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

**Part 6:
RINA data transfer service**

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

Partie 6: Service de transfert de données RINA



IECNORM.COM : Click to view the full PDF of ISO/IEC 4396-6:2023



COPYRIGHT PROTECTED DOCUMENT

© ISO/IEC 2023

All rights reserved. Unless otherwise specified, or required in the context of its implementation, no part of this publication may be reproduced or utilized otherwise in any form or by any means, electronic or mechanical, including photocopying, or posting on the internet or an intranet, without prior written permission. Permission can be requested from either ISO at the address below or ISO's member body in the country of the requester.

ISO copyright office
CP 401 • Ch. de Blandonnet 8
CH-1214 Vernier, Geneva
Phone: +41 22 749 01 11
Email: copyright@iso.org
Website: www.iso.org

Published in Switzerland

Contents

Page

Foreword	iv
1 Scope	1
2 Normative references	1
3 Terms and definitions	1
4 Overview of the service	1
5 Detailed definition of the data transfer service	2
5.1 Requesting application process service description	2
5.1.1 Allocate_Request.submit	2
5.1.2 Allocate_Response.deliver	3
5.1.3 Transfer.submit	4
5.1.4 Transfer.deliver	4
5.1.5 Deallocate.submit	5
5.1.6 Deallocate.deliver	5
5.1.7 Status.submit	5
5.1.8 Status.deliver	6
5.1.9 Modify_Request.submit	6
5.1.10 Modify_Response.deliver	7
5.2 Requested Application Process Definition	8
5.2.1 Allocate_Request.deliver	8
5.2.2 Allocate_Response.submit	9
5.2.3 Transfer.submit	10
5.2.4 Transfer.deliver	10
5.2.5 Deallocate.submit	11
5.2.6 Deallocate.deliver	11
5.2.7 Status.submit	11
5.2.8 Status.deliver	12
5.2.9 Modify_Request.deliver	12
5.2.10 Modify_Response.submit	13
Bibliography	14

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/directives or www.iec.ch/members_experts/refdocs).

ISO and IEC draw attention to the possibility that the implementation of this document may involve the use 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 not 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.

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.html. In the IEC, see www.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-committees.

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

Part 6: RINA data transfer service

1 Scope

This document is a service definition that provides an abstract description of the application programming interface (API) seen by an Application Process using a distributed inter-process communication (IPC) facility (DIF). APIs reflect the specific constraints and conventions of an operating system or programming language. This document does not do that. A service definition specifies the interactions between an Application Process and IPC independent of such specifics.

The application process may be an IPC process and the member of a (N+1)-DIF. Actual APIs will be system specific (or may not exist at all), but this sequence of interaction will be maintained. The notation here is used to emphasize that the participants can only act on what they see and must not make assumptions about any events that may have occurred elsewhere. Hence the primitives are described in terms of primitives invoked locally to cause an action, submit, and primitives locally invoked to deliver information on state.

This is not a design for an API. It cannot be as a basis for any conformance tests. An actual API may make some, all or none of the parameters noted here visible to the user and may add additional primitives of local significance. The purpose of this service definition is to specify information that must or may be available by whatever means, explicit or implicit, to drive the operation of the DIF.

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 4396-1, *Telecommunications and information exchange between systems — Recursive Inter-Network Architecture — Part 1: Reference Model*

3 Terms and definitions

For the purposes of this document, the terms and definitions given in ISO 4396-1 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>
- IEC Electropedia: available at <https://www.electropedia.org/>

4 Overview of the service

An Application Process issues an `Allocate_Request.submit` requesting IPC resources to another Application Process. The request is processed by the Flow Allocator (see ISO 4396-8) to create and manage connections using the Error and Flow Control Protocol (see ISO 4396-9) and should result in a `Allocate_Request.deliver` to the requested Application Process. The requesting process may specify either "symmetric", the same QoS for both directions, or "asymmetric", a different QoS for incoming

and outgoing directions. The requested Application Process invokes an `Allocate_Response.submit` to indicate whether or not it accepts the request. This information is communicated to the requesting IPC Process, which invokes an `Allocate_Response.deliver`.

