



UNIÓN INTERNACIONAL DE TELECOMUNICACIONES

**UIT-T**

SECTOR DE NORMALIZACIÓN  
DE LAS TELECOMUNICACIONES  
DE LA UIT

**Enmienda 1**  
**X.880**  
(11/95)

**REDES DE DATOS Y COMUNICACIÓN  
ENTRE SISTEMAS ABIERTOS**

**APLICACIONES DE INTERCONEXIÓN  
DE SISTEMAS ABIERTOS –  
OPERACIONES A DISTANCIA**

---

**TECNOLOGÍA DE LA INFORMACIÓN –  
OPERACIONES A DISTANCIA:  
CONCEPTOS, MODELO Y NOTACIÓN**

**ENMIENDA 1: OPERACIONES INCORPORADAS**

**Enmienda 1 a la  
Recomendación UIT-T X.880**

(Anteriormente «Recomendación del CCITT»)

---

## PREFACIO

La UIT (Unión Internacional de Telecomunicaciones) es el organismo especializado de las Naciones Unidas en el campo de las telecomunicaciones. El UIT-T (Sector de Normalización de las Telecomunicaciones de la UIT) es un órgano permanente de la UIT. En el UIT-T, que es la entidad que establece normas mundiales (Recomendaciones) sobre las telecomunicaciones, participan unos 179 países miembros, 84 empresas de explotación de telecomunicaciones, 145 organizaciones científicas e industriales y 38 organizaciones internacionales.

Las Recomendaciones las aprueban los Miembros del UIT-T de acuerdo con el procedimiento establecido en la Resolución N.º 1 de la CMNT (Helsinki, 1993). Adicionalmente, la Conferencia Mundial de Normalización de las Telecomunicaciones (CMNT), que se celebra cada cuatro años, aprueba las Recomendaciones que para ello se le sometan y establece el programa de estudios para el periodo siguiente.

En ciertos sectores de la tecnología de la información que corresponden a la esfera de competencia del UIT-T, se preparan las normas necesarias en colaboración con la ISO y la CEI. El texto de la Recomendación UIT-T X.880, Enmienda 1, se aprobó el 21 de noviembre de 1995. Su texto se publica también, en forma idéntica, como Norma Internacional ISO/CEI 13712-1.

---

### NOTA

En esta Recomendación, la expresión «Administración» se utiliza para designar, en forma abreviada, tanto una administración de telecomunicaciones como una empresa de explotación reconocida de telecomunicaciones.

© UIT 1996

Es propiedad. Ninguna parte de esta publicación puede reproducirse o utilizarse, de ninguna forma o por ningún medio, sea éste electrónico o mecánico, de fotocopia o de microfilm, sin previa autorización escrita por parte de la UIT.

RECOMENDACIONES UIT-T DE LA SERIE X

**REDES DE DATOS Y COMUNICACIÓN ENTRE SISTEMAS ABIERTOS**

(Febrero de 1994)

**ORGANIZACIÓN DE LAS RECOMENDACIONES DE LA SERIE X**

| Dominio  | Recomendaciones |
|--|-----------------|
| <b>REDES PÚBLICAS DE DATOS</b>   |                 |
| Servicios y facilidades  | X.1-X.19        |
| Interfaces   | X.20-X.49       |
| Transmisión, señalización y conmutación  | X.50-X.89       |
| Aspectos de redes  | X.90-X.149      |
| Mantenimiento  | X.150-X.179     |
| Disposiciones administrativas  | X.180-X.199     |
| <b>INTERCONEXIÓN DE SISTEMAS ABIERTOS</b>  |                 |
| Modelo y notación  | X.200-X.209     |
| Definiciones de los servicios  | X.210-X.219     |
| Especificaciones de los protocolos en modo conexión                                  | X.220-X.229     |
| Especificaciones de los protocolos en modo sin conexión                              | X.230-X.239     |
| Formularios para enunciados de conformidad de implementación de protocolo            | X.240-X.259     |
| Identificación de protocolos   | X.260-X.269     |
| Protocolos de seguridad  | X.270-X.279     |
| Objetos gestionados de capa  | X.280-X.289     |
| Pruebas de conformidad   | X.290-X.299     |
| <b>INTERFUNCIONAMIENTO ENTRE REDES</b>   |                 |
| Generalidades  | X.300-X.349     |
| Sistemas móviles de transmisión de datos   | X.350-X.369     |
| Gestión  | X.370-X.399     |
| <b>SISTEMAS DE TRATAMIENTO DE MENSAJES</b>   | X.400-X.499     |
| <b>DIRECTORIO</b>  | X.500-X.599     |
| <b>GESTIÓN DE REDES DE INTERCONEXIÓN DE SISTEMAS ABIERTOS Y ASPECTOS DE SISTEMAS</b> |                 |
| Gestión de redes   | X.600-X.649     |
| Denominación, direccionamiento y registro  | X.650-X.679     |
| Notación de sintaxis abstracta uno   | X.680-X.699     |
| <b>GESTIÓN DE INTERCONEXIÓN DE SISTEMAS ABIERTOS</b>                                 | X.700-X.799     |
| <b>SEGURIDAD</b>   | X.800-X.849     |
| <b>APLICACIONES DE INTERCONEXIÓN DE SISTEMAS ABIERTOS</b>                            |                 |
| Cometimiento, concurrencia y recuperación  | X.850-X.859     |
| Tratamiento de transacciones   | X.860-X.879     |
| Operaciones a distancia  | X.880-X.899     |
| <b>TRATAMIENTO ABIERTO DISTRIBUIDO</b>   | X.900-X.999     |



