requirements on ISO 95452 operation are presented.

Proposed Correction:

Add an item c) to the list of requirements on the operation of ISO 9542, as shown below:

> "c) When sending ISH PDUs, or redirecting an End system to a neighbour Intermediate system, as described in clause 7.4.3.3, the numerically lowest Network Entity title shall be chosen as the NET placed in the ISO 9542 PDU." This minimises memory usage in the End systems by ensuring that all ISs identify themselves and each other to ESs using the same Network Entity titles."

Comment 9: Minor

DIS10589 defines the architectural constant MaxPathMetric to bound the maximum path cost of any path used by the forwarding process. It does this to permit an implementation perform an SPF calculation using the "binning" optimisations described in Annex C. It is conceivable that an implementation would not wish to do this optimisation and as a side effect would be capable of representing routes longer than MaxPathMetric. On the other hand, such an implementation would not interwork properly with an implenetation enforcing MaxPathMetric. Unfortunately, aside from defining the constant, there is no normative text in DIS10589 to constrain implementations to obey the value of MaxPathMetric. We therefore suggest that explicit normative text be added to clause 7.2.6.

Proposed Correction:

In clause 7.2.6, on page 13, right column, insert the following sentence immediately following the sentence that reads "Paths which do not meet the above conditions are illegal and shall not be used.":

> "Paths whose metric sum exceeds the value of the architectural constant MaxPathMetric (see table 2) are also illegal and shall not be used."

Comment 10: Minor

The level 2 forwarding process depends upon matching an NSAP address with either an address prefix or an area address. However, DIS 10589 has no normative text to describe their generation and the method for comparing them to a given destination NSAP address. Consequently, it also does not mention these processes in its conformance section. The USA has found the material in informative Annex B.1 to be more confusing than helpful.

Therefore, we recommend that Annex B.1 be deleted, and that the following amendments be made to the body of DIS 10589, and that conformance clause 12.1.2 be updated to call out the new clauses explicitly.

Proposed Correction:

a) Replace clause 7.1.4 with the following text, which explicitly defines the encoding processes for area addresses and for address prefixes, and has a title which better reflects its contents:

> **7.1.4 Encoding Of Addressing Information**
>
> This international standard makes use of four types of address information: NETs, NSAPs, area addresses, and address prefixes. The encoding rules for each of them are given below.
>
> 1. NETs shall be encoded according to the preferred binary encoding specified in ISO 8348/Add.2.
>
> 2. NSAPs shall be encoded according to the preferred binary encoding specified in ISO 8348/Add.2.
>
> 3. The encoded form of an area address shall be obtained by dropping the last "n + 1" octets of the preferred binary encoding of the corresponding NSAP, where "n" is equal to the length of the ID field used by the routeing domain.

4. The encoded form of an address prefix shall be obtained by encoding the prefix (expressed in its abstract syntax), according to the preferred binary encoding, unless the end of the prefix falls within the IDP. In this case, each decimal digit in the prefix shall be encoded as the corresponding semi-octet in the range 0000-1001 and no padding characters shall be inserted.

b) Inset a new normative clause that defines the matching processes, using the following suggested text:

#.#.# Matching an NSAP Address with an Area Address or an Address Prefix

A destination NSAP address can be matched against either an area address or an address prefix. For an area address or for an address prefix which extends into the DSP, it shall be compared directly against the encoded NSAP address, including any padding characters that may be present; for an address prefix which does not extend into the DSP, it shall be compared against NSAP', which is obtained from the encoded NSAP address by removing all padding characters that were inserted by the binary encoding.

The existence of a match shall be determined as follows:

1. If the encoded NSAP (or NSAP') contains fewer semi-octets than the encoded area address (or address prefix), then there is no match.

2. If the NSAP (or NSAP') contains at least as many octets as the area address (or address prefix), and all octets of the encoded area address (or address prefix) are identical to the corresponding leading octets of the encoded NSAP address (or NSAP'), there is a match. Otherwise, there is no match.

