Network Working Group
Internet-Draft
Intended status: Experimental
Expires: October 25, 2014

M. Boucadair
C. Jacquenet
France Telecom
April 23, 2014

Connectivity Provisioning Negotiation Protocol (CPNP)
draft-boucadair-connectivity-provisioning-protocol-03

Abstract

This document specifies the Connectivity Provisioning Negotiation Protocol (CPNP) which is used to facilitate the dynamic negotiation of service parameters between a Customer and a Provider.

CPNP is a generic protocol that can be used for various negotiation purposes that include (but are not necessarily limited to) connectivity provisioning services, storage facilities, CDN (Content Delivery Networks) footprint, etc. CPNP can be extended with new Information Elements.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at http://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on October 25, 2014.

Table of Contents

1. Introduction

This document defines the Connectivity Provisioning Negotiation Protocol (CPNP) that is meant to dynamically exchange and negotiate the connectivity provisioning parameters, and other service-specific parameters, between a Customer and a Provider. CPNP is a tool that introduces automation in the service negotiation and activation procedures, thus fostering the overall service delivery process. CPNP can be seen as a component of the dynamic negotiation meta-domain described in Section 3.4 of [RFC7149].

CPNP is a generic protocol that can be used for other negotiation purposes than connectivity provisioning. For example, CPNP can be used to request extra storage resources, to extend the footprint of a CDN (Content Delivery Networks), to enable additional features from a cloud Provider, etc. CPNP can be extended with new Information Elements (IEs).

[I-D.boucadair-connectivity-provisioning-profile] describes a Connectivity Provisioning Profile (CPP) template to capture connectivity requirements to be met by a transport infrastructure for the delivery of various services such as Voice over IP, IPTV, and VPN services. The CPP document defines the set of IP transfer parameters that reflect the guarantees that can be provided by the underlying transport network together with a reachability scope and capacity needs. CPNP uses the CPP template to encode connectivity provisioning clauses.

As a reminder, several proposals have been made in the past by the community (e.g., COPS-SLS, SrNP (Service Negotiation Protocol), DSNP (Dynamic Service Negotiation Protocol, RNAP (Resource Negotiation and Pricing Protocol), SNAP (Service Negotiation and Acquisition Protocol), etc.). None of these proposals has been standardized by IETF. It is out of the scope of this document to elaborate on the differences between CPNP and the aforementioned proposals.

This document is organized as follows:

- Section 3 defines functional elements involved in CPNP exchanges.
- Section 4 introduces several order processing models and precises those that are targeted by CPNP.
- Section 5 enumerates a non-exhaustive list of use cases that would benefit from CPNP.
- Section 5 discusses some CPNP deployment models.
- Section 7 presents CPNP negotiation model.
- Section 8 provides an overview of the protocol.
- Section 9 specifies CPNP objects.
- Section 10 describes CPNP message validation procedure.
- Section 11 specifies the behavior of involved CPNP functional elements.
- Section 12 discusses some operational guidelines.

2. Terminology

This document makes use of the following terms:

- Customer: is a business role which denotes an entity that is involved in the definition and the possible negotiation of a contract, including Connectivity Provisioning Agreement, with a

Provider. A connectivity provisioning contract is captured in a dedicated CPP template, which specifies (among other information): the sites to be connected, border nodes, outsourced operations (e.g., routing, force via points). The right to invoke the subscribed service may be delegated by the Customer to third-party End Users, or brokering services.

- Network Provider (or Provider): owns and administers one or many transport domain(s) (typically Autonomous System (AS)) composed of IP switching and transmission resources (e.g., routing, switching, forwarding, etc.). Network Providers are responsible for ensuring connectivity services (e.g., offering global or restricted reachability). Connectivity services offered to Customers are captured in contracts from which are derived the technology-specific clauses and policies to be enforced by the components involved in the connectivity service delivery. Offered connectivity services are not restricted to IP.

3. CPNP Functional Elements

The following functional elements are defined:

- CPNP client (or client): denotes a software instance that sends CPNP requests and receives CPNP responses. The current operations that can be performed by a CPNP client are listed below:

  1. Create a quotation order (Section 11.1.1).

  2. Cancel an ongoing quotation order under negotiation (Section 11.1.1).

  3. Withdraw a pre-negotiated order (Section 11.1.2).

  4. Update a pre-negotiated order (Section 11.1.3).

- CPNP server (or server): denotes a software instance that receives CPNP requests and sends back CPNP responses accordingly. The CPNP server is responsible for the following operations:

  1. Quotation order handling (Section 11.2.1).

  2. Cancel an ongoing quotation order (Section 11.2.2).

  3. Handling of an order withdrawal (Section 11.2.3).

4. Order Processing Models

There are basically three models for customer's order processing purposes:

1. Frozen model: The Customer cannot negotiate the parameters of the service delivered by a Provider. After consulting a service portfolio, the Customer selects the offer he/she wants to subscribe and places an order to the Provider. Order handling is quite simple on the Provider side because the service is not customized as per Customer's requirements, but rather pre-designed to target a group of customers having similar requirements (and who therefore share the same Customer Provisioning Profile).

2. Announcement model: The Provider proceeds to the announcement of a set of services templates. The Customer can then initiate a negotiation cycle using these templates to prepare its request order.

3. Negotiation-based model: Unlike the frozen model, the Customer documents his/her requirements in some kind of request for quotation which is then sent to one or several Providers. Solicited Providers then check whether they can address these requirements or not, and get back to the Customer accordingly, possibly with an offer that may not exactly match customer's requirements (e.g., a 100 Mbps connection cannot be provisioned given the amount of available resources, but an 80 Mbps connection can be provided). A negotiation between the Customer and the Provider then follows, and an order is placed by the Customer eventually, upon completion of the negotiation phase.

Even if the frozen model could also yield the instantiation of a CPP template and CPNP could be used to display the said CPP to the Customer and confirm the order is being processed/delivered, etc., this document focuses on the negotiation-based model.

Order processing management on the Network Provider's side is usually connected with the following functional blocks:

- Network Provisioning (including Order Activation, Network Planning, etc.)

- AAA

- Sales-related functional blocks (e.g., billing, Customers credit checks validation, etc.)

o Network Impact Analysis

CPNP does not assume any specific knowledge about these functional blocks, but the instantiation of the connectivity provisioning requests may be conditioned by the information manipulated by some of these blocks. For example, the resources that can be allocated to accommodate the Customer's requirements may depend on the network planning policy as well as the number of orders to be processed simultaneously over a given period of time.

This document does not elaborate on how Customers are identified and managed by the Provider's Information System.

5. Sample Use Cases

A list of CPNP use cases is provided below:

1. [RFC4176] introduces the L3VPN Service Order Management functional block which is responsible for managing the requests initiated by the Customers and tracks the status of the completion of the related operations. CPNP can be used between the Customer and the Provider to negotiate L3VPN service parameters. A CPNP server could therefore be part of the L3VPN Service Order Management functional block discussed in [RFC4176].

2. CPNP can be used between two adjacent domains to deliver IP interconnect services (e.g., enable, update, disconnect). For example, two ASes can be connected via several interconnection points. CPNP can be used between these ASes to upgrade existing links, request additional resources, provision a new interconnection point, etc.

3. An integrated Provider can use CPNP to rationalize connectivity provisioning needs related to its service portfolio. A CPNP server function is used by network operations teams. A CPNP interface to invoke CPNP negotiation cycles is exposed to service management teams.

4. Service Providers can initiate connectivity provisioning requests towards Network Providers using CPNP. Multiple CPNP ordering cycles can be initiated by a Service Provider towards multiple Network Providers. Only a subset of these orders may be put into effect.

5. CPNP can be used in M2M environments to dynamically subscribe to M2M services (e.g., access to data retrieved by a set of sensors, extend sensor coverage, etc.).

6. A Provider offering cloud services can expose a CPNP interface to allow Customers to dynamically negotiate the features they want to subscribe to. These features can be for instance: request additional storage resources, enable security filters, etc.

7. CDN Providers can use CPNP to extend their footprint by interconnecting their CDN infrastructure [RFC6770] (see Figure 1).

