

# JNET: NIVEL DE RED PARA RADIO-PAQUETE AX.25

Julián Muñoz Domínguez - ea4acl

9 de octubre de 1999

A mi familia, amigos, y grandes amores.

Especiales agradecimientos a *Luis Alfredo Ramos (EB4EUF)* por la tarea de relectura y corrección del proyecto, a *Simon Mudd (EA4ELS/G0FNB)* por las facilidades logísticas de impresión de documentos, y a *Eduardo Martínez (EC4DFP/EA4AXU)* por las pruebas realizadas vía radio-paquete.

A todos los familiares y amigos, que me han apoyado y animado en los momentos difíciles.

Agradecimientos al equipo de personas repartidas en el mundo que han hecho posible la existencia de un entorno de software gratuito, sin el cual no hubiera sido posible realizar este proyecto. Especialmente:

- Al software  $\text{L}\text{\AA}\text{X}$  y  $\text{L}\text{\TeX}$  para la edición del proyecto.
- Al entorno gráfico de ventanas KDE.
- Al sistema operativo GNU-Linux.

# Índice General

<b>1</b>	<b>El radio-paquete</b>	<b>6</b>
1.1	Introducción al radio-paquete . . . . .	6
1.2	Desorden del radio-paquete . . . . .	7
1.3	El protocolo AX.25 versión 2.0 . . . . .	8
1.4	El protocolo AX.25 versión 2.2 . . . . .	8
1.5	Jnet - Estudio de un nuevo nivel de red . . . . .	8
<b>2</b>	<b>Los niveles de red existentes</b>	<b>10</b>
2.1	Flexnet . . . . .	10
2.2	Rose . . . . .	11
2.3	Net/Rom . . . . .	12
2.4	INP . . . . .	13
<b>3</b>	<b>Modelo de red de Jnet</b>	<b>14</b>
3.1	Características . . . . .	14
3.2	Modelo de grafo de la red . . . . .	15
3.3	Parámetros de los enlaces . . . . .	17
3.3.1	Parámetros característicos . . . . .	17
3.3.2	Parámetros primarios y funciones características . . . . .	18
3.3.3	Magnitudes y codificación de los parámetros primarios . . . . .	19
3.4	Mecanismo de comunicación . . . . .	20
3.4.1	¿ Circuitos virtuales o datagramas ? . . . . .	20
3.4.2	Tipos de circuitos virtuales . . . . .	20
3.4.3	Características de los circuitos virtuales . . . . .	21
3.4.4	Multiplexación de servicios . . . . .	22
<b>4</b>	<b>Modelo de nodo</b>	<b>23</b>
4.1	Enfoque - Modelo funcional . . . . .	23
4.2	Subsistema de acceso a la red . . . . .	23
4.3	Encaminador . . . . .	25
4.4	Máquina de circuito virtual . . . . .	27
4.5	Máquina de gestión de circuitos virtuales . . . . .	28
4.6	Máquinas de medición de enlaces a nivel de red . . . . .	29

4.7	Máquina de gestión de red . . . . .	29
4.7.1	Tabla de proximidad . . . . .	30
4.7.2	Mapa de red . . . . .	31
4.7.3	Comunicación con otros elementos del nodo . . . . .	31
4.8	Máquina de distribución del mapa de red . . . . .	32
4.9	Decodificador de señalización . . . . .	33
<b>5</b>	<b>Mediciones</b>	<b>34</b>
5.1	Modificaciones de AX.25 v2.2 . . . . .	34
5.2	Máquina de medición de enlace a nivel de enlace - MEASUREMENT MACHINE . . . . .	34
5.2.1	Ubicación . . . . .	34
5.2.2	Especificación . . . . .	36
5.3	Ampliación de AX.25 v2.2 para DL-DATA Confirm . . . . .	44
5.3.1	Funcionamiento y carencias del fragmentado en AX.25 v2.2 . . . . .	44
5.3.2	Cambio de nombre de las primitivas . . . . .	44
5.3.3	Funcionamiento . . . . .	46
5.4	Procedimiento general de medición . . . . .	51
5.4.1	Fase 1 - Petición o recepción de medidas . . . . .	51
5.4.2	Fase 2 - Filtrado paso bajo de medidas . . . . .	52
5.4.3	Fase 3 - Cuantificación mediante umbrales . . . . .	55
5.4.4	Fase 4 - Codificación . . . . .	56
5.5	Medición de Pj . . . . .	57
5.5.1	Procedimiento en el nodo origen de un enlace . . . . .	57
5.5.2	Procedimiento en el nodo destino de un enlace . . . . .	58
5.6	Medición de Fj . . . . .	60
5.6.1	Enlaces con conexión hdlc . . . . .	60
5.6.2	Enlaces sin conexión hdlc . . . . .	61
5.7	Medición de Cj . . . . .	63
5.7.1	Análisis general . . . . .	63
5.7.2	Enlaces con conexión hdlc . . . . .	64
5.7.3	Enlaces sin conexión hdlc . . . . .	70
<b>6</b>	<b>Control de flujo</b>	<b>72</b>
6.1	Control de acceso a la red . . . . .	72
6.1.1	Objetivos . . . . .	72
6.1.2	Control de flujo de Gallager . . . . .	73
6.1.3	Retardo incremental total del circuito virtual . . . . .	74
6.1.4	Función umbral del retardo incremental total . . . . .	75
6.1.5	Ejemplo . . . . .	78
6.1.6	Aplicación a Jnet . . . . .	78
6.1.7	Medición de $r_{actual}$ y regulación del tráfico . . . . .	83
6.2	Control de flujo interno . . . . .	86