NOTE: Any implementation of a matching process that satisfies the requirements listed above may be used. The key point is that matching process must be aware of whether or not the encoded area address or address prefix extends into the DSP, and must then either include or exclude padding characters from the encoded NSAP, as defined above.

Comment 11: Minor

DIS10589 provides the capability of associating manually configured routing information with a circuit via Reachable Address managed objects (RAMOs). RAMOs define sets of NSAPs (using NSAP address prefixes) that can be reached over a given circuit. In the case of multi-destination circuits (e.g., Dynamic Assignment 8208, Broadcast) it is necessary to determine a specific next hop SNPA for NPDUs forwarded based upon RAMOs. In general, this process takes the form of a function that maps the destination NSAP of the NPDU to be forwarded onto a SNPA address of the next hop.

The current text suggests two basic mechanisms for performing such mappings: manual and algorithmic. The procedures associated with RAMOs with mappingType "Manual" are clearly specified in sections 8.1, 8.3.2, 8.3.5 and 11.2. The description of RAMOs that employ algorithmic next hop SNPA mapping (see 8.1 and 8.3.2) suggests that this could be a very general facility ("`mappingType may specify the name of an SNPA address extraction algorithm`"). Currently though, only one extraction algorithm "X121" (extraction of an X.121 address from the IDI of a NSAP from the X.121 address subdomain) is described.

It is assumed that the direct embedding of all, or part, of subnetwork addresses in destination NSAPs will continue to be a common technique in addressing and routing plans. While useful in X.25 environments in which addresses may be allocated from the X.121 address subdomain, the current "X121" mappingType is not useful in other very common scenarios in which next hop SNPA addressing information is embedded in NSAPs. What follows is a proposal to generalize the existing X121 algorithm and to add another general purpose embedded SNPA extraction algorithm to the base text.

First, the "X121" algorithm described in the text should be generalized to address the other addressing subdomains in ISO8348/Add2 in which SNPAs are encoded as the IDI of a NSAP. The algorithm would be applicable to address prefixes from the X.121, F.69, E.163 and E.164 sub-domains. Since the ISO 8348/Add3 IDI encoding scheme follows the same conventions for each of these sub-domains, this change is mostly editorial in nature. The name and description of this SNPA mapping type should be generalized.

Second, a general purpose DSP embedded SNPA extraction algorithm should be added to the text. The scenario to be addressed is the embedding of all or part of the next hop SNPA in an arbitrary position within the DSP of the destination NSAP. This new facility requires the definition of a new mappingType called "extractDSP". Add to Reachable Address Managed Objects a new conditional package with attributes:

sNPAPrefix:	Manually configured fixed prefix of SNPA.
sNPAMask:	Bit string indicating SNPA position within DSP.

The sNPAPrefix and sNPAMask provide support for scenarios in which a portion of a SNPA is embedded in the destination NSAP DSP. This facility is useful when the SNPA addresses on a given circuit have a common fixed part and a part the varies with each system. The fixed part could be configured into the sNPAPrefix and the variable part embedded with NSAPs. The sNPAMask indicates the position of the variable part within the NSAP DSP. If the complete SNPA is embedded in the NSAP DSP, the sNPAPrefix would be null.

Note that this proposal assumes that in scenarios in which only a protion of the subnetwork addressing information is embedded in the DSP, that protion would form a suffix of the complete SNPA address. Although this capability addresses most known scenarios, it is conceivable that more general support could be required. A more general form of this proposal could be formed by replacing the sNPAPrefix with a more general fixed part sNPATemplate.

Proposed Correction:

In the second paragraph of clause 6.3 (Topologies) generalize the description of algorithmic mappings so as not to limit their application to dynamicly assigned DEDs. Replace the phrase:

> “Where the subnetwork to which this SNPA is connected is a general topology subnetwork supporting dynamicly established data links, ...”

with

> “Where this SNPA is connected to a multi-destination subnetwork (e.g., dynamicly assigned DED, broadcast), ...”