## ÍNDICE

|                                    | <i>Página</i> |
|------------------------------------|---------------|
| 1) Subcláusula 3.3 .....           | 1             |
| 2) Subcláusula 8.2.1 .....         | 1             |
| 3) Subcláusula 8.2 .....           | 2             |
| 4) Subcláusula 10.1 .....          | 2             |
| 5) Subcláusula 10.5.1 .....        | 2             |
| 6) Subcláusula 10.5.2 .....        | 2             |
| 7) Subcláusulas 10.6 a 10.16 ..... | 2             |
| 8) Subcláusulas 10.6 a 10.11 ..... | 2             |
| 9) Anexo A .....                   | 4             |
| 10) Anexo D .....                  | 6             |

## **Resumen**

Esta enmienda a la Recomendación X.880 | ISO 13712-1 contiene la definición de tres operaciones incorporadas: sonda, acuse de recibo y cancelación, que son de utilidad general a los diseñadores de aplicaciones basadas en el elemento de servicio de operaciones a distancia.

NORMA INTERNACIONAL

RECOMENDACIÓN UIT-T

## TECNOLOGÍA DE LA INFORMACIÓN – OPERACIONES A DISTANCIA: CONCEPTOS, MODELO Y NOTACIÓN

### ENMIENDA 1 Operaciones incorporadas

#### 1) Subcláusula 3.3

Añadir la nueva definición siguiente después de 3.3.7:

«**3.3.8 equipotente**: Característica de una operación que puede ser invocada repetidamente sin cambiar el estado del realizador.»

Las definiciones que siguen a la nueva definición 3.3.8 se han de reenumerar consecuentemente.

#### 2) Subcláusula 8.2.1

Añadir el siguiente campo (que se indica subrayado) a la clase de objeto de información OPERATION:

|                            |   |  |
|----------------------------|---|--|
| <b>OPERATION ::= CLASS</b> |   |  |
| {                          | <b>&amp;ArgumentType</b> <b>OPTIONAL,</b><br><b>&amp;argumentTypeOptional</b> <b>BOOLEAN OPTIONAL,</b><br><b>&amp;returnResult</b> <b>BOOLEAN DEFAULT TRUE,</b><br><b>&amp;ResultType</b> <b>OPTIONAL,</b><br><b>&amp;resultTypeOptional</b> <b>BOOLEAN OPTIONAL,</b><br><b>&amp;Errors</b> <b>ERROR OPTIONAL,</b><br><b>&amp;Linked</b> <b>OPERATION OPTIONAL,</b><br><b>&amp;synchronous</b> <b>BOOLEAN DEFAULT FALSE,</b><br><u><b>&amp;idempotent</b></u> <u><b>BOOLEAN DEFAULT FALSE,</b></u><br><b>&amp;alwaysReturns</b> <b>BOOLEAN DEFAULT TRUE,</b><br><b>&amp;InvokePriority</b> <b>Priority OPTIONAL,</b><br><b>&amp;ResultPriority</b> <b>Priority OPTIONAL,</b><br><b>&amp;operationCode</b> <b>Code UNIQUE OPTIONAL</b> |  |
| }                          |   |  |
| <b>WITH SYNTAX</b>         |   |  |
| {                          | <b>[ARGUMENT</b> <b>&amp;ArgumentType [OPTIONAL</b> <b>&amp;argumentTypeOptional]]</b><br><b>[RETURN RESULT</b> <b>&amp;returnResult]</b><br><b>[RESULT</b> <b>&amp;ResultType [OPTIONAL</b> <b>&amp;resultTypeOptional]]</b><br><b>[ERRORS</b> <b>&amp;Errors]</b><br><b>[LINKED</b> <b>&amp;Linked]</b><br><b>[SYNCHRONOUS</b> <b>&amp;synchronous]</b><br><u><b>[IDEMPOTENT</b></u> <u><b>&amp;idempotent]</b></u><br><b>[ALWAYS RESPONDS</b> <b>&amp;alwaysReturns]</b><br><b>[INVOKE PRIORITY</b> <b>&amp;InvokePriority]</b><br><b>[RESULT-PRIORITY</b> <b>&amp;ResultPriority]</b><br><b>[CODE</b> <b>&amp;operationCode]</b>  |  |
| }                          |   |  |