6.2.1	Objetivos y necesidades . . . . .	86
6.2.2	Modelo elegido: back pressure . . . . .	86
6.2.3	Topología . . . . .	87
6.2.4	Procedimientos generales . . . . .	88
6.2.5	Optimización : ocultación de congestiones . . . . .	92
6.2.6	Optimización: anulación de ondas de back pressure . . . . .	95
6.2.7	Modelado del control de flujo interno mediante máquinas . . . . .	95
6.2.8	Nodos N . . . . .	96
6.2.9	Nodos X . . . . .	100
6.2.10	Nodos Y . . . . .	100
6.2.11	Nodos MD . . . . .	100
6.2.12	Nodos MI . . . . .	107
6.2.13	Nodos O . . . . .	107
6.2.14	Criterios de congestión de enlaces . . . . .	116
6.2.15	Procedimiento de información de congestión . . . . .	117
<b>7</b>	<b>Encaminamiento</b>	<b>118</b>
7.1	Procedimiento general . . . . .	118
7.2	Métricas . . . . .	118
7.3	Probabilidad de error . . . . .	119
7.4	Primitiva de establecimiento de conexión . . . . .	120
<b>A</b>	<b>Filtro digital para mediciones síncronas</b>	<b>122</b>
<b>B</b>	<b>Parámetros del nodo</b>	<b>128</b>
<b>C</b>	<b>Primitivas</b>	<b>131</b>
<b>D</b>	<b>Medición de la capacidad de un canal a partir de la longitud media de la cola</b>	<b>134</b>
<b>E</b>	<b>Función umbral del retardo incremental total <math>L(r)</math></b>	<b>136</b>
<b>F</b>	<b>Control de flujo de acceso a la red: implementación del algoritmo de resolución numérica de <math>r_{max}</math>.</b>	<b>138</b>
<b>G</b>	<b>Información acerca de este documento</b>	<b>142</b>
<b>H</b>	<b>Licencia de uso</b>	<b>143</b>

# Capítulo 1

## El radio-paquete

### 1.1 Introducción al radio-paquete

El radio-paquete es un sistema que permite enlazar varios sistemas telemáticos a través de un canal radioeléctrico compartido. Se basa en el AX.25, *el protocolo de nivel de enlace* del radio-paquete, y una disciplina de acceso múltiple al canal por detección de portadora y sin detección de colisiones.

Más allá de consideraciones técnicas, quiero destacar que este sistema representa una nueva área de experimentación en la actividad del radioaficionado, el área de las redes de ordenadores en entornos radioeléctricos. Se trata de un entorno específico y difícilmente repetible, unos ciudadanos en sus casas, con sus medios, experimentando de forma altruista, con la finalidad de aprender y comunicarse. Antes del boom de Internet, y del correo electrónico, ya existía una red mundial de correo electrónico apoyada solamente en radio-paquete, y aparecieron multitud de servicios y técnicas, utilizando por ejemplo enlaces vía satélite, enlaces de alta velocidad, o de baja velocidad en onda corta, como queda expuesto en el documento “Comunicaciones digitales en el entorno de la radioafición” [1].

Estas técnicas se han visto ampliamente superadas, con los avances de la telecomunicación comercial; la telefonía móvil e Internet han hecho posible lo que muchos radioaficionados deseaban conseguir con el radio-paquete. Pero estos nuevos sistemas siguen funcionamientos diferentes al radio-paquete, porque su finalidad es completamente diferente: mientras unos dan un servicio fiable y de calidad a la sociedad, el otro solo pretende ser un caldo de cultivo de experimentación y de servicios dirigidos a los radioaficionados.

Por otro lado, cabe destacar que el radio-paquete no tiene definido un nivel de red estándar. Existen diseños e implementaciones de niveles de red, algunos se están utilizando, pero como veremos a continuación el entorno de la radioafición hace realmente difícil realizar un nivel de red adaptado a todas las necesidades.