Since the use of reachable addresses is not limited to DA 8208 circuits, the examples included in the second and third paragraphs of section 8.3.2.2 should be moved to the general description at the end of section 8.1 (Multi-destination Circuits on ISs at a Domain Boundary). Revise the third paragraph of 8.1, incorporating the examples of section 8.3.2.2 as follows:

> “This is achieved by additional information contained in the reachable address managed object. The mappingType attribute specifies the means by which next hop subnetwork addressing information can be derived for NPDUs forwarded based upon a given address prefix. The mappingType attribute may be specified as:
>
> manual — The SNPA address or set of SNPA addresses is manually pre-configured as an attribute of the reachable address managed object.
>
> extractIDI – The SNPA is embedded in the IDI of the destination NSAP address according to the format and encoding rules of ISO8348/Add2. This SNPA extraction algorithm can be used in conjunction with destination addresses from the X.121, F.69, E.163, and E.164 addressing subdomains.
>
> extractDSP — All or a suffix of the SNPA is embedded in the DSP of the destination address. This SNPA extraction algorithm requires manual pre-configuration of sNPAMask and sNPAPrefix attributes of the reachable address managed object. The sNPAMask attribute is a bit mask with 1s indicating the location of the SNPA (suffix) within the destination NSAP DSP. The part of the SNPA extracted from the NSAP is appended to the sNPAPrefix to form the next hop subnetwork addressing information.
>
> An example of a set of Reachable Addresses is show in Table 8.

Table 8 - Example of Reachable Address Information

Address Prefix	Mapping Type	SNPA Address
39 123	manual	X
37 aaaa	manual	B
37 D	manual	Y
37	extractIDI	Extract X.121 SNPA from NSAP IDI
47 0005 C0	extractDSP	sNPAPrefix=Z sNPAMask=00000000FFFFFFFFFFFF
*	manual	R, S, T

Include the paragraph containing the explanation of Table 8. Since the examples are no longer specific to DA SVC establishment, change references to "*calling*" SNPA addresses to "using". Insert a new item e) below, rel-able existing item e).

e) For the ISO ICD prefix `47 0005 C0` use the SNPA address formed by concatenating Z with next 6 octets of the DSP following the `47 0005 C0` prefix.

Minor Editorial Changes for Consistency:

Having moved from 8.3.2.2 (see above) the example of Table 8, this section shall consist of the first paragraph and the note at the end of the section. Generalize the first paragraph to address the fact that both IDI and DSP embedded SNPAs are possible.

In the remaining paragraph of this section and sections 8.3.2.3, 8.3.3, 8.3.5.2, 8.3.5.3, 8.3.5.4 generalize the text with respect to the new SNPA extraction algorithm. In particular:

a) 8.3.2.2, 8.3.2.3, 8.3.3: Change references to extracting SNPAs specificly from the "IDP" to a more general reference to the "destination NSAP".

b) 8.3.5.1, 8.3.5.3, 8.3.5.4: Change references to the specific "mappingType X121" to the more general reference to "algorithmic extraction mappingType".

11.2.10 The Reachable Address Managed Object

To the reachableAddress MANAGED OBJECT CLASS add a new conditional package as follows:

extractDSPPackage **PRESENT IF** the value of mappingType is "extractDSP"

Add the coresponding package definition:

extract DSPPackage **PACKAGE**
  BEHAVIOUR DEFINITIONS extractDSPPackage-B
  BEHAVIOUR DEFINED AS
    When present, the remote SNPA address is determined by extracting the bits from the effective NSAP address indicated by 1's in the sNPAMask and concatenating them to the sNPAPrefix.;;
  ATTRIBUTES
    sNPAMask
      REPLACE-WITH-DEFAULT
      DEFAULT-VALUE ISO10589-ISIS.sNPAMask-Default
      GET-REPLACE,
    sNPAPrefix
      REPLACE-WITH-DEFAULT
      DEFAULT-VALUE ISO10589-ISIS.sNPAPrefix-Default
      GET-REPLACE,
  REGISTERED AS {ISO10589-ISIS.poi extractDSPPackage (TBD)};

11.2.11 Attribute Definitions

Revise the behaviour definition of the mappingType attribute to incorporate new algorithm descriptions. Also remove the restriction that LAN circuits only use the manual mapping type.

