INTERNATIONAL STANDARD

ISO/IEC 14765

First edition
1997-06-15

Information technology — Framework for protocol identification and encapsulation

Technologies de l'information — Cadre général pour identification et encapsulage de protocole

Reference number
ISO/IEC 14765:1997(E)

Contents

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work.

In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

International Standard ISO/IEC 14765 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 6, *Telecommunications and information exchange between systems*, in collaboration with ITU-T. The identical text is published as ITU-T Recommendation X.260.

Annexes A and B of this International Standard are for information only.

This page intentionally left blank

INTERNATIONAL STANDARD

ITU-T RECOMMENDATION

INFORMATION TECHNOLOGY – FRAMEWORK FOR PROTOCOL IDENTIFICATION AND ENCAPSULATION

1 Scope

In a layered approach to protocol architecture, protocols have a relationship to one another such that a protocol at layer (n) uses the services of the layer below it – the (n – 1) services – which, in turn, are provided by a layer (n – 1) protocol. One of the services used by a layer (n) protocol is the *encapsulation* of its (n) Protocol Data Units (PDUs) in a way which is transparent to it. Such encapsulation is realized by the carriage of the (n) PDUs as user data in an (n – 1) Service Data Unit (SDU).

In a limited case, the operation of a particular protocol at layer (n – 1) implies the operation, above layer (n – 1), of a single layer (n) protocol or single set of related (n) / (n + 1)... protocols. However, in a more general case, there may be more than one protocol (or set of related protocols starting) at layer (n) that can operate above layer (n – 1) in a given environment. In such cases, there is a need for explicit *identification* of the protocol (or set of protocols starting) at layer (n).

There also may be a need to manipulate the (n – 1) protocol (i.e. the *encapsulating* protocol) in certain ways specific to the layer (n) protocol (i.e. the *encapsulated* protocol). Such manipulations form the basis of a set of procedures that must be specified for the layer (n) protocol.

The above observations regarding protocol identification and encapsulation are also applicable in cases where an (n) layer is further divided into sublayers.

Cases in which an (n) protocol operates for the purpose of establishing a parallel universe of protocols (regardless of the layered structure of that universe) also give rise to a need for the (n) protocol to be able to identify the protocol(s) in the parallel universe. In these cases, however, there is no encapsulating/encapsulated relationship between the (n) protocol and the parallel universe set of protocols.

The above principles lead to a need to establish a framework for protocol identification and encapsulation. These principles apply to the relationship between two protocols (recognizing that one of them may be a set of related protocols) and can be applied recursively. This Recommendation | International Standard provides a framework for explicit protocol identification and for protocol encapsulation. Implicit protocol identification (see 4.2) is beyond the scope of this Recommendation | International Standard.

2 Normative references

The following Recommendations and International Standards contain provisions which, through reference in this text, constitute provisions of this Recommendation | International Standard. At the time of publication, the editions indicated were valid. All Recommendations and International Standards are subject to revision, and parties to agreements based on this Recommendation | International Standard are encouraged to investigate the possibility of applying the most recent edition of the Recommendations and International Standards listed below. Members of IEC and ISO maintain registers of currently valid International Standards. The Telecommunication Standardization Bureau of the ITU maintains a list of currently valid ITU-T Recommendations.

2.1 Identical Recommendations | International Standards

- ITU-T Recommendation X.200 (1994) | ISO/IEC 7498-1:1994, *Information technology – Open Systems Interconnection – Basic Reference Model: The Basic Model.*
- ITU-T Recommendation X.263 (1995) | ISO/IEC TR 9577:1996, *Information technology – Protocol identification in the network layer*.

2.2 Additional references

- ITU-T Recommendation X.37 (1995), *Encapsulation in X.25 packets of various protocols including frame relay*.
- ISO/IEC 13515[1], *Information technology – Telecommunications and information exchange between systems – Generic Multiprotocol Encapsulation (GME): Application to frame relay and ATM*.