## 1.2 Desorden del radio-paquete

El entorno de utilización del radio-paquete es caótico por naturaleza, por los siguientes motivos:

- Las estaciones siguen una disposición geográfica aleatoria, siendo mayor su densidad en los núcleos de población (frecuente congestión) y menor en zonas o provincias de baja densidad de población (amplias zonas sin cubrir).
- No existe una autoridad coordinadora de las estaciones radioeléctricas, siendo prácticamente imposible garantizar que los operadores de las estaciones cooperen y tomen decisiones y acuerdos en cortos plazos de tiempo. El comportamiento de las estaciones radioeléctricas depende directamente de lo que decidan sus operadores, y es por lo tanto imposible de prever.
- Existe diversidad en frecuencia. Cada estación tiene transceptores radioeléctricos sintonizados en las frecuencias definidas por el operador.
- Existe movilidad. Algunos sistemas de radio-paquete se utilizan en estaciones móviles (ver [1]).
- El medio radioeléctrico es variable. Algunas de las bandas de operación de los radioaficionados tienen comportamientos difíciles de prever.
- Es habitual utilizar anchos de banda estrechos, y tener relaciones señal a ruido bajas.
- Las estaciones tienen dificultades para compartir el medio radioeléctrico:
  - La disciplina de acceso al canal no es muy eficiente.
  - Problema de la estación oculta: ocurre frecuentemente que una estación oiga a dos estaciones que no se oyen entre ellas. Estas estaciones van a colisionar sistemáticamente.
- Asimetría de los enlaces: las estaciones radioeléctricas están equipadas de desigual manera, y pueden existir interferencias locales, haciendo que solo sea posible la comunicación en un sentido.

Este desorden dificulta el diseño de un nivel de red. En sistemas profesionales se solucionan los problemas de forma distribuida a lo largo de todo el diseño: calidad de los sistemas digitales, calidad de las estaciones radioeléctricas, emplazamiento de los nodos, sistemas de recuperación de errores. Aquí esto no es posible, no se puede actuar físicamente sobre la estación, porque suponen costes y trabajo para el operador, hay que adaptarse a las condiciones existentes y sacarles el mejor provecho posible.

### 1.3 El protocolo AX.25 versión 2.0

Actualmente se utiliza la versión 2.0 del protocolo AX.25, definido por la American Radio Relay League [2]. AX.25 2.0 está implementado en multitud de software y hardware de radioaficionados. Sus principales características son:

- Sigue el estándar HDLC del CCITT, protocolo de nivel de enlace orientado a conexión.
- Trabaja en modo balanceado exclusivamente. No existen estaciones maestras ni esclavas.
- Posibilidad de mandar datos a través de una conexión establecida (modo numerado, o conectado), o sin conexión (modo no numerado, de datagramas o “unproto”).
- Rechazo simple.

### 1.4 El protocolo AX.25 versión 2.2

La última revisión del protocolo AX.25 fue publicada en Noviembre de 1997 [3], y no ha sido implementada aún. Son muchas las mejoras, pero su complejidad aumenta significativamente. Las principales novedades con respecto a la versión 2.0 son:

- El protocolo se describe mediante especificación SDL<sup>1</sup>.
- El nivel de enlace se divide en 4 máquinas de estados finitos extendidas.
- Se define la capa superior del nivel físico, y su interfaz con el nivel de enlace. Se presentan dos máquinas: para full-dúplex y simplex.
- Se especifican las primitivas de interacción entre niveles de red, enlace y físico, así como la primitivas de interacción entre las máquinas internas del nivel de enlace.
- Se especifica la negociación de parámetros.
- Rechazo selectivo.

### 1.5 Jnet - Estudio de un nuevo nivel de red

Hemos visto que el entorno del radio-paquete tiene muchas condiciones adversas. *La elección de las condiciones adversas que se tratarán prioritariamente condicionará el diseño del nivel de red.* De esta elección inicial va a depender el resto del estudio.

Es por lo tanto fundamental el criterio de elección. Este criterio es personal, como usuario y administrador de sistemas de radio-paquete, experimentando desde hace 8

---

<sup>1</sup>Mediante diagramas SDL *simplificados* orientados a entender el algoritmo de cada proceso.

años en este entorno, he detectado necesidades no cubiertas o mal cubiertas por los actuales sistemas.

Los principales objetivos de este nuevo nivel de red, bautizado *JNET*, serán:

1. Aprovechar al máximo los recursos disponibles.  
La tara de información producida por el nivel de red debe ser optimizada.
2. Ser poco exigente con el equipamiento utilizado.  
No se exige una probabilidad de error mínima de los enlaces, ni una tasa binaria mínima.
3. Ser flexible con la arquitectura de la red.  
No se exigen disposiciones particulares de los nodos (en estrella, en árbol, etc...).
4. Ser claro.
5. Ser abierto, dando pie a la experimentación.

Estos objetivos son decisivos a la hora de crear y optar por soluciones. Los dos últimos llevan a tomar dos decisiones iniciales:

1. Este proyecto fin de carrera se distribuye libremente con licencia de distribución OpenContent, ver apéndice H.
2. El nivel de red se va a apoyar en AX25 versión 2.2. La utilización de primitivas y de diagramas SDL mejora muchísimo la claridad y facilita la modificación y mejora de este trabajo.