Replace the mappingType-B behaviour definition with the following:

BEHAVIOUR DEFINED AS
The type of mapping to be employed to ascertain the SNPA Address which should be used in forwarding NPDUs for this Reachable Address Prefix. The following values of mappingType are defined:
manual — The set of subnetwork addresses in the sNPAAddresses or lANAddress attribute are to be used.
extractIDI — The SNPA is embedded in the IDI of the destination NSAP address. The mapping algorithm extracts the SNPA to be used according to the format and encoding rules of ISO8348/Add2. This SNPA extraction algorithm can be used in conjunction with Reachable Address Prefixes from the X.121, F.69, E.163, and E.164 addressing subdomains.
extractDSP— All, or a suffix, of the SNPA is embedded in the DSP of the destination address. This SNPA extraction algorithm extracts the embedded subnetwork addressing information by performing a logical AND of the sNPAMask attribute with the destinaition address. The part of the SNPA extracted from the destination NSAP is appended to the sNPAPrefix to form the next hop subnetwork addressing information.

Add attribute definitions for sNPAMask and sNPAPrefix as follows:

sNPAMask **ATTRIBUTE**
WITH ATTRIBUTE SYNTAX ISO 10589-ISIS.NSAPAddress;
MATCHES FOR Equality;
BEHAVIOUR sNPAMask-B **BEHAVIOUR DEFINED AS**
A Bit mask with 1 bits indicating the positions in the effective destination address from which embedded SNPA information is to be extracted. For the extraction the first octect of the sNPAMask is aligned with the first octet (AFI) of the NSAP Address. If the sNPAMask and NSAP Address are of different lengths, the shorter of the two is logically padded with zeros before performing the extraction;;
REGISTERED AS {ISO10589-ISIS.aoi sNPAMask (TBD)};

sNPAPrefix **ATTRIBUTE**
WITH ATTRIBUTE SYNTAX ISO 10589-ISIS.binarySNPAAddress;
MATCHES FOR Equality;
BEHAVIOUR sNPAPrefix-B **BEHAVIOUR DEFINED AS**
A fixed SNPA prefix manually configured as an attribute of a Reachable Address with mappingType extractDSP. The SNPA address to use is formed by concatenating the fixed sNPAPrefix with a variable SNPA part that is extracted from the effective destination address. For Reachable Address Prefixes in which then entire SNPA is embedded in the DSP the sNPAPrefix shall be null;;
REGISTERED AS {ISO10589-ISIS.aoi sNPAPrefix (TBD)};

11.2.17 ASN1 Modules

Define the new mappingType and rename the existing X121 type as follows:

MappingType ::= **ENUMERATED**{manual(0), extractIDI(1), extractDSP(2)}

Add appropriate type definitions as follows:

binarySNPAAddress ::= **SEMI-OCTETSTRING** (SIZE(0..15))
sNPAPrefix-Default ::= NULL -- Note: Zero length prefix
sNPAMask-Default ::= 0 -- Note: Should un-initialized NSAPs should have length = 0?

Comment 12: Minor

Some readers of the specification may be confused by the differing requirements on the usage of area addresses and reachable address prefixes. Some have particularly stumbled on the (unstated, but easily inferred) requirement that no reachable address prefix match the area address of an area in the routing domain and also be as long as or longer than that area address. If this were permitted, some systems would be simultaneously inside and outside the area (and the routing domain). A simple clarifying note in clause 7.1.2 should clear up this potential source of confusion.