If the response is positive, the `Transfer.submit` and `deliver` primitives will be used by both Application Processes to transfer data. When either one determines that it is finished, the `Deallocate` primitives are used to deallocate the IPC resources associated with this instance. The Process using the flow (the same one that did the `Allocate` request) may decide to request a modification of the QoS parameters of such flow via the `Modify_Request.submit` service primitive. The request is processed by the DIF and should result in a `Modify_Request.deliver` to the Application Process at the other end of the flow. Such Application process invokes a `Modify_response.submit` to indicate whether it accepts or not the flow modification. This information is communicated to the requesting IPC Process, which invokes a `Modify_Response.deliver`.

5 Detailed definition of the data transfer service

5.1 Requesting application process service description

5.1.1 `Allocate_Request.submit`

5.1.1.1 `Allocate_Request.submit` (requested-APN, requesting-APN, sym/asym, QoS, port-id, access-control parameters, result)

5.1.1.1.1 When Invoked

This primitive is invoked by an Application Process to request the allocation of IPC resources with the destination application.

5.1.1.1.2 Action Upon Receipt

When a `Allocate_Request.submit` primitive is acted on, the Flow Allocator task of the IPC Process creates a Flow Allocator Instance (FAI) to manage the allocation of IPC resources necessary to fulfil the request. The identifier for the FAI-id, also called a port-id, is returned to the requestor to be used with all operations on this instance of IPC. A response is generated, once the FAI knows whether the request will be accepted. The action of the FAI can be found in the FAI specification.

NOTE The port-id used by the requesting AP is not the same as the one used by the requested AP, nor is it known to the requested AP or IPC Processes.

5.1.1.1.3 Parameters:

- Requested-APN – The application-process-name or a synonym for the requested application.

The term APN is used as a short-hand to denote that this parameter (and all others using APN) may include application-process-name and optionally, application-process-instance-id, application-entity-name, or application-entity-instance-identifier.

- Requesting-APN – The APN or a synonym for the requested application (see above).

Symmetric/Asymmetric (sym/asym) – This Boolean flag indicates whether the QoS parameter list is for a symmetric flow (the QoS is the same in both directions) or an asymmetric flow (the QoS for each direction is different).

- QoS – If the Sym/Asym flag indicates symmetric, then QoS is a single QoS-List. If the Sym/Asym flag indicates asymmetric, then QoS-List consists of Incoming: QoS-List, Outgoing: QoS-List. The `Transfer.submit` primitive will map to the Outgoing QoS-List. The `Transfer.deliver` primitive will be mapped to the Incoming QoS-List.

- Quality of Service Parameter List (QoS-List) – A list of Quality of Service parameters that the requesting AP desires should exist on the flow. The list of parameters is mostly optional. The list of parameters is:
 - Average SDU data-rate (measured in SDUs/sec)
 - Peak data-rate-duration (measured in bits/sec)
 - Peak SDU data-rate-duration (measured in SDUs/sec)
 - Burst period measured in μ secs
 - Burst duration, measured as a fraction of Burst Period
 - Undetected bit error rate measured as a probability
 - Partial Delivery – Can partial SDUs be delivered?
 - Order – Shall SDUs be delivered in order?
 - Max allowable gap in SDUs, (a gap of N SDUs is considered the same as all SDUs delivered, i.e. a gap of N is a “don’t care.”)
 - Delay in μ secs
 - Jitter in μ secs²
 - Cost-time, measured in \$/ms
 - Cost-bits, measured in \$/Mb
- Port-id – The requesting FAI-Id that is returned to the AP that may be used in referring to this instance of IPC.
- Access-control parameters – Access Control parameters (most likely a capability) for the Requesting-APN.
- Result – This parameter indicates whether the request was a success or failure. If a failure, the parameter may provide some indication of the severity of the failure. This is a placeholder in this call.