Este proyecto *NO incluye* la descripción del formato del protocolo de red, se limita a estudiar y proponer soluciones mediante modelos, procedimientos y recomendaciones<sup>2</sup>. Es un sistema dividido en subsistemas funcionales que interaccionan coherentemente, y solucionan problemas mediante procedimientos factibles.

---

<sup>2</sup>Muchos de los cuales, al igual que en AX25 versión 2.2, son descritos mediante diagramas SDL *simplificados*.

# Capítulo 2

## Los niveles de red existentes

Empezaremos analizando los niveles de red de radio-paquete existentes actualmente, para poder sacar lo mejor de cada uno. La mayoría de estos no tiene una descripción formal de su modelo topológico, esta es implícita al funcionamiento de los nodos. Más problemático es el tratamiento de algunos niveles de red como *rose* y *flexnet*, que ni siquiera tienen especificaciones públicas de funcionamiento interno, solo tienen descripciones de uso para los usuarios finales. Se ha llevado a cabo una investigación, consistente en ponerse en contacto con los autores o implementadores de estos protocolos, consiguiendo información no publicable, además de experimentar con estas redes a nivel práctico en enlaces reales de radio-paquete; por estos motivos, algunas de las descripciones hechas a continuación no son directamente comprobables en la bibliografía.

### 2.1 Flexnet<sup>1</sup>

Flexnet es el nivel de red más potente, el más utilizado en Europa, pero es el más opaco (no hay documentación) en lo que refiere a su funcionamiento interno. Su funcionamiento se reparte en dos planos: plano de control, y plano de transporte.

El plano de control establece la señalización entre nodos, realiza las medidas de los enlaces y propaga el mapa de la red. Este intercambio de señalización sólo es posible si puede realizarse la conexión entre los nodos adyacentes. La calidad de los enlaces se caracteriza por el tiempo de ida y vuelta a través del enlace que tarda en recorrer una cantidad determinada de información, medición que realizan periódicamente los nodos.

El plano de transporte realiza el establecimiento de circuitos virtuales, y mueve la información de usuarios mediante un procedimiento poco común: se utiliza el campo de digipeaters de las tramas AX.25 (repetidores digitales, un campo creado para enrutamiento definido en el origen, ver [1]), tanto para la comunicación entre nodos y entre nodos y usuarios finales. Esto permite que las conexiones a nivel de red sean

---

<sup>1</sup>Ver <http://dl0td.afthd.tu-darmstadt.de/~flexnet/> [4]

conexiones AX.25 sin necesidad de encapsular ninguna pdu de red, los paquetes se propagan entre nodos. Estos nodos en terminología flexnet se llaman “digipeaters”, porque actúan como repetidores digitales AX.25, con la diferencia que del campo de digipeaters de las tramas retransmitidas se inserta y quita dinámicamente la dirección del siguiente nodo del camino.

Estos saltos son manejados de forma independiente. Aunque parezca que los datos entrantes al nodo sean retransmitidos ciegamente (este “parecido” se debe a que es el comportamiento normal de los repetidores digitales AX.25), realmente son dos enlaces independientes, en los que se produce el control de flujo y retransmisiones localmente. El control de flujo utiliza las tramas de control de flujo del nivel de enlace (RNR receiver not ready), y se propagan hacia atrás por procedimiento de back pressure (ver [5]).

Otro detalle de importancia es que flexnet también puede transportar tramas no numeradas, aunque en este caso su comportamiento es idéntico a un digipeater AX.25 (retransmisión ciega). Pero además es posible unir diferentes redes flexnet mediante digipeaters no flexnet. Basta incluir la dirección de este digipeater en la llamada original.

El encaminamiento es dinámico, de tipo “estado de enlaces”.

La principal fuerza de este sistema es su gran eficiencia, y su polivalencia, que probablemente a efectos prácticos contrapesen sus inconvenientes, que son:

- Secretismo del protocolo de señalización inter-nodos y criterios de control de flujo.
- Necesidad de establecimiento de enlaces para detección de nodos adyacentes e intercambio de señalización.
- Necesidad de enlaces des congestionados de suficiente calidad y simétricos. El envío de los paquetes de medición de tiempos se hace periódicamente. Flexnet recomienda que los nodos intermedios estén conectados en full-duplex, y no tener varios nodos en la misma frecuencia.

Sin lugar a dudas, este nivel de red es uno de los mejores en la actualidad.

## 2.2 Rose<sup>2</sup>

Al igual que flexnet, utiliza el campo de digipeaters para el acceso a la red. Pero el parecido acaba ahí, ya que para la comunicación entre nodos se utiliza la especificación X.25 en su nivel de paquetes (PLP)[6]. No hay ninguna medición de la calidad de los enlaces, el encaminamiento es estático (definido por el operador del nodo, con posibilidades de rutas alternativas) pero protegido contra bucles [7] (se utilizan facilidades

---

<sup>2</sup>Ver página web de rose, <http://www.rats.org/rose>

X.25 para este fin), el control de flujo sigue un modelo “back pressure” mediante los paquetes de control RNR de PLP/X.25 [8].

