ISO 4396-9:2022(1)

ISO/IEC PRF 4396-9

ISO/IEC JTC 1/SC 6/WG 7

Date: 2022-09

Secretariat: KATS

Date: 2023-10-13

Telecommunications and information exchange between systems — Future network — Recursive inter-network architecture — —

Part 9: Error and flow control protocol

Télécommunications et échange d'information entre systèmes — Architecture récursive inter-réseaux —

Partie 9: Protocole de contrôle d'erreurs et de flux

FDIS stage

Edited DIS - MUST BE USED

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.

The procedures used to develop this document and those intended for its further maintenance are described in the ISO/IEC Directives, Part 1. In particular, the different approval criteria needed for the different types of document should be noted. This document was drafted in accordance with the editorial rules of the ISO/IEC Directives, Part 2 (see www.iso.org/directiveswww.iso.org/directives or www.iec.ch/members_experts/refdocs).

Field Code Changed

Attention is drawnISO and IEC draw attention to the possibility that some of the elementsimplementation of this document may beinvolve the subjectuse of (a) patent(s). ISO and IEC take no position concerning the evidence, validity or applicability of any claimed patent rights. in respect thereof. As of the date of publication of this document, ISO and IEC had received notice of (a) patent(s) which may be required to implement this document. However, implementers are cautioned that this may not represent the latest information, which may be obtained from the patent database available at www.iso.org/patents and https://patents.iec.ch. ISO and IEC shall not be held responsible for identifying any or all such patent rights. Details of any patent rights identified during the development of the document will be in the Introduction and/or on the ISO list of patent declarations received (see www.iso.org/patents) or the IEC list of patent declarations received (see https://patents.iec.ch).

Any trade name used in this document is information given for the convenience of users and does not constitute an endorsement.

For an explanation of the voluntary nature of standards, the meaning of ISO specific terms and expressions related to conformity assessment, as well as information about ISO's adherence to the World Trade Organization (WTO) principles in the Technical Barriers to Trade (TBT) see www.iso.org/iso/foreword.htmlwww.iso.org/iso/foreword.html. In the IEC, see www.iec.ch/understanding-standardswww.iec.ch/understanding-standards.

This document was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 6 *Telecommunications and information exchange between systems*.

A list of all parts in the ISO/IEC 4396 series can be found on the ISO and IEC websites.

Any feedback or questions on this document should be directed to the user's national standards body. A complete listing of these bodies can be found at www.iso.org/members.html and www.iec.ch/national-committeeswww.iso.org/members.html and www.iec.ch/national-committees.

Introduction

This document describes the Error and Flow Control Protocol (EFCP) specification. EFCP is the data transfer protocol of the Recursive InterNetwork Architecture (RINA).[1],[2],[3]).[1],[2],[3] It supports a permanent connection with all types of data transfer services.

RINA is a new network architecture based on the idea that networking is inter-process communication (IPC) and only IPC. RINA imposes the strict separation of mechanisms and policies as one of the main architectural features. EFCP assumes that other RINA components are in place and supplies other mechanisms such as addressing and flow allocation, which are outside of its scope.

EFCP[4][4] is based on the concept of timer-based reliable management of connections.[5],[.[5],[6] In this way, EFCP operates with a minimum exchange of packets to manage connections and to keep protocol machines involved in a connection synchronised. EFCP uses direct control messages to preserve data from being lost, mis-sequenced or duplicated.

The International Organization for Standardization (ISO)

Telecommunications and/or International Electrotechnical Commission (IEC) draw attention to the fact that it is claimed that compliance with this document may involve the use of a patent.

ISO and/or IEC take no position concerning the evidence, validity and scope of this patent right.

The holder of this patent right has assured ISO and/or IEC that he/she is willing to negotiate licences under reasonable and non-discriminatory terms and conditions with applicants throughout the world. In this respect, the statement of the holder of this patent right is registered with ISO and/or IEC. Information may be obtained from the patent database available at www.iso.org/patents or patents.iec.ch.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights other than those in the patent database. ISO and/or IEC shall not be held responsible for identifying any or all such patent rights.