5.1.2 Allocate_Response.deliver

5.1.2.1 Allocate_Response.deliver (requested-APN, requesting-APN, sym/asym, QoS, port-id, access-control parameters, result)

5.1.2.1.1 When invoked

When the FAI is in the Allocate_Pending state and the IPCP processing the allocation request learns about the acceptance or rejection of the flow, it invokes this primitive to notify the Requesting Process about the success or failure of the allocation request associated with this port-id.

5.1.2.1.2 Action Upon Receipt

The Application Process AE and the FAI are in the Allocation_Pending state. If successful, the state transitions to Transfer state. The requesting Application Process may use the Transfer and Deallocate primitives to continue its task. If the result was negative, this request is terminated and returns to the NULL state. If the result indicates a reason that can be corrected, the requesting Application Process may formulate a new request.

5.1.2.1.3 Parameters:

(see [5.1.1.1.3](#) above)

5.1.3 Transfer.submit

5.1.3.1 Transfer.submit (port-id, sdu, BytesToSend, result)

5.1.3.1.1 When Invoked

When the FAI is in the Transfer state and the Process using the flow when it wants to send data, it invokes this primitive to send an SDU on the specified port-id.

5.1.3.1.2 Action Upon Receipt

When a Transfer.submit primitive is invoked, the SDU is delivered to the Connection-End-Point-Id, if this is an asymmetric flow, then it will be delivered to the CEP-id with Outgoing QoS and returns the result and BytesToSend parameters. The BytesToSend parameter indicates how many bytes of data the using process may transfer without an error. The using application may choose to ignore this parameter. Whether an API blocks the process is an API design issue beyond the scope of this document.

BytesToSend (or any interface flow control method) is a local matter and should not appear in a service definition. Hence it is not required, and other means are acceptable. However, it is part of the specification of the policies for the DIF to put appropriate constraints on interface flow control. None is not an acceptable policy.

5.1.3.1.3 Parameters:

- Port-id – The FAI-id that is returned to the using process that may be used in referring to this instance of IPC.
- SDU – The unit of data to be sent and whose identity is maintained upon delivery.
- BytesToSend – This parameter returns an indication to the user of how many bytes it can send. The using process may or may not use this parameter.
- Result – This parameter indicates whether the request was a success or failure. This is a local indication only, indicating that the request was well formed or if the using process has exceeded BytesToSend. It does indicate whether the SDU was delivered. If a failure, the parameter may provide some indication of the severity of the failure.

5.1.4 Transfer.deliver

5.1.4.1 Transfer.deliver (port-id, sdu, BytesToSend, result)

5.1.4.1.1 When Invoked

When the FAI is in the Transfer state and the IPC Process has one or more complete SDUs to deliver, it invokes this primitive is invoked to deliver an SDU on this port-id.

5.1.4.1.2 Action Upon Receipt

When a Transfer.deliver primitive is invoked, the SDU is delivered to the port-id that is bound to this flow, if this is an asymmetric flow the SDU will be delivered from the CEP-id with Incoming QoS and returns the Result and BytesToSend parameters. The BytesToSend parameter indicates how many bytes of data the using process may transfer without an error. The using application may choose to ignore this parameter. Whether an API blocks the process is an API design issue beyond the scope of this document.

5.1.4.1.3 Parameters:

(see [5.1.3.1.3](#) above)

5.1.5 Deallocate.submit**5.1.5.1 Deallocate.submit (port-id, result)****5.1.5.1.1 When Invoked**

This primitive is invoked by the Process using the flow in any state, to request the deallocation of the flow identified by port-id and its associated resources.

5.1.5.1.2 Action Upon Receipt

When a Deallocate.submit primitive is acted on, the FAI will take the necessary steps to deallocate all resources associated with this instance and terminate.

5.1.5.1.3 Parameters:

- Port-id – The FAI-id that is returned to the using process that may be used in referring to this instance of IPC.
- Result – This parameter indicates whether the request was a success or failure. If a failure, the parameter may provide some indication of the severity of the failure. This is a placeholder in this call.