Es menos flexible que flexnet: no permite el transporte de tramas no numeradas, la utilización de digipeaters AX.25 intermedios sólo es posible entre el usuario final y el nodo de acceso.

Es menos eficiente que flexnet: hace lo mismo que flexnet, pero utilizando PLP, lo cual supone mayor tara de información.

## 2.3 Net/Rom<sup>3</sup>

Cronológicamente Net/Rom es el primer nivel de red para radioaficionados. Su modelo de red se basa en un encaminamiento que utiliza el procedimiento de vector de distancia. Tiene por lo tanto los problemas de los sistemas con algoritmo de vector de distancia (los nodos sólo conocen la topología cercana de la red, peores decisiones, peligros de bucles, borrado de enlaces defectuosos).

Es menos flexible que flexnet: no permite el transporte de tramas no numeradas, y la utilización de digipeaters intermedios no es posible, salvo si el operador del nodo crea manualmente algún enlace con un nodo vecino a través de digipeater.

La detección de nodos adyacentes es dinámica, mediante la transmisión periódica de señalización al canal que incluye la lista de destinos alcanzables. La calidad de los enlaces es definida por el operador en cada interfaz<sup>4</sup>; se define una calidad por defecto para un interfaz determinado, y todo enlace a un nodo adyacente detectado por este interfaz tendrá esta calidad.

El protocolo se compone de dos protocolos: un protocolo de nivel de red no orientado a conexión, y un protocolo de nivel 4 orientado a conexión, encargado del establecimiento de circuitos virtuales, del control de flujo (mediante algoritmo de ventanas deslizantes) y de la detección de errores.

Son muchos los problemas de Net/Rom. A continuación describo los más importantes:

- No se adapta a canales con cambios frecuentes de propagación (la mayoría), ya que el procedimiento de detección de adyacencias no incluye un procedimiento lo suficientemente rápido de detección de fallo de enlaces. Esto unido al algoritmo de vector de distancia hace que en condiciones variables la mayoría de los nodos no estén alcanzables, y las rutas estén equivocadas. Un ejemplo claro: si recibimos en un pico de propagación la lista de destinos alcanzables de un nodo, se borrarán rutas anteriores alcanzables, y serán sustituidas por otras rutas que no funcionarán.
- En la práctica se observan bucles.

---

<sup>3</sup>Ver <ftp://hes.iki.fi/pub/doc/ax25-doc-1.0.tar.gz> / netrom.ps [9]

<sup>4</sup>El interfaz es el puerto por el que se accede a un determinado canal radioeléctrico, pudiendo haber nodos multi-puertos

- El procedimiento de control de flujo por ventana es muy lento e ineficiente en circuitos de altos tiempos de respuesta [5].
- La calidad de los enlaces es inexacta, debido a que en la práctica en un mismo canal la calidad de los enlaces puede variar mucho.

## 2.4 INP<sup>5</sup>

INP (Internode Protocol) es una mejora de Net/Rom. Tiene muchos aspectos a tener en consideración.

Sus características son las siguientes:

- Mismo nivel de transporte que Net/Rom.
- Se abandona el sistema de difusión de rutas.
- Se define un nuevo protocolo de señalización entre nodos, que se transmite por enlaces establecidos y no por difusión; permite:
  - Intercambio de rutas.  
Un nodo informa a su vecino de todos los destinos alcanzables (vector de distancia). Puesto que ahora esta información va dirigida a un nodo concreto (al nodo con el que estamos conectados), se pueden aplicar procedimientos de optimización de vectores de distancia (algoritmo de Poison).
  - Medición de enlaces.  
Al igual que flexnet, se mide el tiempo de ida y vuelta para definir la calidad de un enlace.
  - Transporte de señalización de usuario.  
De esta forma se permite difundir información útil para otros protocolos, por ejemplo la resolución de dirección ARP puede llevarse a cabo mediante este mecanismo.

Como inconvenientes:

- Sigue utilizando vectores de distancia (poca escalabilidad).
- Más tara aún que Net/Rom, ya que aparte de las cabeceras de los dos protocolos (red y transporte) encapsulados en las conexiones, hay que enviar paquetes periódicamente para medir el tiempo de ida y vuelta.

---

<sup>5</sup>Ver <http://www.nordlink.org/eng/inp3.htm> [10]

# Capítulo 3

## Modelo de red de Jnet

A lo largo de este proyecto, se exponen y analizan estructuradamente diferentes enfoques funcionales de Jnet (control de flujo, encaminamiento, mediciones, modelo de nodo y red ...). Todos ellos están íntimamente relacionados, cada uno supone la presencia de “piezas de engranaje” que *deben ser factibles de realizar*. En multitud de ocasiones, el análisis ha llevado a elementos demasiado complejos o imposibles de realizar; el proceso seguido en estos casos ha sido cíclico, consistente en modificar otros elementos de algunos enfoques funcionales, hasta llegar a una solución global coherente.