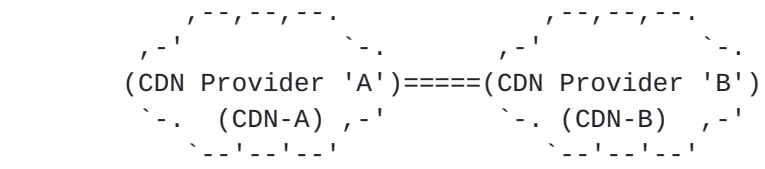

Figure 1: CDN Interconnection

6. Deployment Models

Several CPNP deployment models can be envisaged. Two examples are listed below:

- The Customer deploys a CPNP client while one or several CPNP servers are deployed by the Provider.

- The Customer does not enable any CPNP client. The Provider maintains a Customer Order Management portal. The Customer can initiate connectivity provisioning quotation orders via the portal; appropriate CPNP messages are then generated and sent to the relevant CPNP server. In this model, both the CPNP client and CPNP server are under the responsibility of the same administrative entity (i.e., Network Provider).

Once the negotiation of connectivity provisioning parameters is completed and an order has been placed by the Customer accordingly, the actual network provisioning operations are initiated. The specification of corresponding dynamic resource allocation and policy enforcement schemes, as well as how CPNP servers interact with network provisioning functional blocks are out of the scope of this document.

This document does not make any assumption about the CPNP deployment model either.

7. CPNP Negotiation Model

CPNP adopts a Quotation Order/Offer/Answer model where:

1. The client specifies its requirements via a Provision Quotation Order (PQO).

2. The server makes an offer to either address the requirements of the PQO or suggests a counter-proposal that partially addresses the requirements of the PQO or declines the PQO.

3. The client either accepts or declines the offer.

The server may support the optional functionality to publish available services to the clients. Dedicated templates can be defined for the purpose of service announcements. The client will use these templates to initiate its CPNP negotiation cycle.

Only one Offer/Answer stage is assumed within one single CPNP transaction (Section 8.6). Nevertheless, multiple CPNP transactions can be undertaken by the CPNP client (see Figure 2).

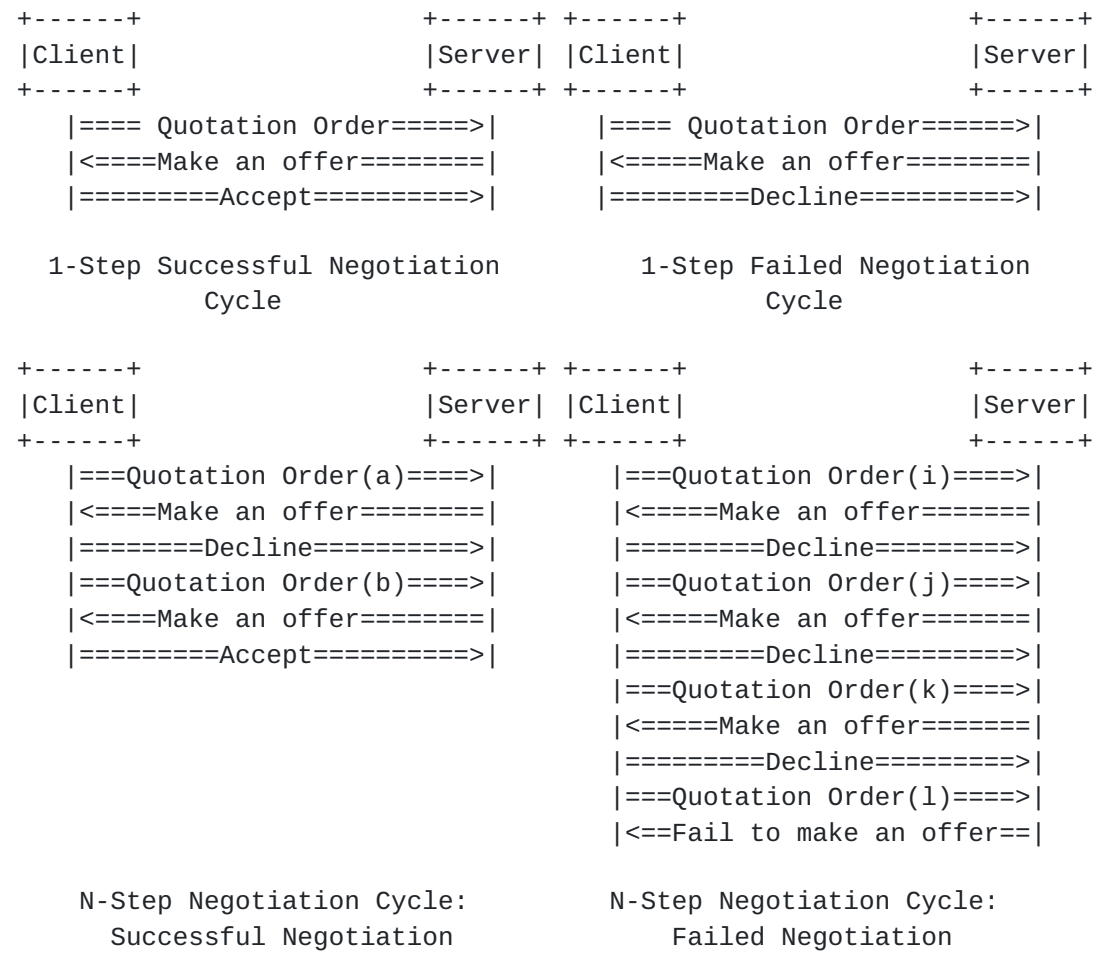

Figure 2: Overall Negotiation Process

8. Protocol Overview

8.1. Client/Server Communication

CPNP is a client/server protocol which runs over UDP. No permanent CPNP session needs to be maintained between the client and the server. There is no need to run CPNP over a reliable transport mode because CPNP messages are acknowledged.

The server uses CPNP_PORT (see Section 14) to bind the CPNP service. CPNP client sends messages to CPNP_PORT. The same port is used as the source port of the request sent to the server to document service requirements, and must be used by the client to listen to messages sent by the server.

CPNP messages can also be transported over DTLS [RFC6347].

CPNP is independent of the IP address family.

8.2. Server Discovery

The CPNP client can be configured with the CPNP server using manual or dynamic configuration means. For example, Providers may configure dedicated SRV records ([RFC2782]).

Discussions about how the client discovers its server(s) are out of the scope of this document. The document assumes a CPNP server can be reached by the CPNP client, thanks to some configuration means.

8.3. Policy Configuration on the CPNP Server

As an input to the CPNP server's decision-making process, the CPNP server may be connected to various external modules such as: Customer Profiles, Network Topology, Network Resource Management, Orders Repository, AAA or Network Provisioning Manager (an example is shown in Figure 3).