5.1.6 Deallocate.deliver**5.1.6.1 Deallocate.deliver (port-id, result)****5.1.6.1.1 When Invoked**

This primitive is invoked in any state by an FAI to notify the local Process that all the resources allocated to the flow associated to port-id are released.

5.1.6.1.2 Action Upon Receipt

When a Deallocate.deliver primitive is acted on, the Application Process takes any action necessary to clean up.

5.1.6.1.3 Parameters:

(see [5.1.5.1.3](#) above)

5.1.7 Status.submit**5.1.7.1 Status.submit (port-id, BytesToSend, QoS-parameters, result)****5.1.7.1.1 When invoked**

This primitive is invoked at any time by the Process using the flow when it wants to obtain a status on the flow.

5.1.7.1.2 Action upon Receipt

On receipt of the status.submit primitive, The Local IPC process generates a response in return. The status.submit returns all information about the flow that is available to the using process, including its state in the result parameter, but not its address.

5.1.7.1.3 Parameters:

- Port-id – The FAI-id that is returned to the using process that may be used in referring to this instance of IPC. This is the port_Id that is passed to it in Allocate_response.indicate.
- BytesToSend – This parameter returns an indication to the user of how many bytes it can send. The using process may or may not use this parameter.
- QoS Parameters – (see QoS Parameters in [5.1.1.1.3](#)).
- Result – This parameter indicates whether the request was a success or failure. This is a local indication only, indicating that the request was well formed or if the using process has exceeded BytesToSend. If a failure, the parameter may provide some indication of the severity of the failure.

5.1.8 Status.deliver

5.1.8.1 Status.deliver (port-id, BytesToSend, QoS-parameters, result)

5.1.8.1.1 When invoked

This primitive is invoked to notify the Process using the flow of significant changes in the status of the flow.

5.1.8.1.2 Action upon Receipt

The Application performs any action it deems necessary.

5.1.8.1.3 Parameters:

- Port-id – The FAI-id that is returned to the using process that may be used in referring to this instance of IPC. This is the port_Id that is passed to it in Allocate_response.indicate.
- BytesToSend – This parameter returns an indication to the user of how many bytes it can send. The using process may or may not use this parameter.
- QoS Parameters – (see QoS Parameters in [5.1.1.1.3](#)).
- Result – This parameter indicates whether the request was a success or failure. This is a local indication only, indicating that the request was well formed or if the using process has exceeded BytesToSend. If a failure, the parameter may provide some indication of the severity of the failure.

5.1.9 Modify_Request.submit

5.1.9.1 Modify_Request.submit (port-id, sync/async, outgoing/incoming, QoS, result)

5.1.9.1.1 When invoked

When the FAI is in the Transfer state the Process using the flow invokes this primitive to request the modification of one or more QoS parameters associated to the flow.

5.1.9.1.2 Action Upon Receipt

When a `Modify_Request.submit` primitive is acted on, the relevant FAI (as identified by the port-id) is informed about the request, and provided with the new service parameters. A response is generated once the FAI knows whether the modification request will be accepted. The action of the FAI can be found in the FAI specification.

5.1.9.1.3 Parameters:

- Port-id – The requesting FAI-Id that is returned to the AP that may be used in referring to this instance of IPC.
- sync/async – A Boolean that defines whether sending PDUs should be paused until the modification operation is completed. If the Boolean is True ('sync' mode) no PDUs should be sent until the operation is completed, otherwise ('async' mode) PDU transmission will not be paused.
- Outgoing/incoming – if this flow is asymmetric, this indicates which direction is to have its QoS modified. "Outgoing" indicates the QoS of the connection that Writes are mapped to. "Incoming" indicates the QoS of the connection Reads are mapped to will modified.
- QoS – If the Sym/Asym flag indicates symmetric, then QoS is a single QoS-List. If the Sym/Asym flag indicates asymmetric, then QoS-List consists of Incoming: QoS-List, Outgoing: QoS-List. The `Transfer.submit` primitive will map to the Outgoing QoS-List. The `Transfer.deliver` primitive will be mapped to the Incoming QoS-List.
- Quality of Service Parameter List (QoS-List) – (see QoS in [5.1.1.1.3](#)).
- Result – This parameter indicates whether the request was a success or failure. If a failure, the parameter may provide some indication of the severity of the failure. This is a placeholder in this call.