El resultado *final* es el expuesto en este capítulo y los siguientes, *no tratándose de modelos y enfoques arbitrarios*, sino condicionados por la condición de realizabilidad de los elementos y la consecución de un diseño coherente.

En este capítulo se define el modelo topológico de la red Jnet; es decir, cómo es el grafo que representa a un conjunto de nodos y la comunicación entre ellos, qué tipo de nodos y enlaces existen, cómo se caracterizan, y el uso de circuitos virtuales.

### 3.1 Características

Ya conocidas las características de los niveles de red existentes, mi intención es diseñar un nivel de red *diferente* a los demás, que aporte aspectos nuevos, que siga *enfoques y conceptos nuevos*, lo suficientemente generales para ser aplicables también en protocolos que se diseñen e implementen en el futuro. El objetivo es enriquecer lo existente, y proponer alternativas.

Las dos características que presentamos a continuación condicionan por completo el modelo de red elegido:

#### Enlaces unidireccionales

Vamos a permitir la utilización de enlaces *completamente* unidireccionales. Esto significa poder utilizar *equipos que sólo puedan recibir, o sólo puedan transmitir*. Esta

capacidad ausente en los actuales niveles de red permite el uso de tales equipos, que son muy económicos y simples, especialmente en el caso de los receptores. En una configuración típica de estación de aficionado, la parte crítica es la transmisión a alta velocidad; con esta opción, es posible utilizar receptores sencillos capaces de recibir señales codificadas a una alta velocidad binaria y un transmisor más sencillo de velocidad más baja<sup>1</sup>.

Esta configuración no es el clásico full-duplex donde las frecuencias del canal de ida y de vuelta están fijadas a priori. Aquí no es necesario prefijar, el propio nivel de red debe ser capaz de aprender por donde transmitir en caso de tener varios canales de transmisión posibles, y adaptarse a las características del canal escogido.

### Enlaces bidireccionales asimétricos

Otro aspecto olvidado en los actuales protocolos de red es la alta asimetría de los enlaces. En este aspecto se desea dar un enfoque extremadamente flexible. Que una estación tenga un canal con capacidad de transmisión y recepción no significa que deba utilizar ambas. Puede desear que a pesar de tener establecida una conexión hdlc, esta sea utilizada solamente en un sentido, o en vez de tener una conexión hdlc tenga dos enlaces unidireccionales (sin conexión hdlc) independientes. En Jnet, los dos sentidos de la comunicación serán medidos y modelados de forma independiente.

## 3.2 Modelo de grafo de la red

El modelo de red Jnet se compone de nodos que encaminan la información a través de enlaces con conexión hdlc (modo numerado), o sin conexión hdlc (modo no numerado), con las siguientes características:

- Todos los nodos son idénticos, son el punto de unión de los enlaces.
- Los enlaces se caracterizan con varios parámetros.
- No aparecen reflejados explícitamente los canales radioeléctricos.

La red se compone de 3 elementos:

Nombre	Representación gráfica
Nodo	○
Enlace sin conexión hdlc	→
Enlace con conexión hdlc	┌→

---

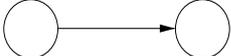
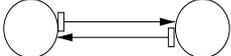
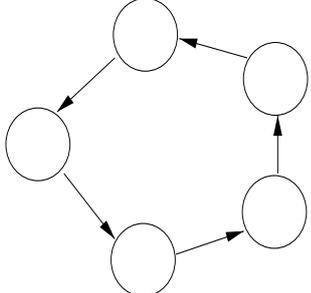
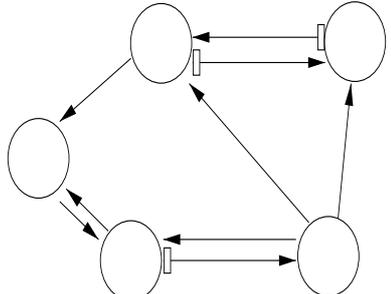
<sup>1</sup>El uso de equipos que solo puedan transmitir exige un cuidado especial en la prevención de las colisiones ya que la política de acceso al canal es ALOHA en vez de CSMA.

**Enlace sin conexión hdlc** significa que el envío se hace mediante tramas no numeradas.

**Enlace con conexión hdlc** significa que una conexión AX.25 debe establecerse entre los dos nodos, por la cual se transmitirá la información.

**El sentido de la flecha** especifica el sentido de la comunicación. El envío bidireccional de información se modela mediante dos enlaces en sentido opuesto.