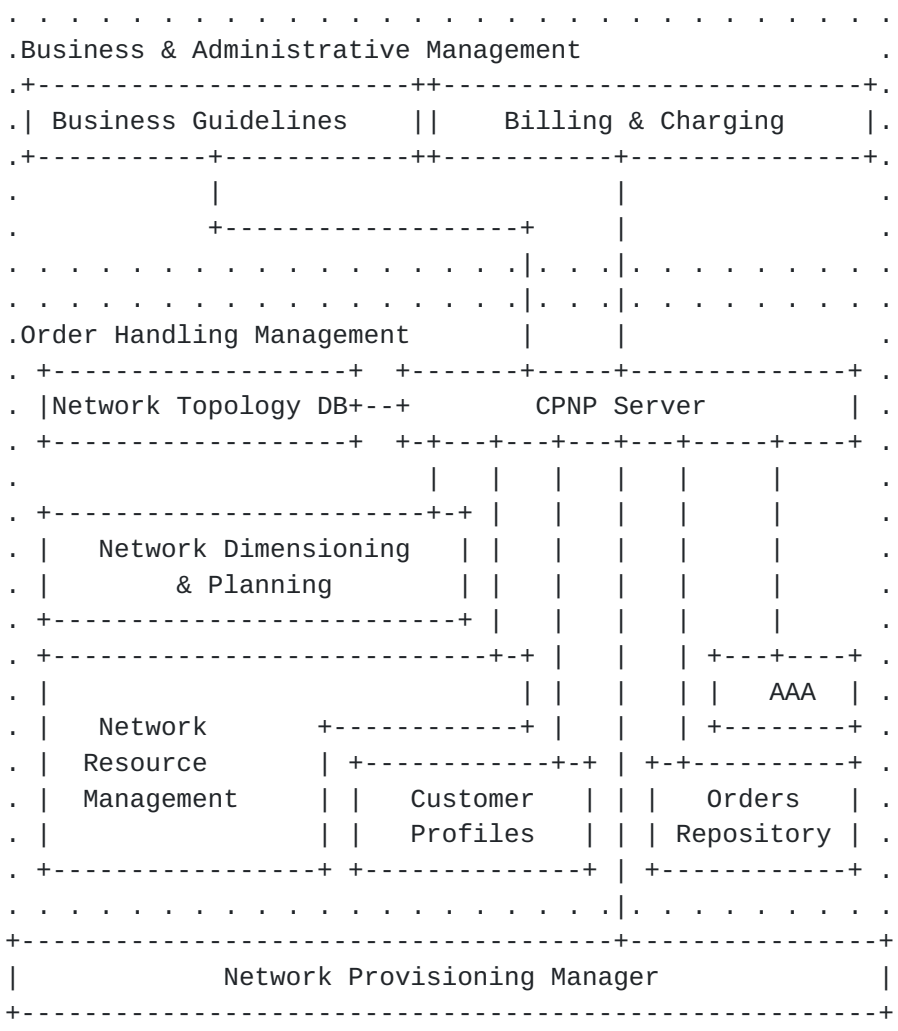

Figure 3: Order Handling Management Functional Block

These external modules provide inputs to the CPNP server, so that it can:

- Check whether a customer is entitled to initiate the connectivity provisioning quotation request.
- Check whether a customer is entitled to cancel the order
- Check whether administrative data (e.g., billing-related information) have been verified before starting handling the request
- Check whether network capacity is available or additional capacity is required

- Receive guidelines from network design and sales blocks (e.g., cost, increase network usage, threshold of how many CPP templates can be processed over a given period of time as a function of the nature of the service to be delivered, etc.)

- Transfer completed orders to network provisioning blocks

- Etc.

The above list of CPNP server operations is not exhaustive.

The following order handling modes can be also configured on the server:

1. Fully automated mode: This mode does not require any action from the administrator when receiving a request for a service. The server can execute its decision-making process related to the orders received and generate corresponding offers.

2. Administrative validation checking: Some or all server's operations are subject to administrative validation procedures. This mode requires an action from the administrator for every request received by the CPNP server. CPNP methods which can be automatically handled or subject to one or several validation administrative checks can be configured on the server.

8.4. CPNP Session

Both the client and server maintain the following CPNP transport session information:

A CPNP session is identified by the following items:

- IP address of the client

- Client's port number

- IP address of the server

- Server's port number

8.5. Extended CPNP Session

An extended PQO session denotes a 4-uplet defined as follows:

- CPNP session

- Incremented Sequence Number

- Customer Order Identifier
- Provider Order Identifier
- Transaction-ID

8.6. CPNP Transaction

A CPNP transaction occurs between a client and a server and comprises all CPNP messages, from the first request sent by the client to the server until a final response sent by the server to the client and which completes the transaction. The CPNP transaction is bound to a CPNP session.

Because multiple CPNP transactions can be maintained by the CPNP client, the client must assign an identifier to uniquely identify a given transaction. This identifier is denoted as Transaction-ID.

The Transaction-ID must be randomly assigned be the CPNP client, according to the best current practice for generating random numbers [RFC4086] that cannot be guessed easily. Transaction-ID is used as part of the validation of CPNP responses received by the client.

Using cryptographically random Transaction-IDs provides some protection against session hijacking.

Sequence Number is set in the first CPNP message by the client. This number is then incremented fro each request within the same CPNP transaction.

8.7. CPNP Operations

The current CPNP operations are listed below. They may be augmented, depending on the nature of some transactions or because of security considerations that may suggest a CPNP client/server authentication phase before negotiation begins.

- PROVISION: This operation is used to initiate a connectivity provisioning quotation order. Upon receipt of a PROVISION request, the server may response with a PROCESSING, OFFER or a FAIL message.
- PROCESSING: This operation is used to inform the remote party the message was received and that the order quotation or the offer is being processed.

- OFFER: This operation is used by the server to inform the client about an Offer that is supposed to best accommodate the requirements indicated in the PROVISION message.

- ACCEPT: This operation is used to confirm the acceptance of an offer made by the server.

- ACK: This operation is used by the server to acknowledge the receipt of an ACCEPT or WITHDRAW message.

- DECLINE: This operation is used by a client to reject an offer made by the server.

- CANCEL: This operation is used by the client to cancel an ongoing connectivity provisioning quotation order.

- WITHDRAW: This operation is used by the client to withdraw a pre-negotiated connectivity provisioning order.

- UPDATE: This operation is used by the client to update an existing connectivity provisioning order. For example, this method can be invoked to add a new site. This method will trigger a new negotiation cycle.

- FAIL: This operation is used by the server when it cannot accommodate the requirements documented in the PQO conveyed in the PROVISION message. This operation can also be used by a server to inform the client about an error encountered when processing the received message. The message includes the status code which provides more information about the error.

CPNP primitives are service-independent. This document specifies objects that are required for connectivity provisioning negotiation (see Section 8.8). Additional service-specific objects to be carried in CPNP messages can be defined in the future to accommodate other deployment needs.

8.8. Connectivity Provisioning Documents

CPNP makes use of several flavors of Connectivity Provisioning Documents (CPD). These documents follow the CPP template described in [I-D.boucadair-connectivity-provisioning-profile].

- Requested Connectivity Provisioning Document: refers to the CPD included by a CPNP client in a PROVISION request.

- Offered Connectivity Provisioning Document: This document is included by a CPNP server in an OFFER message. This information

reflects the proposal of the server to accommodate all or a subset of the clauses depicted in a CPD. A validity date is associated with the offer.

- Agreed Connectivity Provisioning Document: If the client accepts the offer, the offered CPD is included in an ACCEPT message. This CPD is also included in an ACK message.

Figure 4 shows a typical CPNP transaction and the use of Connectivity Provisioning Documents:

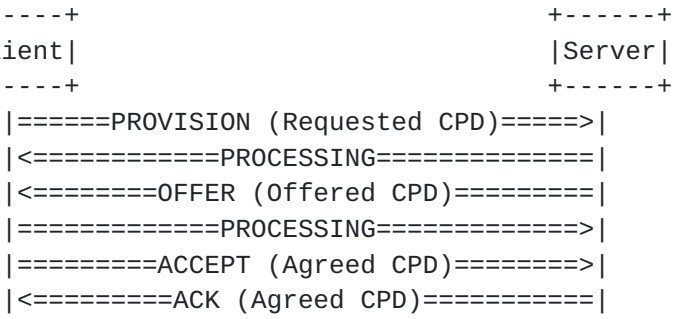

Figure 4: Connectivity Provisioning Documents

8.9. Child Connectivity Provisioning Quotation Orders

If the server detects network resources from another Network Provider need to be allocated in order to accommodate the requirements described in a PQO (e.g., context of an inter-domain VPN service where additional PE router resources need to be allocated), the server may generate child PQOs to request the appropriate network provisioning operations (see Figure 5). In such situation, the server behaves as a CPNP client. The server associates the parent order with its child PQOs. This is typically achieved by locally adding the reference of the child PQO to the parent order.

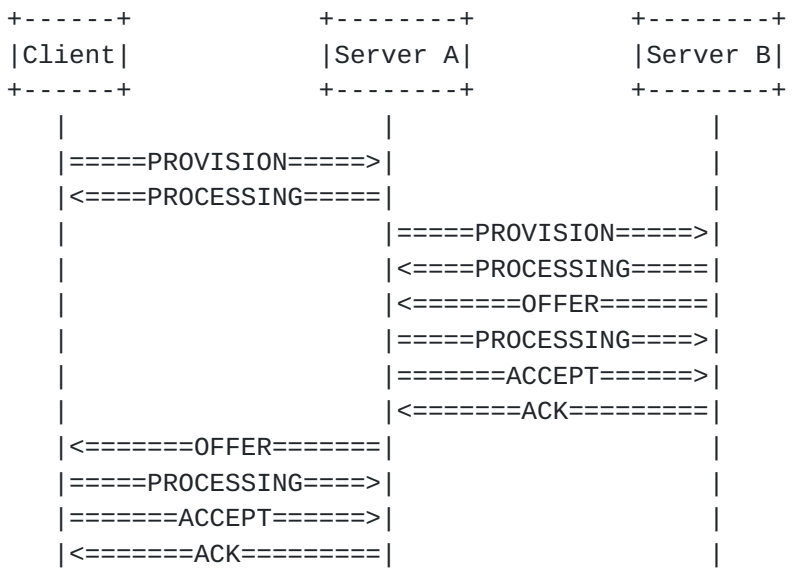

Figure 5: Example of Child Orders

8.10. State Management

Both the client and the server maintain repositories to store ongoing orders. How these repositories are maintained is deployment-specific. It is out of scope of this document to elaborate on such considerations.Timestamps are also logged to track state change. Tracking may be needed for various reasons (including regulatory ones).