3 Abbreviations

EdP	Encapsulated Protocol
EF	Encapsulation Function
EgP	Encapsulating Protocol
EPIF	Encapsulated Protocol Information Field
IdP	Identified Protocol(s)
IgP	Identifying Protocol
IPI	Initial Protocol Identifier
PCI	Protocol Control Information
PDU	Protocol Data Unit
PEM	Protocol Encapsulation Method
PId	Protocol Identification
PIE	Protocol Identification and Encapsulation
PIM	Protocol Identification Method
SDU	Service Data Unit
SPI	Subsequent Protocol Identifier

4 Definitions and concepts

4.1 Basic Reference Model concepts

The following concepts from ITU-T Rec. X.200 | ISO/IEC 7498-1 are used here:

a) concatenation;
b) layer;
c) protocol;
d) Protocol Control Information (PCI);
e) Protocol Data Unit (PDU);
f) protocol identification;
g) protocol identifier;
h) segmentation/reassembly;
i) Service Data Unit (SDU);
j) sublayer.

4.2 Additional definitions and concepts

The definitions and concepts below apply to this Recommendation | International Standard.

4.2.1 explicit protocol identification method: An explicit PIM is one in which Protocol Control Information (PCI) is used to identify a protocol, a set of related protocols, or a family of protocols.

---

[1] Presently at the stage of draft.

4.2.2 implicit protocol identification method: An implicit PIM is one in which there is no PCI used to identify a protocol. Identification occurs through mechanisms such as coupling in a Recommendation or International Standard of an IgP with an IdP [e.g. stating that a specific (n) protocol is used above an (n – 1) protocol]; association of a physical port of a system with one or a set of related protocols; or association at provisioning of a "permanent" connection.

4.2.3 set of alternative protocols: Given protocols $prot_1$, $prot_2$, etc., then $prot_1$, $prot_2$, etc. all operate at the same layer or sublayer.

4.2.4 set of related protocols: Given protocols $prot_1$, $prot_2$, etc., then $prot_1$ operates at layer (n), $prot_2$ operates at layer (n + 1), etc. (where the layers may also be hierarchical sublayers).

4.2.5 family of protocols: Given a set of alternative protocols $prot_1$, $prot_2$, etc., a single identifier is used to identify the set of alternative protocols as a whole, thereby requiring additional methods to identify one member of the family.

5 Overview

5.1 General

As discussed above, there may exist a relationship between protocols that gives rise to a need for one protocol – the Identifying Protocol (IgP) – to identify one of a set of alternative protocols, a set of related protocols, or a family of protocols – the Identified Protocol(s) (IdP). As a result of the identification process, a second relationship may be created between an Encapsulating Protocol (EgP) and an Encapsulated Protocol(s) (EdP). In some cases, the IgP and the EgP may be the same protocol. It is usually the case that an IdP and an EdP are the same.

To provide a basis for developing the necessary relationships among specific protocols, a framework is developed here to set out the principles of Protocol Identification and Encapsulation (PIE). These principles recognize the following aspects of PIE:

a) development of Protocol Identification Methods (PIMs) for identifying an IdP (e.g. the location in the IgP – such as in a particular field: header, trailer, etc. – used to identify the IdP);

b) for each PIM, registration of values of IdPs;

c) requirement on an IgP to specify the PIM it uses to identify IdPs and any further IgP-specific procedures involving the PIM;

d) development of Protocol Encapsulation Methods (PEMs) for use by EgPs; and

e) specification of operations (e.g. limitations, specific manipulations, etc.) of an EgP for a specific EdP.

The above aspects are depicted in Figure 1.

Annex A presents the current status of Recommendations and International Standards in alignment with the framework depicted in Figure 1.

5.2 Interworking and encapsulation

Interworking and *encapsulation* of protocols are two closely related concepts. For the purposes of this Recommendation | International Standard, the following distinctions are made.

Interworking occurs between two or more protocols at the same layer (or sublayer). It is concerned only with the semantic aspects of the (n) layer protocols. In particular, interworking is concerned with the transformation between the semantics of an (n) layer protocol used on one interface and the semantics of other (n) layer protocols used on the other interfaces. The protocols used on the different interfaces may or may not be the same. The transformation between protocols may result in the preservation of the semantic content of all protocols on an end-to-end basis. The transformation only applies to the set of abstract capabilities (or *service)* which the protocols have in common. On the other hand, the transformation may result in a loss of semantic content when crossing interfaces.