Proposed Correction:

Add the following note under item a3:

> “NOTE — a consequence of this requirement is that a reachable address prefix may not match any area address of an area in the routeing domain.”

Comment 13: Minor

The normative material in clause 7.2.12 is not current referenced in the conformance clause.

Proposed Correction:

Replace item c) uder clause 12.1.2 with:

> “Selection of paths according to 7.2.7 and 7.2.12”

Comment 14: Minor

Protocol Implementation Conformance Statements (PICS) are used in a number of ways, including:

a) as a check-list to implementors;

b) as a detailed indication of the capabilities of an implementation for suppliers, acquisition authorities, and users; and,

c) as a descriptive device between implementors and protocol testing agents for the purposes of test case selection, documentation and cross reference of abstract test cases, and reference in Protocol Conformance Test Reports (PCTR).

To fulfill these possible uses, PICS must embody a reasonably fine level of granularity with respect to basic protocol functions, timers, parameters, and PDUs. The OSI Conformance Testing Methodology and Framework [DIS9646] and recent direct guidance from SC21 [SC6/N6113] state that:

a) Each PICS proforma is required to list all capabilities (including mandatory ones) and all PDUs.

b) It must be possible to answer "NO" to any question, including those for mandatory features.

c) If broad (global) questions of conformance are included, the PICSs must still list all mandatory capabilities individually so as to allow clear identification of which features have been implemented.

Review of the PICS contained in Annex A of DIS10589 reveals that, while technically correct, it does not contain a suitable level of detail to satisfy its potential uses (particularilly in the area of testing) nor the requirements of SC21.

Proposed Correction:

Note: The US has not performed a detailed review of the technical content of this replacement text. The proposed text is included at this time so as to permit early review by other member bodies.

See Annex A of these comments for proposed replacement PICS tables for the protocol summary section (pp 103-104) .

Comment 15: Editorial

In clause 6.3, the word “administrative domains” in the last sentence of the first paragraph should be changed to “routeing domains”. Since the inter-domain routeing function operates between routeing domains (not explicitly between administrative domains), the suggested change will be a more precise statement of the desired cooperation between the two routeing protocols.

Comment 16: Editorial

The requirement for all systems in a routeing domain to use the same length for their ID fields should be included in 7.1.2. although 7.1.1 defines how an IS shall parse the address information, 7.1.2 does not now contain a complementary requirement for the deployment of systems.

Proposed Correction:

Add the following text as a new item a4:

> “All systems located within a given routeing domain must have NETs or NSAPs whose ID fields are of equal length.”

Comment 17: Editorial

In clause 7.2.2 on page 10 the two paragraphs after the NOTE talk about “reporting” or not “reporting” a metric. For consistency with the LSP encodings of clause 9, it would be preferable to rephrase these sentences along the lines of “supporting” or “not supporting” a given metric. This ties in better with the “S” bit of the LSP encodings.

Comment 18: Editorial

The second sentence of clause 7.2.10.1 on page 15 is worded in a way that could imply that there could be “level 2 only” ISs, which we decided not to invent.

Proposed Correction:

Reword the sentence as follows:

> “Participation in the partition repair process by a Level 2 Intermediate system is predicated on the fact that all L2 ISs also function as L1 ISs within their own area”

Comment 19: Editorial

The following are miscellaneous typographical, grammatical, and spelling errors which should be corrected:

a) Title — The title syould be amended from “...routeing exchange protocol...” to “...routeing information exchange protocol...”

b) Clause 2.1 — abbreviaion “IPS-T&IEBS” should be expanded to “Information Technology — Telecommunications and Information exchange between systems”

c) Clause 6.8.1.4, page 9 — Change “NPID” to “NLPID”

d) Clause 7.1.5, page 11 — in the last sentence “precedure” ⇒ “procedure”.

e) Clause 7.2.3, page 12 — in the 2nd line of the 4th paragraph: “it’s” ⇒ “its”.