The following patents may apply:

10,135,689 Position parameterized recursive network architecture with topological addressing

9,762,531 Position parameterized recursive network architecture with topological addressing

9,584,633 Method and system for managing network communications

9,288,176 Position parameterized recursive network architecture with topological addressing

8,769,077 Parameterized recursive network architecture with topological addressing

8,352,587 Parameterized recursive network architecture with topological addressing

8,103,797 Parameterized recursive network architecture with topological addressing

Future network information exchange between systems — Recursive inter-network architecture —

Part 9: Error and flow control protocol

1 Scope

This document provides the Error and Flow Control Protocol (EFCP) specification. EFCP provides an inter-process communication (IPC) service to an application process, which can be a (N+1)-IPC process (IPCP), with the requested Quality of Service (QoS). One or more service data units (SDUs) are passed on the (N)-port-id to the (N)-DIF (distributed IPC facility) to be sent to the destination application process. Protocol data unitesunits (PDUs) transferred by the (N)-DIF are delivered to the (N)-port-id for the using Application Process. This document describes the placement of EFCP within RINA, the components EFCP consists of, and the mechanisms and policies that are involved in EFCP's work, and the timers and control mechanisms required to manage the connection.

EFCP comprises two logical components, the data transfer procedures (DTP), providing tightly bound mechanisms and the data transfer control procedures (DTCP), which provides loosely bound mechanisms.

ItThis document provides:

- the service definition;
- an overview of EFCP;
- a description of the placement of EFCP within recursive internetwork architecture (RINA);
- the common elements of data transfer protocol (DTP) and data transfer control protocol (DTCP);
- DTP structure and functions;
- DTCP structure and functions;
- an informative list of all policies in EFCP.en

2 Normative references

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

ISO/IEC 4396-1, *Telecommunications and information exchange between systems – Future Network - — Recursive Inter-Network Architecture– — Part 1: RINA Reference Model*

ISO/IEC 4396-7, *Telecommunications and information exchange between systems – Future Network - — Recursive Inter-Network Architecture– — Part 7: RINA Flow Allocator*

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO/IEC 4396-1, ISO/IEC 4396-7 and the following apply.

ISO and IEC maintain terminology databases for use in standardization at the following addresses:

— ~~—~~ISO Online browsing platform: available at ~~https://www.iso.org/obp~~https://www.iso.org/obp

— ~~—~~IEC Electropedia: available at ~~https://www.electropedia.org/~~https://www.electropedia.org/

3.1
data transfer state vector
DT-SV
vector that provides shared state information for the connection maintained by the data transfer protocol (DTP) and the data transfer control protocol (DTCP)

3.2
maximum PDU lifetime
MPL
maximum time a protocol data unit (PDU) sent by a member of a distributed inter process communication (IPC) facility (DIF) may exist in the DIF

3.3
data transfer application entity
DTAE
error and flow control protocol (EFCP)-protocol machine (PM) task managing the shared state with its peer

4 Overview of the EFCP

EFCP assumes the fundamental idea that reliability can be achieved only through the bounding of three times. These times are the maximum PDU lifetime (MPL), the maximum time for the delay in acknowledging a PDU (A-time), and the maximum time to exhaust retransmitting a PDU (R-time). If these times are bounded by the quantities MPL A (maximum delay to send an acknowledgement) and R (maximum time before giving up retransmitting a PDU), then reliability can be guaranteed. Using these bounds, two important times are set up to reliably manage the data transfer, the sender and receiver inactivity times. The time Δt is calculated ~~through~~as the ~~following quantity:~~sum of Maximum PDU Lifetime, MPL; Maximum Time to before Sending an Ack, A; and the Maximum Time to Exhaust Retries, R.

~~$\Delta t = MPL + A + R$~~

The upper bound for the receiver's inactivity timer is 2 ~~Δt~~ Δt and 3 ~~Δt~~ Δt for the sender's inactivity time.

Every flow instantiated within a DIF triggers the creation of an instance of EFCP and its associated state vector~~.[3]~~.[3] For security purposes, all EFCP connections have flow control, although retransmission control may not be required. The EFCP is able to provide other services, such as reliable transmission with QoS.

DTP uses a single PDU that carries addresses, a connection-id, a sequence number, and various flags to signal congestion, and other conditions that are relevant to the operation of the protocol.

While there are four PDU types defined for DTCP, these are supported by only three PDU formats: Selective Ack/Nack/Flow, Ack/Flow, and Rendezvous/Flow. Each of these ~~carry~~carries addresses, a connection-id, a sequence number, and retransmission control and/or flow control information, etc. The opcodes indicate which fields in the PDU are valid. Not only is Ack and Flow Control information sent to update the sender, but the receiver's left window edge is sent with Credit as a check on the state. All connections are flow controlled to provide a countermeasure against a denial-of-service attack. While

DTCP controls the flow of DTP PDUs, DTCP never has cause to inspect their contents. In essence, DTP writes to the state vector and DTCP reads the DTP state from it.

EFCP is based on concepts first described by Watson in,[5],[5] and adapted as part of the Recursive Inter-Network Architecture (RINA) as the mechanism to send and receive variable-length blocks of information.