Encapsulation (or *tunneling* as it is sometimes called) occurs when a given protocol's PDU (or set of PDUs if the protocol provides segmentation/reassembly capabilities) is used to carry the PDUs of another protocol [that is, the user data parameter of an (n – 1) SDU is used to carry the (n) PDU(s)]. In the general case, no other relationships, such as a strict layering relationship, need exist between the two protocols (e.g. allowing for sublayering or for a given protocol to be encapsulated by protocols with different layer classifications). Encapsulation completely preserves the semantics of the EdP.

From the perspective of this Recommendation | International Standard, *port access*, as a method for interworking as defined in Recommendation X.300, is viewed as a method of encapsulation.

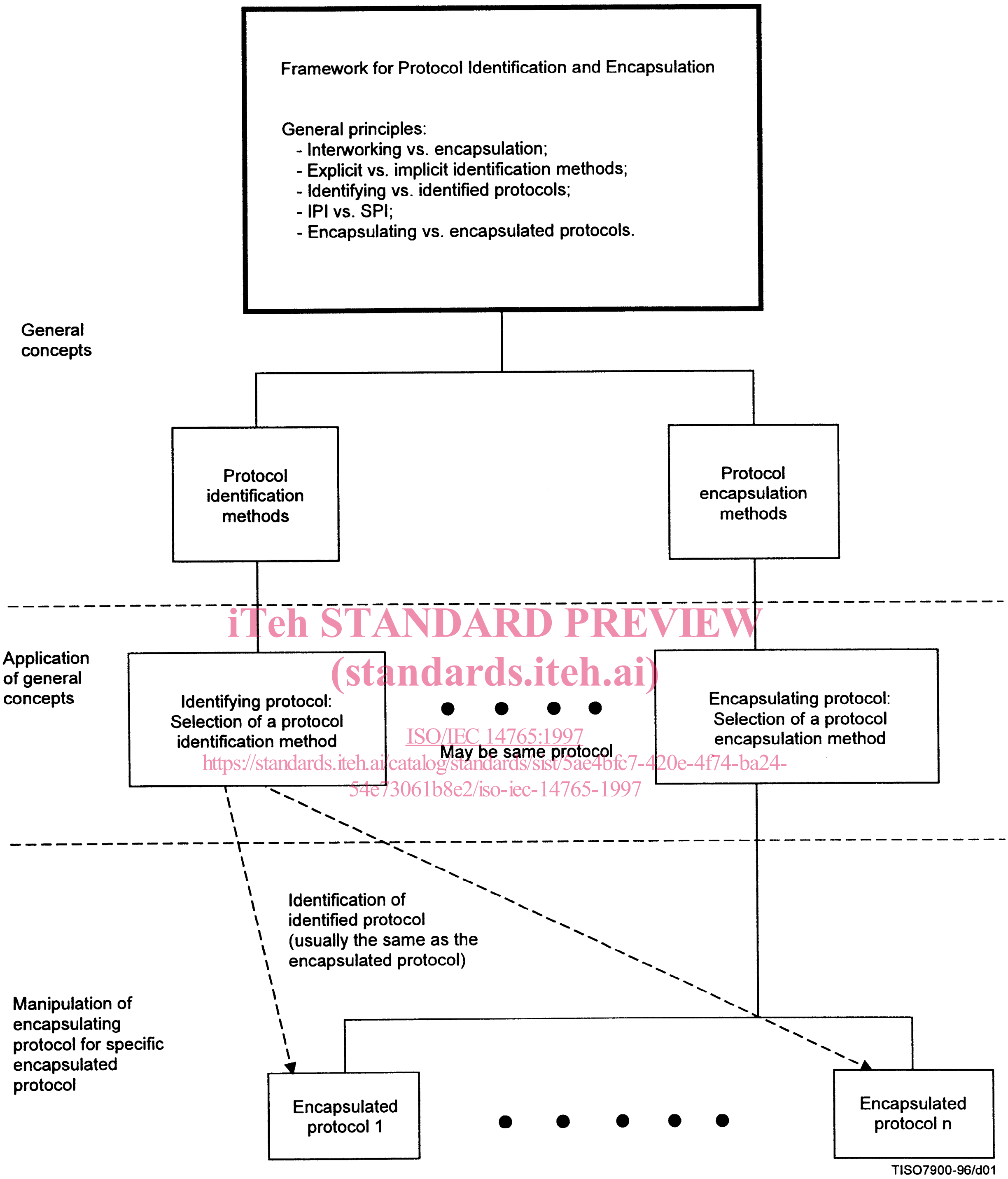


Figure 1 – Framework for protocol identification and encapsulation principles

6 Principles of protocol identification

6.1 Need for protocol identification

The need for PId arises, for the general case, when there is more than one IdP (or set of related IdPs) that can be used in a specific environment (e.g. layer or parallel universe). In such cases, identification of the IdP (or set of related IdPs) is necessary to allow for meaningful communication. The process of PId needs to be performed for a specific instance of communication. Such instances can be:

a) for the lifetime of a connection of the IgP, so that identification or negotiation/selection of one of a number of alternative IdPs (or alternative sets of related IdPs) is required to be done during the IgP's connection establishment phase;

b) for the transmission of a single SDU (in the case of a connection-mode IgP, the selection of allowing multiple concurrent IdPs would have been identified during the IgP's connection establishment phase).

For cases where a multiplicity of alternative IdPs is selected for use in an instance of communication of the IgP, the IdPs may operate concurrently or sequentially with respect to the IgP. The use of multiple IdPs may require agreement of the identities of the specific set of alternative protocols to be used for the instance of communication.

It is also possible that a set of alternative protocols can be identified as a single family, in which case further identification methods are needed to identify a specific member of a family in an instance of communication.

As a result of the need developed above for protocol identification, the following are necessary:

a) registries of values to identify protocols;

b) Protocol Identification Methods (PIMs) to provide a basis for negotiating/selecting one or more IdPs;

c) explicit PCI in the IgP to identify the specific IdP (or family or set of related protocols).

These elements are discussed below.

6.2 Protocol identifier registries and values