### Ejemplos

Dos nodos unidos por un enlace con conexión hdlc	
Dos nodos unidos por un enlace sin conexión hdlc	
Dos nodos comunicados bidireccionalmente por dos enlaces sin conexión hdlc	
Dos nodos comunicados bidireccionalmente por un enlace con conexión hdlc (ver nota)	
Anillo	
Configuración mixta	

*Nota: Es un único enlace hdlc, pero utilizado para transmitir información bidireccionalmente. Por este motivo aparecen dos flechas.*

### 3.3 Parámetros de los enlaces

Los parámetros de los enlaces se estructuran en dos niveles, mediante funciones matemáticas que llamamos “funciones características”:

**Parámetros característicos** Definen el enlace de forma abstracta a nivel de grafo. Son los únicos tomados en consideración para las operaciones que se basan en el grafo (o mapa) de la red.

**Parámetros primarios** Son magnitudes medidas y codificadas por un nodo, caracterizando uno de los enlaces que entran o sale de este nodo, y posteriormente comunicadas al resto de la red. Mediante el tratamiento de parámetros primarios se obtienen los parámetros característicos de los enlaces.

**Funciones características** Toman como parámetros los parámetros primarios y su resultado es un parámetro característico.

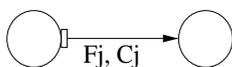
Mediante esta estructuración se consigue una mayor *flexibilidad*:

- Se puede mantener el mismo nivel de red manteniendo los parámetros característicos, incluso *cambiando el tipo de parámetros primarios medidos en cada nodo*. En Jnet por ejemplo, existen varias formas de medir las tasa binarias de los enlaces; si los algoritmos propuestos en este proyecto por motivos específicos no resultaran factibles de implementar, siempre cabría la posibilidad de realizar otras medidas y cambiar los parámetros primarios (y consecuentemente las funciones características).
- Se pueden realizar *medidas de forma distribuida*. Un parámetro característico puede ser obtenido mediante parámetros primarios transmitido por el nodo destino y el nodo origen de un enlace. Por ejemplo la probabilidad de error de paquete podría ser obtenida realizando un coeficiente entre el número de paquetes transmitidos por el nodo origen (=parámetro primario medido y transmitido por el nodo origen) y el número de paquetes recibidos por el nodo destino (=parámetro primario medido y transmitido por el nodo destino).
- Se puede cambiar el modelo de grafo de la red (parámetros característicos) manteniendo las magnitudes medidas.

#### 3.3.1 Parámetros característicos

##### Enlace con conexión hdlc

Son definidos por dos parámetros,  $C_j$  y  $F_j$ .



**F<sub>j</sub>** es la tasa binaria de la información transportada por el enlace, en bits por segundo.

**C<sub>j</sub>** es la capacidad máxima de este enlace en bits por segundo.

### Enlace sin conexión hdlc

Son definidos por tres parámetros, C<sub>j</sub>, F<sub>j</sub> y P<sub>j</sub>.



**F<sub>j</sub>** es la tasa binaria de la información transportada por el enlace, en bits por segundo.

**C<sub>j</sub>** es la capacidad máxima de este enlace en bits por segundo.

**P<sub>j</sub>** es la probabilidad de no error de paquete en el enlace (o bien la probabilidad de que un paquete no se pierda en el enlace).

## 3.3.2 Parámetros primarios y funciones características

### Enlaces con conexión hdlc

En los enlaces con conexión hdlc, los parámetros característicos son iguales a los parámetros primarios, y por lo tanto las funciones características son la función identidad.

**F<sub>j<sub>p</sub></sub>, C<sub>j<sub>p</sub></sub>** son los parámetros primarios.

**K, L** son las funciones características, tales que:

$$F_j = K(F_{j_p}) \quad C_j = L(C_{j_p})$$

Siendo  $K=L=Id$  (función identidad)

### Enlaces sin conexión hdlc

En los enlaces sin conexión hdlc, los parámetros característicos F<sub>j</sub> y P<sub>j</sub> son iguales a los parámetros primarios, y por lo tanto sus dos funciones características M y O son la función identidad.

En cambio C<sub>j</sub> se obtiene combinando F<sub>j<sub>p</sub></sub>, P<sub>j<sub>p</sub></sub> y otro parámetro primario H<sub>j<sub>p</sub></sub>: H<sub>j<sub>p</sub></sub> es el ancho de banda de canal no utilizado.

**F<sub>j<sub>p</sub></sub>, H<sub>j<sub>p</sub></sub>, P<sub>j<sub>p</sub></sub>** son los parámetros primarios.

**M, N, O** son las funciones características, tales que:

$$F_j = M(F_{j_p}) \quad C_j = N(F_{j_p}, P_{j_p}, H_{j_p}) \quad P_j = O(P_{j_p})$$

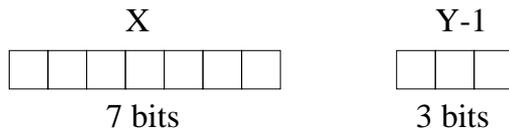
Siendo  $M=O=Id$  (función identidad)

N se especifica en el apartado 5.7.3.

### 3.3.3 Magnitudes y codificación de los parámetros primarios

Para contribuir a la máxima eficiencia de la distribución de los parámetros primarios en la red, vamos a definir su rango de valores y su codificación.