8.10.1. On the Client Side

The following lists the states which can be associated with a given order on the client's side:

- Created: when the order has been created. It is not handled by the client until the administrator allows to process it.
- AwaitingProcessing: when the administrator ordered to process an order and this order is not handled yet.
- PQOSent: when the PQO has been sent to the server.
- ServerProcessing: when the server has confirmed the receipt of the order.
- OfferReceived: when an offer has been received from the server.
- OfferProcessing: when a received offer is currently processed by the client.
- AcceptSent: when the client confirmed the offer to the server.
- AcceptAck: when the offer is acknowledged by the server.
- Cancelled: when the order has failed or has been cancelled.

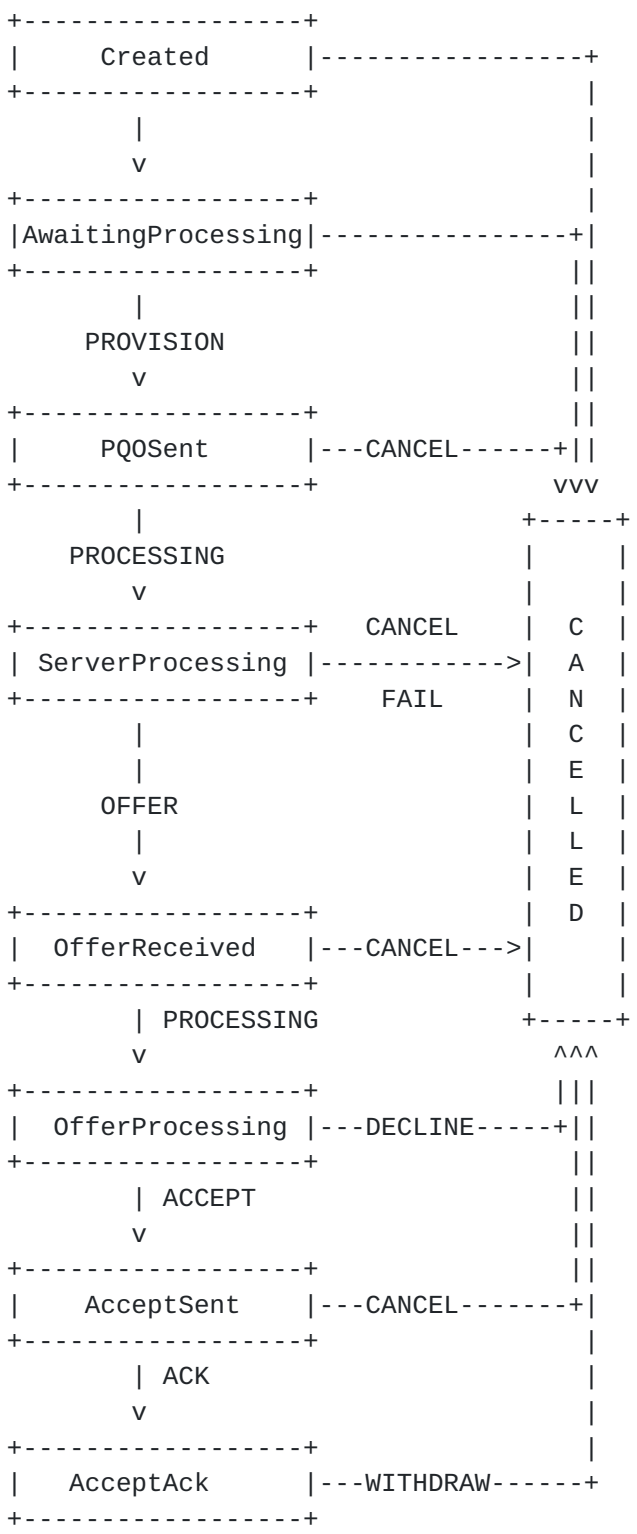

Figure 6: CPNP Finite State Machine (Client Side)

8.10.2. On the Server Side

The following lists the states which can be associated with a given order on the server's side:

- PQOReceived: when the connectivity provisioning quotation order request has been received from the client.
- AwaitingProcessing: when the order request is being processed by the server. An action from the server administrator may be needed.
- OfferProposed: when the request has been successfully handled and an offer has been sent to the requesting client.
- ProcessingReceived: when the server received a PROCESSING for an offer sent to the client.
- AcceptReceived: when the server received a confirm for the offer from the client.
- AcceptAck: when the server acknowledged the offer to the client.
- Cancelled: when the order request has failed or has been cancelled. Associate resources must be released in the latter case.
- ChildCreated: when a child PQO has been created because resources from another network provider are needed.
- ChildPQOSent: when a child PQO has been sent to another server.
- ChildServerProcessing: when a child PQO is currently processed by another server.
- ChildOfferReceived: when an offer has been received to a child PQO.
- ChildOfferProcessing: when a received offer is currently processed.
- ChildAcceptSent: when the child offer is confirmed to another server.
- ChildAcceptAck: when the child offer is acknowledged by another server.