f) NOTE on page 18, 1st column — “and an event signalled...” ⇒ “and it shall should signal an event...”

g) clause 7.3.3.1, last line of first paragraph — “managments” ⇒ “management”

h) Clause 7.3.11, page 22 — Delete “(i.e., the first six octets)”

i) Table 8 on page 40 — in the second row move “123” from the second to the first column

j) In the NOTE at end of clause 8.3.2.2 on page 40, remove the words “DCM or”

k) IS Neighbours field length on the bottom of page 43 — “11” ⇒ “(IDLength + 5)

l) ES Neighbours field length on page 55 “multiple of 6” ⇒ “multiple of (IDLength + 1)”

Annex A
Proposed PICS Protocol Summary Tables

A.4.2 Protocol Summary: ISO 10589:19xx General

Item	Functionality/Description	References	Status	Support N/A
AllIS	Are all basic IS–IS routeing functions implemented?	12.1.2	M	Yes □
System Managmeent	Is the system capable of being managed by the specified management information?	11	M	Yes □
Authentication	Is PDU authentication based on passwords implemented?	7.3.7-7.3.10, 7.3.15.1-7.3.15.4, 8.2.3-8.2.4, 8.4.1.1	M	Yes □ No □
Default Metric	Is the default metric supported?	7.2.2, 7.2.6	M	Yes □
Delay Metric	Is the delay metric supported?	7.2.2, 7.2.6	O	Yes □ No □
Expense Metric	Is the expense metric supported?	7.2.2, 7.2.6	O	Yes □ No □
Error Metric	Is the error metric supported?	7.2.2, 7.2.6	O	Yes □ No □
ID Field Length	What values of RoutingDomainIDLength are supported by this implementation? Is the value configurable by system management?	7.1.1	M	values = Yes □ No □
Forwarding Rate	How many ISO 8473 PDUs can the implementation forward per second?	12.2.5.1.b	M	PDUs/sec =
Performance	Are the implementation performance criteria met?	12.2.5	M	Yes □

System Environment: General

Item	Functionality/Description	References	Status	Support N/A
ISO9542	Are the appropriate ISO 9542 operations implemented	10.3, 8.2.1-8.2.2, 8.3.4, 8.4.5, 8.4.6	M	Yes □
Timer Jitter	Is jitter introduced in all periodic timers whose expiration causes transmission of a PDU?	10.1	M	Yes □

Subnetwork Dependent Functions: General

Item	Functionality/Description	References	Status	Support N/A
*LAN	Are the subnetwork dependent functions for broadcast subnetworks implemented?	8.4	O.1	Yes ☐ No ☐
LAN IS Adjacencies	Are the LAN IS adjacency establishment operations implemented?	8.4.1-8.4.3	LAN: M	☐ Yes ☐
LAN ES Adjacencies	Are the LAN ES adjacency establishment operations implemented?	8.4.6	LAN: M	☐ Yes ☐
LAN DIS	Are the LAN designated IS operations implemented?	8.4.4, 8.4.5	LAN: M	☐ Yes ☐
*8208 Static	Are the subnetwork dependent functions for ISO 8208 subnetworks implemented?	8.3	O.1	Yes ☐ No ☐
8208 SNDCF	Are the ISO8208 Subnetwork Dependent Convergence Functions implemented?	8.3.1, 8.3.2.1	C.1: M	☐ Yes ☐
*PtPt	Are the subnetwork dependent functions for point-to-point subnetworks implemented?	8.2	O.1	Yes ☐ No ☐
PtPt IS Adjacencies	Are the point-to-point IS adjacency establishment operations implemented?	8.2.2-8.2.5	C.2: M	☐ Yes ☐
PtPt ES Adjacencies	Are the point-to-point ES adjacency establishment operations implemented?	8.2.1	C.2: M	☐ Yes ☐
PtPt IIH PDU	Are point-to-point IIH PDUs correctly constructed and parsed?	9.7	C.2: M	☐ Yes ☐