EFCP separates mechanism and policy. There is a default action associated with each policy. The policy can take additional action beyond the default or replace the default action. A policy may only reference the state variables defined by this document and the parameters specified by the management objects to instrument EFCP. Policies may define both local and persistent variables. Local variables are assumed to be new instantiations on each call, while persistent variables retain their value between calls. The latter allows a policy to maintain state, if necessary. An optional parameter list may be present. The parameters used should be variables internal to the policy. The informative Annex AAnnex A provides a summary of the EFCP policies.

5 Role of EFCP within the IPC process

5.1 General

The EFCP is responsible for carrying out the data transfer and data transfer control tasks. In order to do so, each EFCP instance exchanges protocol data units (PDUs) with an EFCP instance in another inter process communication (IPC) process (IPCP) (although this other instance couldcan also be located in the same IPCP in a degenerate case).

The IPCP provides IPC services to one or more application processes (which in turn may be a higher-layer IPCP). Each instantiation of an IPC data-transfer-service is called a flow and is identified by a port-id. Application processes can write or read service data units (SDUs) from the port, thus communicating with one or more remote application process instances. When an application process writes an SDU to a port, first the delimiting module associated to the port may fragment the SDU or concatenate it with other SDUs (or SDU fragments), producing one or more user data fields (UDFs). UDFs are delivered to the EFCP instance associated to the port, which will add a header to the UDF [the protocol control information (PCI)], producing a PDU that is delivered to multiplexing task (MT). The MT multiplexes PDUs from different EFCP instances into one or more N-1 ports provided by one or more underlying IPCPs.

In the “read path”, the MT will deliver to the EFCP module the PDUs addressed to the IPCP. EFCP then makes available each PDU to the relevant EFCP instance, which will process the PCI, remove it and deliver the UDF to the delimiting module to obtain the original SDUs. SDUs will be buffered until they are read by the application process owning the port associated to the EFCP instance.

The lifecycle of EFCP instances is managed by a layer management component of the IPCP called flow allocator (FA). When an application process requests the creation of a flow to the IPCP, the FA is the component that deals with the request. Part of the responsibilities of the FA are to create the EFCP instance (with the appropriate mix of policies) and bind it to the port providing the flow.

The FA monitors the EFCP connection and, at some point, it can decide that it is time to create a new EFCP instance, bind it to the flow and remove the old one (for example, to prevent sequence number rollover to happen). This capability is enabled by the fact that ports and EFCP instances are decoupled: only ports are exposed to application processes outside of the layer. At any moment in time there can only be one active EFCP instance bound to a specific port.

When the application process using the flow [or the operating system (OS) in its behalf, for example if the application process has crashed] decides that it no longer needs it, it invokes the deallocate flow primitive. This operation causes the FA to destroy the EFCP instance and release the port that was associated to the flow.

5.2 Description of the Data Transfer Protocol (DTP) Task

A DTP Task is always created when a flow is allocated. DTP performs all mechanisms associated with the Data Transfer PDU, e.g. sequencing, invokes SDU Protections and Delimiting to do fragmentation/reassembly.

While the DTCP-State can be discarded during long periods of no traffic, the DTP-State Vector cannot. The DTP-State is only discarded after an explicit release by the AP or by the System (if the AP crashes), i.e. the port-ids are released. The DIF mustshall have procedures to ensure that when traffic resumes the CEP-id is associated with the correct port-id.

The indirection introduced by the distinction between the port-id and CEP-id avoids the requirement of a separate connection for integrity mechanisms (encryption). To prevent replay, insertion, or deletion, a sequence number is introduced but it can't be allowed to wrap. When using encryption in a DIF, as a flow sequence number nears its limit, a new connection is allocated with different CEP-ids and bound to the port-id. This will be monitored by the flow allocator instance (FAI) that created the flow. The FAI will create the new connections and create the bindings with the port-id. There will be a period when PDUs are accepted from both connections. After 2*Δt, the earlier connection will be allowed to expire. For flows without encryption or flows that do not send sufficient data for sequence numbers to roll over, the port-id and the CEP-id may be the same.

Depending on the policies in force on the flow, the Delimiting will provide fragmentation/reassembly functions (breaking SDUs into more than one PDU) and concatenation/separation functions (combining multiple SDUs into one PDU). DTP creates PDUs and processes them upon arrival and imposes sequencing. DTCP provides either retransmission or flow control or both. If there is flow control and the flow control window is closed, then PDUs are put on the closed window queue. There is a policy to monitor the length of this queue, and sending PDUs comes under the control of the DTCP. A Transmission Control Policy may override and queue PDUs to slow the data rate for other reasons. If flow control is not present or the window is open, then the PDUs are posted to the Multiplexing Task. If retransmission control is being used, a copy of the PDU is placed on the retransmission queue and a timer started.