5.1.10 `Modify_Response.deliver`

5.1.10.1 `Modify_Response.deliver` (QoS parameters, port-id, outgoing/incoming, result)

5.1.10.1.1 When invoked

When the FAI is in the `Modification_pending` state the IPCP invokes this primitive to notify the local Process about the success or failure of the flow modification request associated with this port-id.

5.1.10.1.2 Action Upon Receipt

The Application Process AE and the FAI are in the `Modification_Pending` state. If successful, the state transitions to `Transfer` state and the flow properties are updated to the new QoS Parameters. The requesting Application Process may use the `Transfer` and `Deallocate` primitives to continue its task. If the result was negative, this request also transitions to the `Transfer` state, without changing the flow properties.

5.1.10.1.3 Parameters:

- Port-id – The requesting FAI-Id that is returned to the AP that may be used in referring to this instance of IPC.
- sync/async – A Boolean that defines whether sending PDUs should be paused until the modification operation is completed. If the Boolean is True ('sync' mode) no PDUs should be sent until the operation is completed, otherwise ('async' mode) PDU transmission will not be paused.

- Outgoing/incoming – if this flow is asymmetric, this indicates which direction is to have its QoS modified. "Outgoing" indicates the QoS of the connection that Writes are mapped to. "Incoming" indicates the QoS of the connection Reads are mapped to will modified.
- QoS – If the Sym/Asym flag indicates symmetric, then QoS is a single QoS-List. If the Sym/Asym flag indicates asymmetric, then QoS-List consists of Incoming: QoS-List, Outgoing: QoS-List. The Transfer.submit primitive will map to the Outgoing QoS-List. The Transfer.deliver primitive will be mapped to the Incoming QoS-List.
- Quality of Service Parameter List (QoS-List) – (see QoS in [5.1.1.1.3](#)).
- Result – This parameter indicates whether the request was a success or failure. If a failure, the parameter may provide some indication of the severity of the failure. This is a placeholder in this call.

5.2 Requested Application Process Definition

5.2.1 Allocate_Request.deliver

5.2.1.1 Allocate_Request.deliver (Requesting-APN, Requested-APN, sym/asym, QoS, port-id, access-control, result)

5.2.1.1.1 When invoked

When a Create_Request primitive is received at the requested IPC process, the Flow Allocator of the IPC Process determines whether the requesting AP has access to the requested AP. If not, a negative Create_Response is generated and sent to the requesting FAI. No FAI is created by the Requested Flow Allocator.

If the resources exist and the requesting AP has access to the requested AP, an FAI is created to manage the allocation of IPC resources necessary to fulfil the request. The identifier for the FAI, the port-id, is used in this primitive as a reference for all operations relating to this instance. The action of the FAI can be found in the Flow Allocator specification. This primitive is invoked by the Flow Allocator which can provide IPC to the requested Application Process to determine whether the requested Application Process will accept the allocation.

5.2.1.1.2 Action Upon Receipt

The Requested Application Process determines whether it can accept the request and replies accordingly by invoking an Allocate_Response.submit.

NOTE The port-id used by the requesting AP is not the same as used by the requested AP, nor is it known to the requested AP or IPC Processes.

5.2.1.1.3 Parameters:

- Requested-APN – The application-process-name or a synonym for the requested application.

The term APN is used as a short-hand to denote that this parameter (and all others using APN) may include application-process-name and optionally, application-process-instance-id, application-entity-name, or application-entity-instance-identifier.

- Requesting-APN – The APN or a synonym for the requested application. (see above).

Symmetric/Asymmetric (sym/asym) - This Boolean flag indicates whether the QoS parameter list is for a symmetric flow (the QoS is the same in both directions) or an asymmetric flow (the QoS for each direction is different).