C.1: if 8208 Static or 8208 DA then M else –

C.2: if PtPt or 8208 Static then M else –

Update Process: General

Item	Functionality/Description	References	Status	Support N/A
LSP Periodic Generation	Is periodic generation of new local LSPs implemented?	7.3.2, 7.3.5, 7.3.13	M	Yes □
LSP Event Driven Generation	Is event driven generation of new local LSPs implemented?	7.3.6	M	Yes □
Pseudonode LSP Generation	Is generation of pseudo node LSPs implemented?	7.3.8, 7.3.10	LAN: M	□ Yes □
Multiple LSP Generation	IS multiple LSP generation implemented?	7.3.4	M	Yes □
LSP Propagation	Is propagation of LSPs implemented?	7.3.12, 7.3.14, 7.3.15.1, 7.3.15.5	M	Yes □
LSP Lifetime Control	Are the LSP lifetime control operations implemented?	7.3.16.4, 7.3.16.3	M	Yes □
CSNP Generation	Is the generation of CSNPs implemented?	7.3.15.3, 7.3.17	M	Yes □
PSNP Generation	Is the generation of PSNPs implemented?	7.3.15.4, 7.3.17	M	Yes □
SNP Processing	Are the sequence number PDU processing procedures implemented?	7.3.15.2, 7.3.17	M	Yes □
LSDB Overload	Are the LSP database overload operations implemented?	7.3.19	M	Yes □

Decision Process: General

Item	Functionality/Description	References	Status	Support N/A
Minimum Cost Path	Is computation of a single minimum cost path based upon each supported metric implemented?	7.2.6	M	Yes □
Equal Cost Paths	Is computation of equal minimum cost paths based upon each supported metric implemented?	7.2.6	O	Yes □ No □
Down Stream Paths	Is computation of downstream routes based upon each supported metric implemented?	7.2.6	O	Yes □ No □
Multiple LSPs Recognition	Are multiple LSPs used only when a LSP with LSP#0 and remaining lifetime greater than 0 is present?	7.2.5	M	Yes □
Overloaded IS Exclusion	Are links to ISs with overloaded LSDBs ignored?	7.2.8.1	M	Yes □
Two Way Connectivity	Are links not reported by both end ISs ignored?	7.2.8.2	M	Yes □
Path Preference	Is the order of preference for path selection implemented?	7.2.8.2	M	Yes □
Excess Path Removal	Is removal of excess paths implemented?	7.2.7	M	Yes □
FIB Construction	Is the construction of ISO8473 Forwarding Information Bases implemented?	7.2.9	M	Yes □

Forward/Receive Process: General

Item	Functionality/Description	References	Status	Support N/A
FIB Selection	Is selection of appropriate Forwarding Information Base implemented?	7.4.2	M	Yes □
NPDU Forwarding	Is forwarding of ISO8473 PDUs implemented?	7.4.3.1, 7.4.3.3	M	Yes □
Receive Process	Are the basic receive process functions implemented?	7.4.4	M	Yes □

A.4.2.1 Protocol Summary: ISO 10589: 19XX Level 1 Specific Functions

Item	Functionality/Description	References	Status	Support N/A
*L1IS	Are Level 1 IS-IS routeing functions implemented?	12.1.3	M	Yes □
Area IS Count	How many ISs can this system support in a single area?	12.2.5	L1IS: M	N =
L1 Manual ES Adjacency	Are the manual ES adjacencies implemented?	7.3.3.1	L1IS: M	Yes □

Level 1 Subnetwork Dependent Functions

Item	Functionality/Description	References	Status	Support N/A
L1 LAN IIH PDU	Are L1 LAN IIH PDUs correctly constructed and parsed?	9.5	C.3: M	□ Yes □