A register of values (which itself can be a Recommendation or International Standard or part thereof) is used to record how a protocol, when used as an IdP, is identified. Such a register should be easily modifiable and authority for such modifications shall be identified.

It is permissible for an IdP to appear in more than one register, with the same or a different value.

6.3 Protocol identification methods

A PIM is used to identify a specific IdP (or family or set of related protocols) for use in a specific instance of communication. The PIM can be either implicit or explicit (see 4.2). Implicit PIMs are beyond the scope of this Recommendation | International Standard.

Associated with an explicit PIM is a register of allowed protocol identifier values (see 6.2). It is possible for the same register to be associated with many PIMs rather than developing a new register for use with different PIMs.

An explicit PIM requires the use of PCI to identify protocols. There can be many PIMs, although a particular IgP may support only a few (usually one). An IgP shall specify the PIM it uses. Such specification shall also include the location and number of octets of the PCI used for the PIM.

When an IgP supports several PIMs, it may be desirable to identify an IdP using the PIM that results in the least number of octets. In any case, the particular PIM used to identify an IdP should be specified to ensure interworking.

A PIM may allow for negotiation/selection of IdPs for a specific instance of communication as follows:

a) only one IdP to be selected (for use with a connection of the IgP, where the IdP is identified by the PIM during the connection request phase of the IgP or just identification of the specific IdP during the data transfer phase of the IgP);

b) only one IdP to be selected for use with a connection of the IgP but where negotiation of the specific IdP (from a set of alternative IdPs) takes place during the connection establishment phase of the IgP;

c) a multiplicity of alternative IdPs to be selected (perhaps requiring negotiation of the specific set of alternative IdPs by the PIM during the connection establishment phase of the IgP or just identification of the specific IdP during the data transfer phase of the IgP).

In cases where usage of a multiplicity of alternative IdPs has been agreed for an instance of communication, a PIM may also provide for the specification of whether, during the transfer of data, only one, or, alternatively, more than one IdP is to be used in conjunction with a single SDU. That is, the PIM may also specify aspects of encapsulation (see clause 7).

In cases a) and b) above, there is no need to further identify the IdP during the data transfer phase of the IgP. In case c), further identification of the IdP(s) is required in the IgP's PDUs or IdP's SDUs.