### 3) Subcláusula 8.2

Añadir la nueva subcláusula siguiente:

«8.2.14 El campo `&idempotent` (equipotente) especifica si la operación es equipotente o no, y toma el valor TRUE si lo es, y FALSE en los demás casos.»

### 4) Subcláusula 10.1

Escribir de nuevo el ítem a) como sigue, (con el nuevo texto que se indica subrayado):

«a) operaciones generalmente útiles (`emptyBind`, `emptyUnbind`, `no-op`, `probe`, `acknowledge`, `cancel`) y sus errores asociados;»

### 5) Subcláusula 10.5.1

Escribir de nuevo la definición de la `no-op OPERATION` añadiendo el siguiente campo adicional (que se indica subrayado):

```
no-op OPERATION ::=
{
    IDEMPOTENT           TRUE
    ALWAYS RESPONDS    FALSE
    CODE                local:-1
}
```

### 6) Subcláusula 10.5.2

Escribir de nuevo 10.5.2 tal como sigue, (con el nuevo texto que se indica subrayado):

«10.5.2 Esta operación es equipotente y no retorna resultado.»

### 7) Subcláusulas 10.6 a 10.16

Renumerarlas como 10.12 a 10.22 respectivamente.

### 8) Subcláusulas 10.6 a 10.11

Añadir las nuevas subcláusulas siguientes numeradas como subcláusulas 10.6 a 10.11:

#### 10.6 Sondeo

10.6.1 La operación `probe` (sondeo) investiga sobre el resultado de una operación invocada anteriormente. Se especifica como sigue:

```
probe OPERATION ::=
{
    ARGUMENT    SEQUENCE
    {
        invokeId    [0] InvokeId
    }
    RESULT      ENUMERATED{running(0), finished(1), unknown(2), ...}
    IDEMPOTENT  TRUE
    CODE        local:-2
}
```

10.6.2 Hay un solo argumento, del tipo `InvokeId` (Id de invocación), que identifica la operación invocada que se está investigando.

10.6.3 La petición siempre retorna un resultado que indica si la invocación de la operación está todavía en curso (`running`), su realización ha terminado (`finished`) o se trata de una invocación desconocida (`unknown`).



NOTA – Una invocación puede ser desconocida porque nunca se ha producido o porque ha sido olvidada por el realizador.

**10.6.4** La operación es equipotente.

**10.6.5** Un sondeo (con resultado de terminada) provoca, como efecto secundario, la retransmisión de cualquier retorno de la invocación de que se trata, salvo si la operación es equipotente.

NOTA – Esto implica que el realizador de una operación no equipotente ha de retener la respuesta (resultado o error) si la operación sondeo se ha incluido en el lote de operaciones.

## 10.7 Acuse

**10.7.1** La operación `acknowledge` (acuse) acusa recibo del retorno de alguna invocación (no equipotente). Se especifica como sigue:

```

acknowledge OPERATION ::=
{
    ARGUMENT    InvokeId
    RESULT      ENUMERATED{acknowledged(0), unknown(1), ...}
    IDEMPOTENT  TRUE
    CODE        local:-3
}

```

**10.7.2** Hay un solo argumento, del tipo `InvokeId` (Id de invocación), que identifica la invocación de cuyo retorno se está acusando recibo.