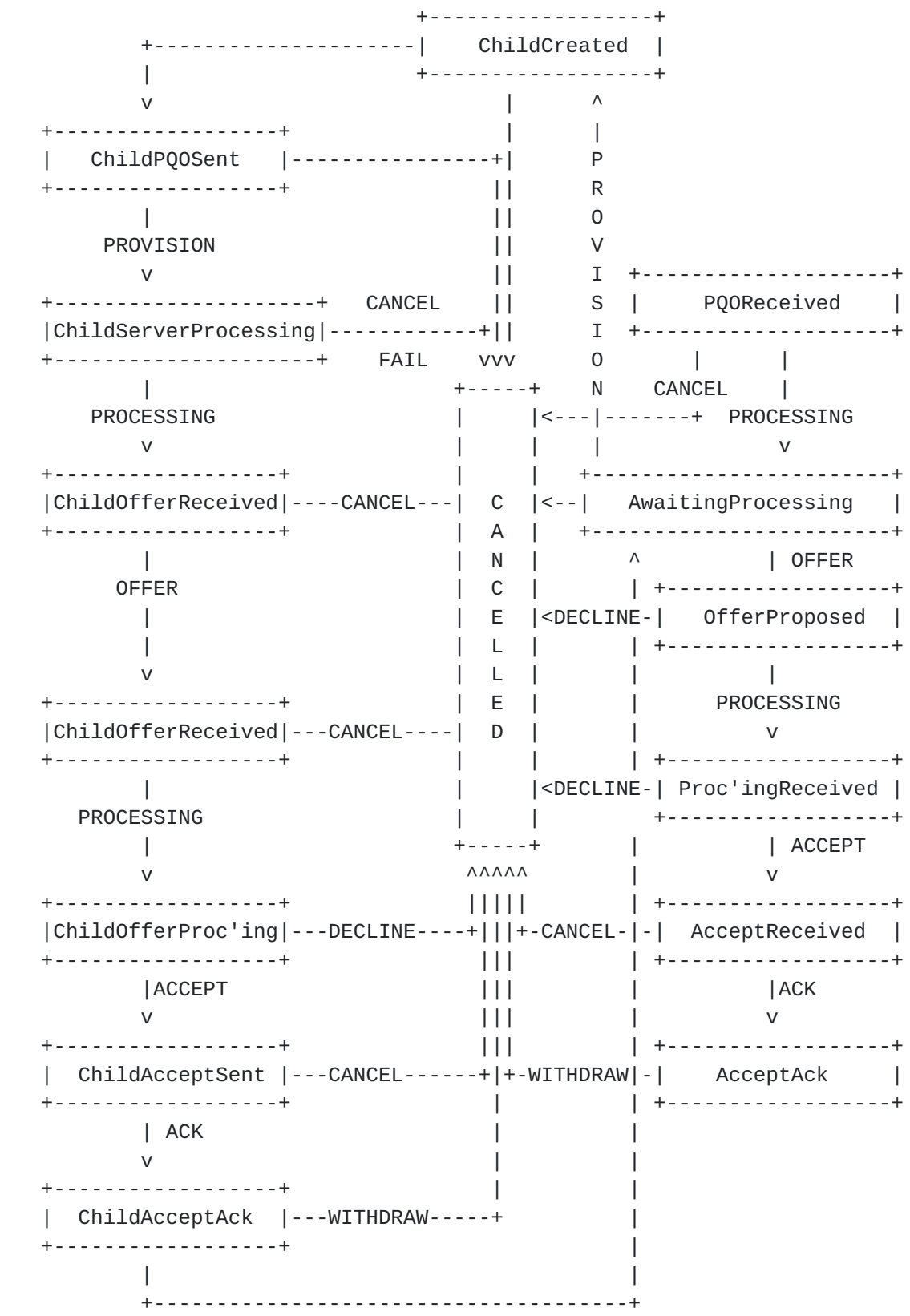

Figure 7: CPNP Finite State Machine (Server Side)

8.11. CPNP Timers

CPNP adopts a simple retransmission procedure which relies on a retransmission timer denoted as RETRANS_TIMER and maximum retry threshold. The use of RETRANS_TIMER and maximum retry threshold are described in Section 11.

The response timer (RESPONSE_TIMER) is the maximum time, defined in seconds, between a connectivity provisioning quotation order request sent to the server and the corresponding response. If this timer is exceeded, the corresponding quotation order is cancelled by the client. Relevant CANCEL messages are generated accordingly.

An offer expiration timer (EXPIRE_TIMER) represents the maximum time, defined in minutes, after which an offer made by the server will be invalid.

9. CPNP Objects

This section defines CPNP objects using the RBNF format defined at [RFC5511].

This document focuses on connectivity provisioning objects; additional Information Elements can be defined in the future.

9.1. Attributes

9.1.1. CUSTOMER_ORDER_IDENTIFIER

CUSTOMER_ORDER_IDENTIFIER is an identifier which is assigned by a client to identify an order. This identifier must be unique. Rules for assigning this identifier are specific to the client. The value of CUSTOMER_ORDER_IDENTIFIER is included in all CPNP messages.

The server handles CUSTOMER_ORDER_IDENTIFIER as an opaque value.

9.1.2. PROVIDER_ORDER_IDENTIFIER

PROVIDER_ORDER_IDENTIFIER is an identifier which is assigned by a server to identify an order. This identifier must be unique to the server. Rules for assigning this identifier are specific to the server. The value of PROVIDER_ORDER_IDENTIFIER is included in all CPNP message, except PROVISION messages.

The client handles PROVIDER_ORDER_IDENTIFIER as an opaque value.

9.1.3. TRANSACTION_ID

This object conveys the Transaction-ID introduced in Section 8.6.

9.1.4. SEQUENCE_NUMBER

Sequence Number is a number that is incremented on every new CPNP message within a CPNP transaction. This number is used to avoid reply attacks.

9.1.5. NONCE

NONCE is a random value assigned by the CPNP server. It is RECOMMENDED to assign unique NONCE values for each order.

NONCE is mandatory to be included in subsequent CPNP client operations such as: withdraw an existing order or update an order.

If the NONCE validation checks fail, the server rejects the request with a FAIL message including the appropriate failure reason code.

9.1.6. EXPECTED_RESPONSE_DATE

This attribute indicates the date by when the client is expecting to receive a response from the server. If no offer is received by then, the client will consider the quotation order as rejected.

EXPECTED_RESPONSE_DATE follows date format specified in [RFC1123].

9.1.7. EXPECTED_OFFER_DATE

This attribute indicates the date by when the server is expecting to make an offer to the client. If no offer is received by then, the client will consider the order as rejected.

EXPECTED_OFFER_DATE follows date format specified in [RFC1123].

9.1.8. VALIDITY_DATE

This attribute indicates the date of a validity of an offer made by the server. If the offer is not accepted before this date expires, the server will consider the client has rejected the offer; the server will silently clear this order.

VALIDITY_DATE follows date format specified in [RFC1123].

9.1.9. CONNECTIVITY_PROVISIONING_DOCUMENT

The RBNF format of the Connectivity Provisioning Document is shown below:

```
<CONNECTIVITY_PROVISIONING_DOCUMENT> ::=
                          <Connectivity Provisioning Component> ...
<Connectivity Provisioning Component> ::=
                          <CONNECTIVITY_PROVISIONING_PROFILE> ...
<CONNECTIVITY_PROVISIONING_PROFILE> ::=
                          <Customer Nodes Map>
                          <SCOPE>
                          <QoS Guarantees>
                          <Availability>
                          <CAPACITY>
                          <Traffic Isolation>
                          <Conformance Traffic>
                          <Flow Identification>
                          <Overall Traffic Guarantees>
                          <Routing and Forwarding>
                          <Activation Means>
                          <Invocation Means>
                          <Notifications>
<Customer Nodes Map> ::=  <Customer Node> ...
<Customer Node> ::=  <IDENTIFIER>
                     <LINK_IDENTIFIER>
                     <LOCALISATION>
```

9.1.10. Information Elements

An Information Element is an optional object which can be included in a CPNP message.

9.1.10.1. Customer Description

The client may include administrative information such as:

- Name
- Contact Information

The format of this Information Element is as follows:

```
<Customer Description> ::= <NAME> <Contact Information>
<Contact Information> ::=  <EMAIL_ADDRESS> [<POSTAL_ADDRESS>]
                           [<TELEPHONE_NUMBER> ...]
```

9.1.10.2. Provider Description

The server may include administrative information in an offer such as:

- Name
- AS Number
- Contact Information

The format of this Information Element is as follows:

<Provider Description> ::= <NAME><Contact Information>[<AS_NUMBER>]

9.1.10.3. Negotiation Options

The client may include some negotiation options such as:

- Cost: the client may include an empty or a preferred COST attribute to request the cost from the server. The server will provide the cost information in the response.
- Setup purpose: A client may request to setup a connectivity only for testing purposes during a limited period. The order can be extended to become permanent if the client was satisfied during the test period. This operation is achieved using UPDATE method.

Other negotiation options may be defined in the future.

The format of this Information Element is as follows:

<Negotiation Options> ::= [<COST>][<PURPOSE>]

9.2. Operation Messages

This section specifies the RBNF format of CPNP operation messages.

9.2.1. PROVISION

The format of PROVISION message is shown below:

```
<PROVISION Message> ::= <VERSION>
                        <METHOD_CODE>
                        <SEQUENCE_NUMBER>
                        <TRANSACTION_ID>
                        <CUSTOMER_ORDER_IDENTIFIER>
                        [<EXPECTED_RESPONSE_DATE>]
                        <CONNECTIVITY_PROVISIONING_DOCUMENT>
                        [<INFORMATION_ELEMENT>...]
```

A PROVISION message must include an order identifier which is generated by the client. Because several orders can be issued to several servers, the PROVISION message must also include a Transaction-ID. The message may include an EXPECTED_RESPONSE_DATE which indicates by when the client is expecting to receive an offer from the server. PROVISION message must also include a requested connectivity provisioning document.

When the client sends the PROVISION message to the server, the state of the order changes to "PQOSent".

9.2.2. PROCESSING

The format of PROCESSING Message is shown below:

```
<PROCESSING Message> ::= <VERSION>
                         <METHOD_CODE>
                         <SEQUENCE_NUMBER>
                         <TRANSACTION_ID>
                         <CUSTOMER_ORDER_IDENTIFIER>
                         <PROVIDER_ORDER_IDENTIFIER>
                         [<EXPECTED_OFFER_DATE>]
```

Upon receipt of a PROVISION message, the server proceeds with parsing rules (see Section 10). If no error is encountered, the server generates a PROCESSING response to the client to indicate the PQO has been received and it is being processed. The server must generate an order identifier which identifies the order in its local order repository. The server MUST copy the content of CUSTOMER_ORDER_IDENTIFIER and TRANSACTION_ID fields as conveyed in the PROVISION message. The server may include an EXPECTED_OFFER_DATE by when it expects to make an offer to the client.