6.4 Protocol identifiers

Protocol identifiers, when explicit, occur in PCI and are based on values maintained in a register (see 6.2).

The circumstance in which an IdP is used determines whether there is a need for it to identify itself. Such need arises when:

a) the IdP is one of a specific family of protocols which has been identified;

b) the IdP has not been identified by an IgP and alternative IdPs exist that may be used.

The second case may be regarded as a general or null family. Nevertheless, the IdP must identify itself in both cases. For both of these cases, the location of the protocol identifier must be specified. Such an identifier is known as an *Initial Protocol Identifier* (IPI). Typically, the IPI will be at the beginning of the (n – 1) layer's SDU; in this case, the IdPs may be regarded as *header-based protocols*. However, the identifier may also be at the end of the SDU in the case of *trailer-based protocols*. When both header- and trailer-based protocols use the same (n – 1) protocol, identifying mechanisms are needed in the (n – 1) protocol to properly differentiate between the header- and trailer-based protocols in the (n) layer.

An IdP need not (but, nevertheless, still may) identify itself when its usage has been unambiguously identified by an IgP.

An IdP may also be an IgP. In such cases, the identifier used for identifying subsequent protocols is known as a *Subsequent Protocol Identifier* (SPI). As stated in 6.3, the PIM used by the SPI (including the location of the SPI in the IgP's PCI) shall be specified. It may be the case that what an IgP views as an SPI may be an IPI from the perspective of the IdP.

A relationship between IPI and SPI is depicted in Figure 2.

It is possible for a subsequent protocol, in turn, to identify further protocols within a layer (i.e. to have a nesting of protocols). It is also possible, in some limited cases, for there to be multiple "initial" protocols. For example, when a data compression protocol is used as the initial protocol, the compressed protocol itself is identified by an IPI.

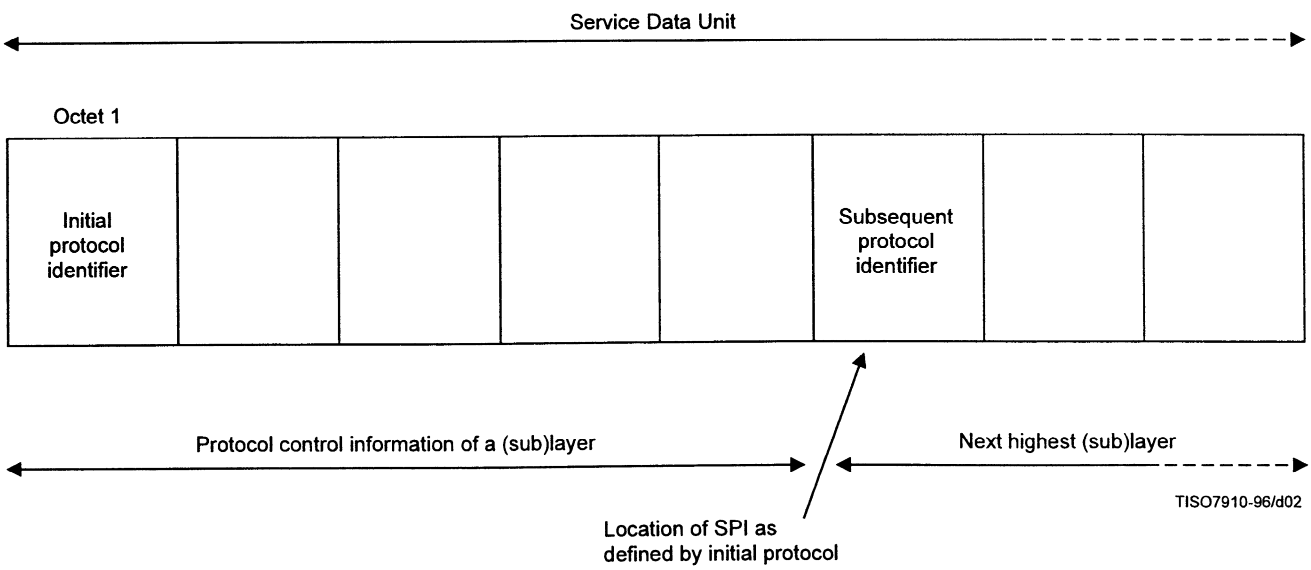


Figure 2 – Relationship of IPI and SPI

7 Principles of protocol encapsulation

Protocol encapsulation implies a relationship between two protocols – an EgP and an EdP. This relationship involves the following dimensions:

a) manipulations of an EdP;

b) manipulations/limitations of a specific EgP for a specific EdP;

c) identification of the EdP, as needed, as it may be encapsulated in an EgP;

d) encapsulation of one or more of the EdP's PDUs, including their delimitation, in the EgP.

These dimensions are embodied in an Encapsulation Function (EF).

The operation of an EF involves two elements:

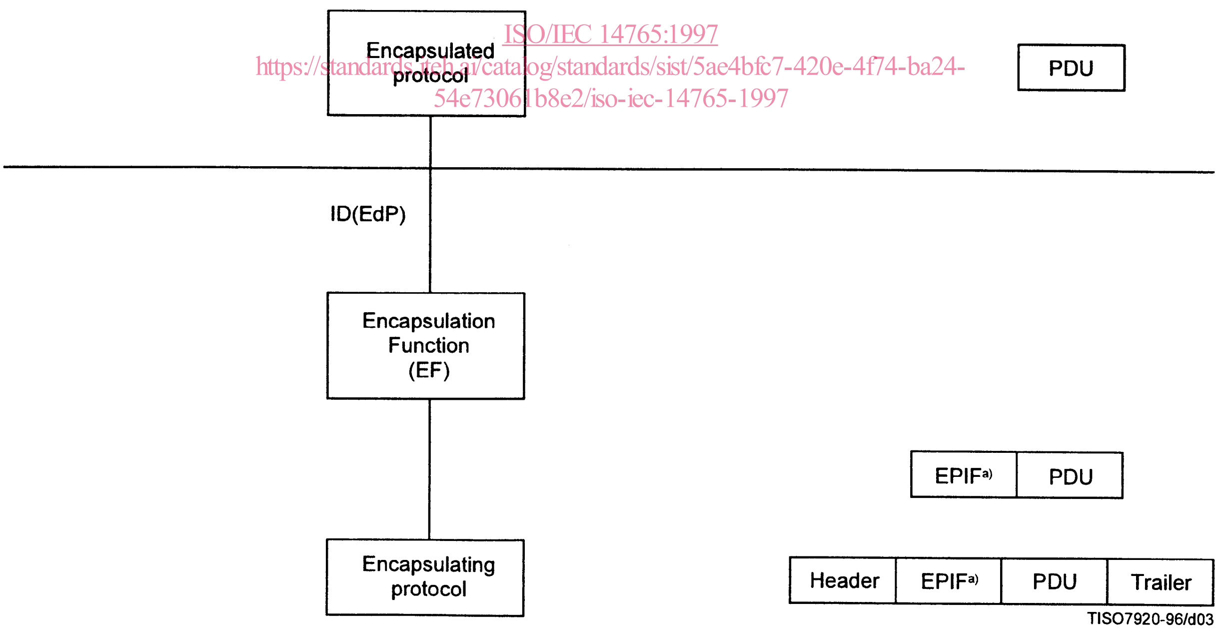
a) the static definition of the above dimensions of the EF;

b) the dynamic operation of the EF in conjunction with zero or more other EFs, in the context of their respective static definitions, to encapsulate the PDU(s) of EdP(s), as provided in the user data parameter in primitives of the service supported by the EgP, during a particular instance of communication.

It is beyond the scope of this framework to specify any limitations during instances of communication on how many EdPs may be encapsulated in an EgP or how EFs with similar characteristics in one or more of the above dimensions may be combined.

7.1 Encapsulation function

An EF performs encapsulation as discussed above in clause 7. The EF resides in the same system as the EgP and EdP. The generic operation of the EF is depicted in Figure 3.



a) EPIF: Encapsulated Protocol Information Field, when present, may contain:
– EdP identification; and/or
– EdP-PDU delimiting information (e.g. length information).

Figure 3 – Generic operation of an encapsulation function