C.3: if L1IS and LAN then M else –

Level 1 Update Process

Item	Functionality/Description	References	Status	Support N/A
L1 LS PDU	Are L1 LS PDUs correctly constructed and parsed?	9.8	L1IS: M	Yes □
L1 CSN PDU	Are L1 CSN PDUs correctly constructed and parsed?	9.10	L1IS: M	Yes □
L1 PSN PDU	Are L1 PSN PDUs correctly constructed and parsed?	9.12	L1IS: M	Yes □

Level 1 Decision Process

Item	Functionality/Description	References	Status	Support N/A
L1 Nearest L2 IS Identification	Is the identification of the nearest L2 IS implemented?	7.2.9.1	L1IS: M	Yes □
L1 Area Addresses Computation	Is the computation of area addresses implemented?	7.2.11	L1IS: M	Yes □

A.4.2.2 Protocol Summary: ISO 10589: 19XX Level 2 Specific Functions

Item	Functionality/Description	References	Status	Support N/A
*L2IS	Are Level 2 IS-IS routeing functions implemented?	12.1.4	O	Yes □ No □
IS Count	What is the total number of ISs that this L2 IS can support?	12.2.5	C.4: M	□ N =
L2IS Count	How many level 2 ISs does this implementation support?	12.2.5.1	L2IS: M	□ N =
*RA Prefix	Are Reachable Address Prefixes supported on circuits?	8.1, 7.3.3.2	L2IS: O	□ Yes □ No □
External Metrics	Are external metrics supported?	7.2.2, 7.2.12, 7.3.3.2	RA Prefix: M	□ Yes □
*Partition	Is level 1 partition repair implemented?	7.2.10	L2IS: O	□ Yes □ No □

C.4:

Level 2 Subnetwork Dependent Functions

Item	Functionality/Description	References	Status	Support N/A
L2 LAN IIH PDU	Are L2 LAN IIH PDUs correctly constructed and parsed?	9.6	C.5: M	□ Yes □
*8208 DA	Are ISO8208 Dynamic Assignment circuits implemented?	8.3	O.1	Yes □ No □
RA Adjacency Management	Are the reachable address adjacency management operations implemented?	8.3.2.2-8.3.5.6	8208 DA: M	□ Yes □
Call Establishment Metric Increment	Are non-zero values of the callEstablishmentMetricIncrement supported?	8.3.5	8208 DA: O	□ Yes □ No □
Reverse Path Cache	Is 8208 reverse path cache implemented?	8.3.3	8208 DA: O	□ Yes □ No □

C.5: if L1IS and LAN then M else –

Level 2 Update Process

Item	Functionality/Description	References	Status	Support N/A
L2 LS PDU	Are L2 LS PDUs correctly constructed and parsed?	9.9	L2: M	□ Yes □
L2 CSN PDU	Are L2 CSN PDUs correctly constructed and parsed?	9.11	L2: M	□ Yes □
L2 PSN PDU	Are L2 PSN PDUs correctly constructed and parsed?	9.13	L2: M	□ Yes □

Level 2 Decision Process

Item	Functionality/Description	References	Status	Support N/A
L2 Attached Flag	Is the setting of the attached flag implemented?	7.2.9.2	L2IS: M	□ Yes □
L2 Partition DIS election	Is the election of partition L2 DIS implemented?	7.2.10.2	Partition: M	□ Yes □
L2 Partition Area Addresses Computation	Is the computation of L1 partition area addresses implemented?	7.2.10.3	Partition: M	□ Yes □
L2 DIS Partition Repair	Is partition detection and repair via virtual L1 links implemented?	7.2.10.1	Partition: M	□ Yes □

Level 2 Forward/Receive Process

Item	Functionality/Description	References	Status	Support N/A
L2 NPDU Encapsulation	Is the encapsulation of NPDUs implemented?	7.2.10.4, 7.4.3.2	Partition: M	□ Yes □
L2 NPDU Decapsulation	Is the decapsulation of NPDUs implemented?	7.4.4	Partition: M	□ Yes □