**10.7.3** La petición siempre retorna un resultado que indica que el retorno se considera ahora acusado (`acknowledged`) o que la invocación de operación de que se trata es desconocida (`unknown`).

NOTA – Una invocación puede ser desconocida porque nunca se ha producido o porque ha sido olvidada por el realizador.

**10.7.4** La operación es equipotente.

**10.7.5** Esta operación debe incluirse en todo lote de operaciones que incluya la operación sondeo.

## 10.8 Sondeo y accuse

**10.8.1** El conjunto de operaciones `ProbeAndAcknowledge` (sondeo y accuse) comprende las dos operaciones que indica su nombre, y con frecuencia se necesitarán ambas en un lote. Se especifica como sigue:

```

ProbeAndAcknowledge OPERATION ::= {probe | acknowledge}

```

## 10.9 Cancelación

**10.9.1** La operación `cancel` (cancelación) pide la terminación prematura de la realización de una operación. Sólo las operaciones que incluyan el error `cancelled` (cancelada) (véase 10.11) en su campo `&Errors` (errores) pueden ser canceladas. Se especifica como sigue:

```

cancel OPERATION ::=
{
    ARGUMENT    InvokeId
    ERRORS      {cancelFailed}
    IDEMPOTENT  TRUE
    CODE        local:-4
}

```

**10.9.2** Hay un solo argumento, del tipo `InvokeId` (Id de invocación), que identifica la operación invocada que está siendo cancelada.

**10.9.3** Si falla la petición se retorna un error `cancelFailed` (cancelación fallida) (véase 10.10).

**10.9.4** La operación es equipotente.

## 10.10 Cancelación fallida

**10.10.1** Un error `cancelFailed` (cancelación fallida) señala un problema en la realización de una cancelación. Se especifica como sigue:

```
cancelFailed ERROR ::=
{
  PARAMETER          SET
  {
    problem          [0] CancelProblem,
    operation        [1] InvokeId
  }
  CODE               local:-2
}

CancelProblem ::= ENUMERATED
{unknownOperation(0), tooLate(1), operationNotCancellable(2), ...}
```

**10.10.2** Los distintos parámetros tienen el significado que se define en 10.10.2.1 y 10.10.2.2.

**10.10.2.1** El problema (`problem`) específico encontrado en la cancelación se indica entre las siguientes posibilidades:

- a) `unknownOperation` (operación desconocida) – Esta invocación de operación no se ha producido o ha sido olvidada.
- b) `tooLate` (demasiado tarde) – Esta operación ya ha sido realizada o su ejecución se halla en una etapa que no permite una cancelación.
- c) `operationNotCancellable` (operación no cancelable) – La operación invocada no era una que pudiera ser cancelada.

**10.10.2.2** La identificación de la operación (`operation`) (invocación) que había de ser cancelada.

## 10.11 Cancelada

El error `cancelled` (cancelada) se notifica si se cancela alguna operación. Este error debe incluirse en el campo `&Errors` (errores) de la operación afectada. Se especifica como sigue:

```
cancelled ERROR ::= {CODE local:-3}
```

## 9) Anexo A

*Cambiar la referencia del primer módulo como sigue, (con la modificación que se indica subrayada):*

**Remote-Operations-Information-Objects {joint-iso-itu-t remote-operations(4) informationObjects(5) version2(1)}**

Añadir el siguiente campo (que se indica subrayado) a la clase de objeto de información OPERATION:

|                            |                                  |   |
|----------------------------|----------------------------------|---|
| <b>OPERATION ::= CLASS</b> |                                  |   |
| {                          |                                  |   |
|                            | <b>&amp;ArgumentType</b>         | <b>OPTIONAL,</b>  |
|                            | <b>&amp;argumentTypeOptional</b> | <b>BOOLEAN OPTIONAL,</b>  |
|                            | <b>&amp;returnResult</b>         | <b>BOOLEAN DEFAULT TRUE,</b>  |
|                            | <b>&amp;ResultType</b>           | <b>OPTIONAL,</b>  |
|                            | <b>&amp;resultTypeOptional</b>   | <b>BOOLEAN OPTIONAL,</b>  |
|                            | <b>&amp;Errors</b>               | <b>ERROR OPTIONAL,</b>  |
|                            | <b>&amp;Linked</b>               | <b>OPERATION OPTIONAL,</b>  |
|                            | <b>&amp;synchronous</b>          | <b>BOOLEAN DEFAULT FALSE,</b>                                       |
|                            | <b><u>&amp;idempotent</u></b>    | <b><u>BOOLEAN DEFAULT FALSE,</u></b>                                |
|                            | <b>&amp;alwaysReturns</b>        | <b>BOOLEAN DEFAULT TRUE,</b>  |
|                            | <b>&amp;InvokePriority</b>       | <b>Priority OPTIONAL,</b>   |
|                            | <b>&amp;ResultPriority</b>       | <b>Priority OPTIONAL,</b>   |
|                            | <b>&amp;operationCode</b>        | <b>Code UNIQUE OPTIONAL</b>   |
| }                          |                                  |   |
| <b>WITH SYNTAX</b>         |                                  |   |
| {                          |                                  |   |
|                            | <b>[ARGUMENT</b>                 | <b>&amp;ArgumentType [OPTIONAL      &amp;argumentTypeOptional]]</b> |
|                            | <b>[RETURN RESULT</b>            | <b>&amp;returnResult]</b>   |
|                            | <b>[RESULT</b>                   | <b>&amp;ResultType [OPTIONAL      &amp;resultTypeOptional]]</b>     |
|                            | <b>[ERRORS</b>                   | <b>&amp;Errors]</b>   |
|                            | <b>[LINKED</b>                   | <b>&amp;Linked]</b>   |
|                            | <b>[SYNCHRONOUS</b>              | <b>&amp;synchronous]</b>  |
|                            | <b><u>IDEMPOTENT</u></b>         | <b><u>&amp;idempotent]</u></b>                                      |
|                            | <b>[ALWAYS RESPONDS</b>          | <b>&amp;alwaysReturns]</b>  |
|                            | <b>[INVOKE PRIORITY</b>          | <b>&amp;InvokePriority]</b>   |
|                            | <b>[RESULT-PRIORITY</b>          | <b>&amp;ResultPriority]</b>   |
|                            | <b>[CODE</b>                     | <b>&amp;operationCode]</b>  |
| }                          |                                  |   |

Cambiar la referencia del tercer módulo como sigue, (con la modificación que se indica subrayada):

**Remote-Operations-Useful-Definitions {joint-iso-itu-t remote-operations(4) useful-definitions(7) version2(1)}**

Cambiar la definición de la operación no-op OPERATION añadiendo el siguiente campo que se indica (subrayado):

|                            |                          |                    |
|----------------------------|--------------------------|--------------------|
| <b>no-op OPERATION ::=</b> |                          |                    |
| {                          |                          |                    |
|                            | <b><u>IDEMPOTENT</u></b> | <b><u>TRUE</u></b> |
|                            | <b>ALWAYS RESPONDS</b>   | <b>FALSE</b>       |
|                            | <b>CODE</b>              | <b>local:-1</b>    |
| }                          |                          |                    |

Anãdir los nuevos ítems siguientes a este módulo:

```

probe OPERATION ::=
{
    ARGUMENT SEQUENCE
    {
        invokeId [0] InvokeId
    }
    RESULT ENUMERATED{running(0), finished(1), unknown(2), ...}
    IDEMPOTENT TRUE
    CODE local:-2
}

acknowledge OPERATION ::=
{
    ARGUMENT InvokeId
    RESULT ENUMERATED{acknowledged(0), unknown(1), ...}
    IDEMPOTENT TRUE
    CODE local:-3
}

ProbeAndAcknowledge OPERATION ::= {probe | acknowledge}

cancel OPERATION ::=
{
    ARGUMENT InvokeId
    ERRORS {cancelFailed}
    IDEMPOTENT TRUE
    CODE local:-4
}

cancelFailed ERROR ::=
{
    PARAMETER SET
    {
        problem [0] CancelProblem,
        operation [1] InvokeId
    }
    CODE local:-2
}

CancelProblem ::= ENUMERATED
    {unknownOperation(0), tooLate(1), operationNotCancellable(2), ...}

cancelled ERROR ::= {CODE local:-3}

```

## 10) Anexo D

Introducir en el cuadro los siguientes cambios (que se indican subrayados):

| Cláusula | Valor de identificador de objeto   |
|----------|--|
| Anexo A  | {joint-iso-itu-t remote-operations(4) informationObjects(5) version2( <u>1</u> )}<br>{joint-iso-itu-t remote-operations(4) useful-definitions(7) version2( <u>1</u> )} |