Upon receipt of a PROCESSING message, the client verifies whether it has issued a PQO to that server and which contains the CUSTOMER_ORDER_IDENTIFIER and TRANSACTION_ID. If no such PQO is found, the PROCESSING message is silently ignored. If a PQO is found, the client may check if it accepts the EXPECTED_OFFER_DATE and then, it changes to state of the order to "ServerProcessing".

9.2.3. OFFER

The format of OFFER message is shown below:

```
<OFFER Message> ::= <VERSION>
                    <METHOD_CODE>
                    <SEQUENCE_NUMBER>
                    <TRANSACTION_ID>
```

```
                        <CUSTOMER_ORDER_IDENTIFIER>
                        <PROVIDER_ORDER_IDENTIFIER>
                        <NONCE>
                        <VALIDITY_DATE>
                        <CONNECTIVITY_PROVISIONING_DOCUMENT>
                        [<INFORMATION_ELEMENT>...]
```

The server answers with an OFFER message to a PROVISION request received from the client. The offer will be considered as rejected by the client if no confirmation (ACCEPT message sent by the client) is received by the server before the expiration of the validity date.

9.2.4. ACCEPT

The format of ACCEPT message is shown below:

```
<ACCEPT Message> ::=  <VERSION>
                      <METHOD_CODE>
                      <SEQUENCE_NUMBER>
                      <TRANSACTION_ID>
                      <CUSTOMER_ORDER_IDENTIFIER>
                      <PROVIDER_ORDER_IDENTIFIER>
                      <NONCE>
                      <CONNECTIVITY_PROVISIONING_DOCUMENT>
                      [<INFORMATION_ELEMENT>...]
```

This message is used by a client to confirm the acceptance of an offer received from a server. The fields of this message are copied from the received OFFER message.

9.2.5. ACK

The format of ACK message is shown below:

```
<ACK Message> ::=  <VERSION>
                   <METHOD_CODE>
                   <SEQUENCE_NUMBER>
                   <TRANSACTION_ID>
                   <CUSTOMER_ORDER_IDENTIFIER>
                   <PROVIDER_ORDER_IDENTIFIER>
                   <CONNECTIVITY_PROVISIONING_DOCUMENT>
                   [<INFORMATION_ELEMENT>...]
```

This message is issued by the server to close a CPNP transaction. In particular, this message is sent as a response to an ACCEPT, WITHDRAW, DECLINE, or CANCEL message. The ACK message must include the copy of the Connectivity Provisioning Document as stored by the server, in particular:

- A copy of the requested/offered CPD is included by the server if it successfully handled a CANCEL message.
- A copy of the updated CPD is included by the server if it successfully handled an UPDATE message.
- A copy of the offered CPD is included by the server if it successfully handled an ACCEPT message in the context of a PROVISION transaction.
- An empty CPD is included by the server if it successfully handled a DECLINE message.

9.2.6. DECLINE

The format of DECLINE message is shown below:

```
<DECLINE Message> ::=  <VERSION>
                       <METHOD_CODE>
                       <SEQUENCE_NUMBER>
                       <TRANSACTION_ID>
                       <CUSTOMER_ORDER_IDENTIFIER>
                       <PROVIDER_ORDER_IDENTIFIER>
```

The client can issue a DECLINE message to reject an offer. CUSTOMER_ORDER_IDENTIFIER, PROVIDER_ORDER_IDENTIFIER and TRANSACTION_ID are used by the server as keys to find the corresponding order. If an order matches, the server changes the state of this order to "Cancelled" and then returns an ACK with a copy of the requested CPD to the requesting client.

If no order is found, the server returns a FAIL message to the requesting client.

A flow example is shown in Figure 8.

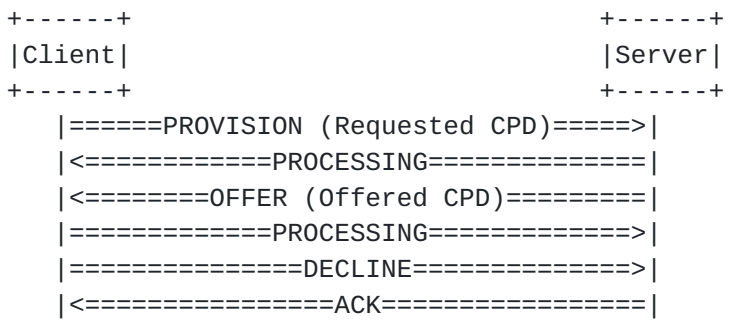

Figure 8: DECLINE Flow Example

9.2.7. CANCEL

The format of CANCEL message is shown below:

```
<CANCEL Message> ::= <VERSION>
                     <METHOD_CODE>
                     <SEQUENCE_NUMBER>
                     <TRANSACTION_ID>
                     <CUSTOMER_ORDER_IDENTIFIER>
                     [<CONNECTIVITY_PROVISIONING_DOCUMENT>]
```

The client can issue a CANCEL message at any stage during the CPNP negotiation process. CUSTOMER_ORDER_IDENTIFIER and TRANSACTION_ID are used by the server as keys to find the corresponding order. If an order matches, the server changes the state of this order to "Cancelled" and then returns an ACK with a copy of the requested CPD to the requesting client.

If no order is found, the server returns a FAIL message to the requesting client.

9.2.8. WITHDRAW

The format of WITHDRAW message is shown below:

```
<WITHDRAW Message> ::= <VERSION>
                       <METHOD_CODE>
                       <SEQUENCE_NUMBER>
                       <TRANSACTION_ID>
                       <CUSTOMER_ORDER_IDENTIFIER>
                       <PROVIDER_ORDER_IDENTIFIER>
                       <NONCE>
                       [<CONNECTIVITY_PROVISIONING_DOCUMENT>]
                       [<INFORMATION_ELEMENT>...]
```

This message is used to withdraw an offer already subscribed by the Customer. Figure 9 shows a typical usage of this message.

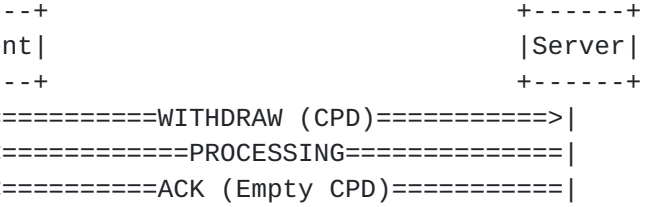

Figure 9: WITHDRAW Flow Example

The CPNP must include the same CUSTOMER_ORDER_IDENTIFIER, PROVIDER_ORDER_IDENTIFIER, and NONCE as those used when creating the order.

Upon receipt of a WITHDRAW message, the server checks whether an order matching the request is found. If an order is found, the state of the order is changed to "Cancelled" and an ACK message including an Empty CPD is returned to the requesting client. If no order is found, the server returns a FAIL message to the requesting client.

9.2.9. UPDATE

The format of UPDATE message is shown below:

```
<UPDATE Message> ::= <VERSION>
                     <METHOD_CODE>
                     <SEQUENCE_NUMBER>
                     <TRANSACTION_ID>
                     <CUSTOMER_ORDER_IDENTIFIER>
                     <PROVIDER_ORDER_IDENTIFIER>
                     <NONCE>
                     <EXPECTED_RESPONSE_DATE>
                     <CONNECTIVITY_PROVISIONING_DOCUMENT>
                     [<INFORMATION_ELEMENT>...]
```

This message is sent by the CPNP client to update an existing connectivity provisioning agreement. The CPNP must include the same CUSTOMER_ORDER_IDENTIFIER, PROVIDER_ORDER_IDENTIFIER, and NONCE as those used when creating the order. The CPNP client includes a new CPD which integrates the requested modifications. A new Transaction_ID must be assigned by the client.

Upon receipt of an UPDATE message, the server checks whether an order, having state "Completed", matches CUSTOMER_ORDER_IDENTIFIER, PROVIDER_ORDER_IDENTIFIER, and NONCE.