- QoS – If the Sym/Asym flag indicates symmetric, then QoS is a single QoS-List. If the Sym/Asym flag indicates asymmetric, then QoS-List consists of Incoming: QoS-List, Outgoing: QoS-List. The Transfer.submit primitive will map to the Outgoing QoS-List. The Transfer.deliver primitive will be mapped to the Incoming QoS-List.
- Quality of Service Parameter List (QoS-List) – A list of Quality of Service parameters that the requesting AP desires should exist on the flow. The list of parameters is mostly optional. The list of parameters is:
 - Average data-rate (measured at the application in bits/sec)
 - Average SDU data-rate (measured in SDUs/sec)
 - Peak data-rate-duration (measured in bits/sec)
 - Peak SDU data-rate-duration (measured in SDUs/sec)
 - Burst period measured in μ secs
 - Burst duration, measured as a fraction of Burst Period
 - Undetected bit error rate measured as a probability
 - Partial Delivery – Can partial SDUs be delivered?
 - Order – Shall SDUs be delivered in order?
 - Max allowable gap in SDUs, (a gap of N SDUs is considered the same as all SDUs delivered, i.e. a gap of N is a “don’t care.”)
 - Delay in μ secs
 - Jitter in μ secs²
 - Cost-time, measured in \$/ms
 - Cost-bits, measured in \$/Mb
- Port-id – The requesting FAI-Id that is returned to the AP that may be used in referring to this instance of IPC.
- Access-control parameters – Access Control parameters (most likely a capability) for the Requesting-APN.
- Result – This parameter indicates whether the request was a success or failure. If a failure, the parameter may provide some indication of the severity of the failure. This is a placeholder in this call.

5.2.2 Allocate_Response.submit

5.2.2.1 Allocate_Response.submit (port-id, result)

5.2.2.1.1 When invoked

When an Allocate_Request.deliver primitive is received at the requested AP, the Application Process will determine if it can honour the request and respond with an Allocate_response to the FAI in the destination IPC process. This primitive is invoked by the requested Application Process to respond to an allocation request from IPC.

5.2.2.1.2 Action Upon Receipt

The FAI formulates a Create_Response to the requesting FAI and if it was successful binds the port-id to the appropriate Connection-Endpoint-Id to allow data transfer to commence.

5.2.2.1.3 Parameters:

(see 5.2.1.1.3 above)

5.2.3 Transfer.submit

5.2.3.1 Transfer.submit (port-id, sdu, BytesToSend, result)

5.2.3.1.1 When Invoked

This primitive is invoked by the Process when in the Transfer state to send an SDU on the specified port-id.

5.2.3.1.2 Action Upon Receipt

When a Transfer.submit primitive is invoked, the SDU is delivered to the Connection-Endpoint-Id and returns the result and BytesToSend parameters. The BytesToSend parameter indicates how many bytes of data the using process may transfer without an error. The using application may choose to ignore this parameter. Whether an API blocks the process is an API design issue beyond the scope of this document.

BytesToSend (or any interface flow control method) is a local matter and should not appear in a service definition. Hence it is not required, and other means are acceptable. However, it is part of the specification of the policies for the DIF to put appropriate constraints on interface flow control. None is not an acceptable policy.

5.2.3.1.3 Parameters:

- Port-id – The FAI-id that is returned to the using process that may be used in referring to this instance of IPC. This is the port_Id that is passed to it in Allocate_response.indicate.
- SDU – The unit of data to be sent and whose identity is maintained upon delivery.
- BytesToSend – This parameter returns an indication to the user of how many bytes it can send. The using process may or may not use this parameter.
- Result – This parameter indicates whether the request was a success or failure. This is a local indication only, indicating that the request was well formed or if the using process has exceeded BytesToSend. It does indicate whether the SDU was delivered. If a failure, the parameter may provide some indication of the severity of the failure.

5.2.4 Transfer.deliver

5.2.4.1 Transfer.deliver (port-id, sdu, result)

5.2.4.1.1 When Invoked

This primitive is invoked when in the Transfer state to deliver an SDU on this port-id.