When PDUs arrive for DTP, SDU Protection determines if they can be processed, and if so, they are ordered. If Retransmission Control is in use, i.e. DTCP is present and a Data Transfer PDU is received, if the Sequence Number of the PDU is less or equal than the last sequence number acknowledged, then this PDU is a duplicate and is discarded. Otherwise, the PDU is put on the PDU Reassembly Queue and Delimiting is invoked to create SDUs and deliver them to the using application process.

Since EFCP may allow gaps in the data stream, the concept of the left window edge exists regardless of whether DTCP is present. With DTCP present, it is assumed that all PDUs with a sequence number smaller than or equal to the value of the receiver left window edge (RcvLeftWindowEdge) have been acknowledged. This means that there will be no retransmissions of PDUs with sequence numbers less than or equal to the RcvLeftWindowEdge. All gaps have been resolved one way or another. When DTCP is not present, this implies that all PDUs with sequence numbers smaller than or equal to the value of the RcvLeftWindowEdge have been processed for delivery to the process above.

If out of order delivery is allowed, then it mustshall be in terms of SDUs. As indicated by the previous discussion of the use of Left Window Edge in DTP, this makes the most sense when there is one SDU per PDU. If not, the PDUs have to be effectively ordered to reassemble them. Randomly choosing initial sequence numbers for sequences of PDUs that constitute an SDU is not recommended since this can adversely affect the transmission rate.

This has implications for whether or not order is imposed on the PDUs. Regardless of whether there is an ordering, a PDU identifier of some sort is required. Most implementations assign these sequentially. Random assignment is not recommended for the following reasons:

a) a) much less of the sequence number space can be used within an MPL;

b) b) the assumption mustshall be made that an SDU is no larger than the Maximum PDU size.

If b) does not hold, then SDUs mustshall be reassembled and the order of the PDUs mustshall be known at least within a sequence. If sequence numbers are assigned randomly and the number of PDUs sent within an MPL is a sizable fraction total sequence number space of PDUs, the chances of the same sequence number occurring with an MPL is fairly high. It can be necessary to limit transmission to 25 % or even less of the sequence number space, whereas with sequential assignment the full sequence number space can be used.

Transmission Control provides the means to control the sending or not sending of PDUs beyond what is indicated by the feedback from the apposite DTCP instance. For example, there can be conditions under which the destination indicates PDUs may be sent, but other conditions, e.g. congestion, that indicate that PDUs should not be sent or fewer sent or sent at a slower rate. Transmission Control also responds to the detection of Lost Control PDUs.

5.3 DTP user service API definition

5.3.1 Read API or read immediate invoked

Parameters: Port-id, pointer to a buffer, array of buffers, or list of buffers, a length in bytes.

This event occurs when the user of the flow does a Read or Read Immediate. Several options affect the Read, such as Partial Delivery and MaxSDUSize. A Read will read available SDUs, but not those with the A-timer running on them, i.e. there are gaps that may be filled. Read Immediate overrides thatthe constraints and reads as many SDUs as requested. Partial Delivery indicates that SDUs can be delivered into more than one buffer or with multiple Reads. Incomplete Delivery indicates that SDUs with gaps can be delivered, i.e. missing PDUs. In this latter case, the API should indicate that a gap in the SDU is being delivered (usually by indicating an error with the Read). This error should differ from a read of a zero-length SDU.

5.3.2 Read/Read immediate action

```
If Read Then
   Copy as many SDUs as requested in the available buffers taking into
   account whether partial delivery is allowed.

   Updating RcvLeftWindowEdge and RcvRightWindowEdge has already occurred.

Fi
If Read Immediate Then
   Copy as many SDUs as requested from the list of available SDUs taking
   into account whether partial delivery is allowed. If that does not fulfil
   the request, go to the PDUReassemblyQueue and invoke Delimiting on the
   PDUs there to fulfil the request. This will require deactivating A-
   Timers and updating the RcvLeftWindowEdge and RcvRightWindowEdge and
   sending the appropriate Ack/Flow Control PDU.

Fi
```

5.3.3 Write action

Parameters: Port-id, pointer to a buffer, array of buffers, or list of buffers, a length in bytes.

This API call delivers SDUs to the delimiting module, which delimits SDUs into some number of User-data fields for the DTP task. Some OSs may assign semantics to reading a zero-length SDU. This implies that the API mustshall allow a zero-length SDU and not "optimize" it out of existence.