- If no order is found, the CPNP server generates a FAIL error with the appropriate error code.
- If an order is found, the server checks whether it can honor the request:
  - A FAIL message is sent to the client if the server cannot honor the request. The client may initiate a new PQO negotiation cycle.
  - An OFFER message including the updated connectivity provisioning document is sent to the client. For example, the server maintains an order for provisioning a VPN service that connects sites A, B and C. If the client sends an UPDATE

message to remove site C, only sites A and B will be included in the OFFER sent by the server to the requesting client.

A flow chart that illustrates the use of UPDATE operation is shown in Figure 10.

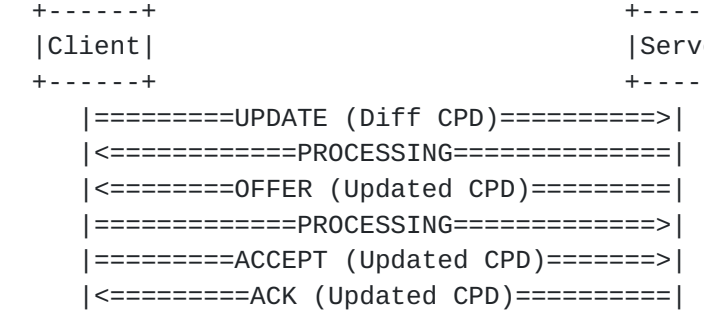

Figure 10: UPDATE Flow Example

9.2.10. FAIL

The format of FAIL message is shown below:

```
<FAIL Message> ::= <VERSION>
                   <METHOD_CODE>
                   <SEQUENCE_NUMBER>
                   <TRANSACTION_ID>
                   <CUSTOMER_ORDER_IDENTIFIER>
                   <PROVIDER_ORDER_IDENTIFIER>
                   <STATUS_CODE>
```

This message is sent in the following cases:

- The server can not honor an order received from the client (i.e., received in a PROVISION or UPDATE request).
- The server encounters an error when processing a CPNP request received from the client.

The status code indicates the error code. The following codes are currently supported; other codes will be defined in future versions of the document:

1 (Validation Error): The message can not be validated (see Section 10).
2 (Authentication Required): the request cannot be handled because authentication is required.
3 (Authorization Required): : the request cannot be handled because authorization failed.

4 (Administratively prohibited): the request can not be handled because of administrative policies.
5 (Out of Resources): the request can not be honored because there is not enough capacity.
6 (Network Presence): the request can not be honored because there is no network presence.

10. Message Validation

Both client and server proceed with CPNP message validation. The following tables summarize the validation checks to be followed.

10.1. On the Client Side

Operation	Validation Checks
PROCESSING	{Source IP address, source port, destination IP address, destination port, Transaction-ID, Customer Order Identifier} must match an existing PQO with a state set to "PQOSent". The sequence number carried in the packet must match the sequence number maintained by the client.
OFFER	{Source IP address, source port, destination IP address, destination port, Transaction-ID, Customer Order Identifier} must match an existing order with state set to "PQOSent" or {Source IP address, source port, destination IP address, destination port, Transaction-ID, Customer Order Identifier, Provider Order Identifier} must match an existing order with a state set to "ServerProcessing". The sequence number carried in the packet must match the sequence number maintained by the client.
ACK (PROVISION Transaction)	{Source IP address, source port, destination IP address, destination port, Transaction-ID, Customer Order Identifier, Provider Order Identifier, Offered Connectivity Provisioning Order} must match an order with a state set to "AcceptSent". The sequence number carried in the packet must match the sequence number maintained by the client.
ACK (UPDATE Transaction)	{Source IP address, source port, destination IP address, destination port, Transaction-ID, Customer Order Identifier, Provider Order Identifier, Updated Connectivity Provisioning Order} must match an order with a state set to "AcceptSent". The sequence number carried in the packet must match the sequence number maintained by the client.
ACK (WITHDRAW Transaction)	{Source IP address, source port, destination IP address, destination port, Transaction-ID, Customer Order Identifier, Provider Order Identifier, Empty Connectivity Provisioning Order} must match an order with a state set to "Cancelled". The sequence number carried in the packet must match the sequence number maintained by the client.

10.2. On the Server Side

Method	Validation Checks
PROVISION	The source IP address passes existing access filters (if any).
ACCEPT	{Source IP address, source port, destination IP address, destination port, Transaction-ID, Customer Order Identifier, Provider Order Identifier, Nonce, Offered Connectivity Provisioning Order} must match an order with state set to "OfferProposed" or "ProcessngReceived". The sequence number carried in the packet must match the sequence number maintained by the server.
DECLINE	{Source IP address, source port, destination IP address, destination port, Transaction-ID, Customer Order Identifier, Provider Order Identifier, Nonce} must match an order with state set to "OfferProposed" or "ProcessngReceived". The sequence number carried in the packet must match the sequence number maintained by the server.
UPDATE	The source IP address passes existing access filters (if any) and {Customer Order Identifier, Provider Order Identifier, Nonce} must match an existing order with state "Completed".
WITHDRAW	The source IP address passes existing access filters (if any) and {Customer Order Identifier, Provider Order Identifier, Nonce} must match an existing order with state "Completed".

11. Theory of Operation

Both CPNP client and server proceeds to message validation checks as specified in Section 10.

11.1. Client Behavior

11.1.1. Order Negotiation Cycle

To place a connectivity provisioning quotation order, the client initiates first a local order object identified by a unique identifier assigned by the client. The state of the quotation order is set to "Created". The client then generates a PROVISION request which includes the assigned identifier, possibly an Expected Response Date, a Transaction-ID and a Requested Connectivity Provisioning Document. The client may include additional Information Elements such as Negotiation Options.

The client may be configured to not enforce negotiation checks on EXPECTED_OFFER_DATE; if so no EXPECTED_RESP_TIMER attribute (or

EXPECTED_RESP_TIMER set to infinite) should be included in the quotation order.

Once the request is sent to the server, the state of the request is set to "PQOSent" and a timer, if a response time is included in the quotation order, is set to the expiration date as included in the PROVISION request. The client also maintains a copy of the extended transport session details used to generate the PROVISION request. The CPNP client must listen on the same port number that it used to send the PROVISION request.

If no answer is received from the server before the retransmission timer expires (i.e., RETRANS_TIMER), the client proceeds to retransmission until maximum retry is reached (i.e., 3 times). The same sequence number is used for retransmitted packets.

If a FAIL message is received, the client may decide to issue another (corrected) request towards the same server, cancel the local order, or contact another server. The behavior of the client depends on the error code returned by the server in the FAIL message.

If a PROCESSING message matching the CPNP transport session is received, the client updates the CPNP session with the PROVIDER_ORDER_IDENTIFIER information. If the client does not accept the expected offer date that may have been indicated in the PROCESSING message, the client may decide to cancel the quotation order. If the client accepts the EXPECTED_OFFER_DATE, it changes the state of the order to "ServerProcessing" and sets a timer to the value of EXPECTED_OFFER_DATE. If no offer is made before the timer expires, the client changes the state of the order to "Cancelled".

If an OFFER message matching the extended CPNP session is received, the client checks if a PROCESSING message having the same PROVIDER_ORDER_IDENTIFIER has been received from the server. If a PROCESSING message was already received for the same order but the PROVIDER_ORDER_IDENTIFIER does not match the identifier included in the OFFER message, the client ignores silently the message. If a PROCESSING message having the same PROVIDER_ORDER_IDENTIFIER was already received and matches the CPNP transaction identifier, the client changes the state of the order to "OfferReceived" and sets a timer to the value of VALIDITY_DATE indicated in the OFFER message.

If an offer is received from the server (i.e., as documented in an OFFER message), the client may accept or reject the offer. The client accepts the offer by generating an ACCEPT message which confirms that the client agrees to subscribe to the offer documented in the OFFER message; the state of the order is passed to "AcceptSent". The transaction is terminated if an ACK message is

received from the server. If no ACK is received from the server, the client proceeds with the re-transmission of the ACCEPT message.

The client may also decide to reject the offer by sending a DECLINE message. The state of the order is set by the client to "Cancelled". If an offer is not acceptable by the client, the client may decide to contact a new server or submit another order to the same server. Guidelines to issue an updated order or terminate the negotiation are specific to the client.

11.1.2. Order Withdrawal Cycle

A client may withdraw a completed order. This is achieved by issuing a WITHDRAW message. This message must include Customer Order Identifier, Provider Identifier and Nonce returned during the order negotiation cycle specified in Section 11.1.1.