5.2.4.1.2 Action Upon Receipt

When a Transfer.deliver primitive is invoked, the SDU is delivered to the port-id that is bound to this flow and returns the Result and BytesToSend parameters. The BytesToSend parameter indicates how many bytes of data the using process may transfer without an error. The using application may choose

to ignore this parameter. Whether an API blocks the process is an API design issue beyond the scope of this document.

5.2.4.1.3 Parameters:

(see [5.2.3.1.3](#) above)

5.2.5 Deallocate.submit

5.2.5.1 Deallocate.submit (port-id, result)

5.2.5.1.1 When Invoked

This primitive is invoked by the Application Process in any state to deallocate all resources allocated to this instance.

5.2.5.1.2 Action Upon Receipt

When a Deallocate.submit primitive is acted on, the FAI will take the necessary steps to deallocate the resources associated with this instance and terminate.

5.2.5.1.3 Parameters:

- Port-id – The FAI-id that is returned to the using process that may be used in referring to this instance of IPC. This is the port_Id that is passed to it in Allocate_response.indicate.
- Result – This parameter indicates whether the request was a success or failure. If a failure, the parameter may provide some indication of the severity of the failure. This is a place holder in this call.

5.2.6 Deallocate.deliver

5.2.6.1 Deallocate.deliver (port-id, result)

5.2.6.1.1 When Invoked

This primitive is invoked in any state by an FAI to notify the local application process all the resources allocated to this instance are released.

5.2.6.1.2 Action Upon Receipt

When a Deallocate.deliver primitive is acted on, the Application Process takes any action necessary to clean up.

5.2.6.1.3 Parameters:

(see [5.2.5.1.3](#) above)

5.2.7 Status.submit

5.2.7.1 Status.submit (port-id, BytesToSend, QoS-parameters, result)

5.2.7.1.1 When invoked

This primitive is invoked by the Application anytime it wants to obtain a status on the flow.

5.2.7.1.2 Action upon Receipt

On receipt of the Status.submit primitive, The Local IPC process generates a response in return. The status.submit returns all information about the flow that is available to the using process, including its state in the result parameter, but not its address.

5.2.7.1.3 Parameters

- Port-id – The FAI-id that is returned to the using process that may be used in referring to this instance of IPC. This is the port_Id that is passed to it in Allocate_response.indicate.
- BytesToSend – This parameter returns an indication to the user of how many bytes it can send. The using process may or may not use this parameter.
- QoS Parameters – (see QoS Parameters in [5.1.1.1.3](#)).
- Result – This parameter indicates whether the request was a success or failure. This is a local indication only, indicating that the request was well formed or if the using process has exceeded BytesToSend. If a failure, the parameter may provide some indication of the severity of the failure.

5.2.8 Status.deliver

5.2.8.1 Status.deliver (port-id, BytesToSend, QoS-parameters, result)

5.2.8.1.1 When invoked

This primitive is invoked to notify the using process of significant changes in the status of the flow.

5.2.8.1.2 Action upon Receipt

The Application performs any action it deems necessary.

5.2.8.1.3 Parameters

- Port-id – The FAI-id that is returned to the using process that may be used in referring to this instance of IPC. This is the port_Id that is passed to it in Allocate_response.indicate.
- BytesToSend – This parameter returns an indication to the user of how many bytes it can send. The using process may or may not use this parameter.
- QoS Parameters – (see QoS Parameters in [5.1.1.1.3](#)).
- Result – This parameter indicates whether the request was a success or failure. This is a local indication only, indicating that the request was well formed or if the using process has exceeded BytesToSend. If a failure, the parameter may provide some indication of the severity of the failure.

5.2.9 Modify_Request.deliver

5.2.9.1 Modify_Request.deliver (port-id, sync/async, outgoing/incoming, QoS, result)

5.2.9.1.1 When invoked

When a Write_Request primitive is received at the requested IPC process, the relevant FAI at the IPC Process (identified by the port-id parameter) is informed. The action of the FAI can be found at the Flow Allocator specification (see ISO 4396-7).