If no ACK is received from the server, the client proceeds with the re-transmission of the message.

11.1.3. Order Update Cycle

A client may update a completed order. This is achieved by issuing an UPDATE message. This message must include Customer Order Identifier, Provider Order Identifier and Nonce returned during the order negotiation cycle specified in Section 11.1.1. The client must include in the UPDATE message an updated CPD with the requested changes.

Subsequent messages exchange is similar to what is documented in Section 11.1.1.

11.2. Server Behavior

11.2.1. Order Processing

Upon receipt of a PROVISION message from a client, the server sets a CPNP session, stores Transaction-ID and generates a Provider Order Identifier. Once preliminary validation checks are completed (Section 10), the server may return a PROCESSING message to notify the client the quotation order is received and it is under processing; the server may include an expected offer date to notify the client by when an offer will be proposed. An order with state "AwaitingProcessing" is created by the server. The server runs its decision-making process to decide which offer it can make to honor the received order. The offer should be made before the expected offer date expires.

If the server cannot honor the request, it sends backs a FAIL message with the appropriate error code.

If the server can honor the request, it creates an OFFER message. The server must indicate the Transaction-ID, Customer Order Identifier as indicated in the PROVISION message, and the Provider Order Identifier generated for this order. The server must also include Nonce and the offered Connectivity Provisioning Document. The server includes an offer validity date as well. Once sent to the client, the server changes the state of the order to "OfferSent" and a timer set to the validity date is initiated.

If the server determines that additional network resources from another network provider are needed to accommodate a quotation order, it will create child PQO(s) and will behave as a CPNP client to negotiate child PQO(s) with possible partnering providers (see Figure 5).

If no PROCESSING, ACCEPT or DECLINE message is received before the expiry of the RETRANS_TIMER, the server re-sends the same offer to the client. This procedure is repeated until maximum retry is reached.

If an ACCEPT message is received before the offered validity date expires, the server proceeds with validation checks as specified in Section 10. The state of the corresponding order is passed to "AcceptReceived". The server sends back an ACK message to terminate the order processing cycle.

If a CANCEL/DECLINE message is received, the server proceeds with the cancellation of the order. The state of the order is then passed to "Cancelled".

11.2.2. Order Withdrawal

A client may withdraw a completed order by issuing a WITHDRAW message. Upon receipt of a WITHDRAW message, the server proceeds with the validation checks, as specified in Section 10.

- If the checks fail, a FAIL message is sent back to the client with the appropriate error code.
- If the checks succeed, the server clears the clauses of the Connectivity Provisioning Document, changes the state of the order to "Cancelled", and sends back an ACK message with an Empty Connectivity Provisioning Document.

11.2.3. Order Update

A client may update an order by issuing an UPDATE message. Upon receipt of an UPDATE message, the server proceeds with the validation checks as specified in Section 10.

- If the checks fail, a FAIL message is sent back to the client with the appropriate error code.
- Subsequent messages exchange is similar to what is specified in Section 11.1.1. The server should generate a new Nonce value to be included in the offer made to the client.

12. Operational Guidelines

12.1. Logging on the CPNP Server

The CPNP server SHOULD be configurable to log various events and associated information. Such information includes:

- Client's IP Address
- Any event change (e.g., new quotation order, offer sent, order confirm, order cancellation, order withdraw, etc.)
- Timestamp

12.2. Business Guidelines & Objectives

The CPNP server can operate in the following modes:

1. Fully automated mode: The CPNP server is provisioned with a set of business guidelines and objectives that will be used as an input to the decision-making process. The CPNP server will service received orders that falls into these business guidelines; otherwise requests will be escalated to an administrator that will formally validate/invalidate an order request. The set of policies to be configured to the CPNP server are specific to each administrative entity managing a CPNP server.
2. Administrative-based mode: This mode assumes some or all CPNP server' operations are subject to a formal administrative validation. CPNP events will trigger appropriate validation requests that will be forwarded to the contact person(s) or department which is responsible for validating the orders. Administrative validation messages are relayed using another protocol (e.g., SMTP) or a dedicated tool.

Business guidelines are local to each administrative entity. How validation requests are presented to an administrator are out of

scope of this document; each administrative entity may decide the appropriate mechanism to enable for that purpose.

13. Security Considerations

Means to defend the server against denial-of-service attacks must be enabled. For example, access control lists (ACLs) can be enforced on the client, the server or the network in between, to allow a trusted client to communicate with a trusted server.

The client and the server should be mutually authenticated. Out of band mechanisms can be used instead of integrating them into CPNP.

The client must silently discard CPNP responses received from unknown CPNP servers. The use of a randomly generated Transaction-ID makes it hard to forge a response from a server with a spoofed IP address belonging the legitimate CPNP server. Furthermore, CPNP messages from the server must also include correct identifiers of the orders. Two order identifiers are used: one generated by the client and the second one is generated by the server.

Only authorized clients must be able to modify existing CPNP orders. The use of a randomly generated Nonce by the server makes it hard to modify an order on behalf of a third-party.

The use of sequence numbers protects the CPNP server from replay attacks.

The Provider must enforce means to protect privacy-related information captured in a CPP documents exchanged using CPNP messages [RFC6462]. In particular, this information must not be revealed to external parties without the consent of Customers. Providers should enforce policies to make Customer fingerprinting more difficult to achieve. For more discussion about privacy, refer to [RFC6462][RFC6973].

14. IANA Considerations

Authors of the document request IANA to assign a UDP port for CPNP.

A registry for CPNP methods should be created. The following codes are reserved:

1: PROVISION
2: PROCESSING
3: OFFER
4: ACCEPT
5: DECLINE

6: ACK
7: CANCEL
8: WITHDRAW
9: UPDATE
10: FAIL

A registry for CPNP errors should be created. The following codes are reserved:

1: Message Validation Error
2: Authentication Required
3: Authorization Failed
4: Administratively prohibited
5: Out of Resources
6: Network Presence Error

15. Acknowledgements

Thanks to Diego R. Lopez for his comments.

16. References

16.1. Normative References

[RFC1123] Braden, R., "Requirements for Internet Hosts - Application and Support", STD 3, RFC 1123, October 1989.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC4086] Eastlake, D., Schiller, J., and S. Crocker, "Randomness Requirements for Security", BCP 106, RFC 4086, June 2005.

[RFC5511] Farrel, A., "Routing Backus-Naur Form (RBNF): A Syntax Used to Form Encoding Rules in Various Routing Protocol Specifications", RFC 5511, April 2009.

16.2. Informative References

[I-D.boucadair-connectivity-provisioning-profile] Boucadair, M., Jacquenet, C., and N. Wang, "IP Connectivity Provisioning Profile (CPP)", draft-boucadair-connectivity-provisioning-profile-05 (work in progress), April 2014.

[RFC2782] Gulbrandsen, A., Vixie, P., and L. Esibov, "A DNS RR for specifying the location of services (DNS SRV)", RFC 2782, February 2000.

[RFC4176] El Mghazli, Y., Nadeau, T., Boucadair, M., Chan, K., and A. Gonguet, "Framework for Layer 3 Virtual Private Networks (L3VPN) Operations and Management", RFC 4176, October 2005.

[RFC6347] Rescorla, E. and N. Modadugu, "Datagram Transport Layer Security Version 1.2", RFC 6347, January 2012.

[RFC6462] Cooper, A., "Report from the Internet Privacy Workshop", RFC 6462, January 2012.

[RFC6770] Bertrand, G., Stephan, E., Burbridge, T., Eardley, P., Ma, K., and G. Watson, "Use Cases for Content Delivery Network Interconnection", RFC 6770, November 2012.

[RFC6973] Cooper, A., Tschofenig, H., Aboba, B., Peterson, J., Morris, J., Hansen, M., and R. Smith, "Privacy Considerations for Internet Protocols", RFC 6973, July 2013.

[RFC7149] Boucadair, M. and C. Jacquenet, "Software-Defined Networking: A Perspective from within a Service Provider Environment", RFC 7149, March 2014.

Authors' Addresses

Mohamed Boucadair
France Telecom
Rennes 35000
France

Email: mohamed.boucadair@orange.com

Christian Jacquenet
France Telecom
Rennes 35000
France

Email: christian.jacquenet@orange.com