**F<sub>j<sub>p</sub></sub>, C<sub>j<sub>p</sub></sub>, H<sub>j<sub>p</sub></sub>** Las tasas binarias se codifican tal que:  $Valor = X \times 10^Y$ , y se transmite la pareja (X, Y-1) en formato binario, de la siguiente manera:



El rango de estas variables es:  $X \in [0, 127]$ ;  $Y \in [1, 8]$

**Valor mínimo de F<sub>j<sub>p</sub></sub>, C<sub>j<sub>p</sub></sub>, H<sub>j<sub>p</sub></sub>**

$X = 0; Y = 1 \Rightarrow F_{j_p}, C_{j_p}, H_{j_p} = 0$  [Se puede utilizar para marcar un enlace inactivo, bloqueado, o incapaz de transportar más información]

$X = 1; Y = 1 \Rightarrow F_{j_p}, C_{j_p}, H_{j_p} = 10 \text{ bps}$

**Valor máximo de F<sub>j<sub>p</sub></sub>, C<sub>j<sub>p</sub></sub>, H<sub>j<sub>p</sub></sub>**

$X = 127; Y = 8 \Rightarrow F_{j_p}, C_{j_p} = 12.7 \text{ Gbps}$

Por lo tanto:  $F_{j_p}, C_{j_p} \in [10 \text{ bps}, 12.7 \text{ Gbps}]$

**P<sub>j<sub>p</sub></sub>** La probabilidad de no error de paquete se codifica con un entero de 5 bits. Sea P este entero; P es tal que:  $P_{j_p} = P \times \frac{1}{31}$ .  $P_{j_p} \in [0, 1]$ , con saltos de 1/31 (32 niveles).

## 3.4 Mecanismo de comunicación

### 3.4.1 ¿ Circuitos virtuales o datagramas ?

Es la primera gran decisión a tomar. Un nivel de red por datagramas permite un encaminamiento muy dinámico y eficiente, incluso con posibilidades de multiencajamiento y control de flujo optimizado a multi-encaminamiento (ver [5]). La comunicación por datagramas no tiene estados<sup>2</sup>, es decir que el nodo no tiene memoria de una conexión particular, solo conoce el estado de la red, que es el mismo para todos los paquetes que procesa.

Pero Jnet optimiza la comunicación, define el estado de un camino determinado, y el estado de congestión de los enlaces, resuelve los problemas localmente, no extremo a extremo, y necesita que los caminos sean determinísticos; los nodos no serán iguales, actuarán de forma diferente en función de su posición en el camino<sup>3</sup>, el estado de la comunicación no estará solamente en el nodo origen y destino, debe de estar en todos los nodos participantes. En Jnet se van a intentar controlar algunos factores críticos de una comunicación, y para ello se requiere definir estados, algoritmos específicos a cada estado, reservar recursos en el nodo para memorizar el estado y las variables de cada comunicación. Es decir, surge la necesidad de manejar circuitos virtuales para tener una visión global y detallada de cada comunicación; esto no resulta posible (o eficiente) con una red orientada a datagramas.

---

*Jnet es una red orientada a Circuitos Virtuales.*

---

### 3.4.2 Tipos de circuitos virtuales

El circuito virtual es la unidad básica de comunicación entre dos usuarios finales. Debido a la unidireccionalidad o asimetría de los enlaces, los circuitos virtuales serán unidireccionales. Una comunicación bidireccional se modela mediante dos circuitos unidireccionales. No obstante para obtener la plena funcionalidad del control de flujo interno, el circuito de ida debe estar ligado con el circuito de vuelta; esto lleva a definir dos tipos de circuitos virtuales:

**Circuito virtual unidireccional** Define una comunicación unidireccional entre el nodo origen y el nodo destino. El control de flujo interno no podrá alcanzar su plena funcionalidad, y por lo tanto cuando este no sea suficiente se recurrirá al descarte de paquetes. Esta modalidad permite aprovechar situaciones en las que solo existe la posibilidad de comunicación unidireccional a nivel de red entre usuarios. Se trata de una modalidad nueva, que permitiría por ejemplo el uso del protocolo de difusión por satélite Pacsat<sup>4</sup> a nivel de red.

---

<sup>2</sup>El estado de cada pdu viaja con ella, en su cabecera por ejemplo.

<sup>3</sup>Lo observaremos especialmente cuando abordemos el control de flujo interno, ver apartado 6.2.

<sup>4</sup>Ver “Comunicaciones digitales en el entorno de la radioafición” [1]

**Circuito virtual bidireccional** Define una comunicación bidireccional entre dos nodos. Se compone de un circuito virtual de ida y un circuito virtual de vuelta, que no tienen porqué seguir el mismo camino. La denominación “ida” o “vuelta” es completamente arbitraria, ambos sentidos son tratados de la misma forma, pero independientemente. Es necesaria la clara diferenciación de los dos circuitos virtuales que componen el circuito virtual bidireccional, sin que por ello signifique que uno es más importante que otro.

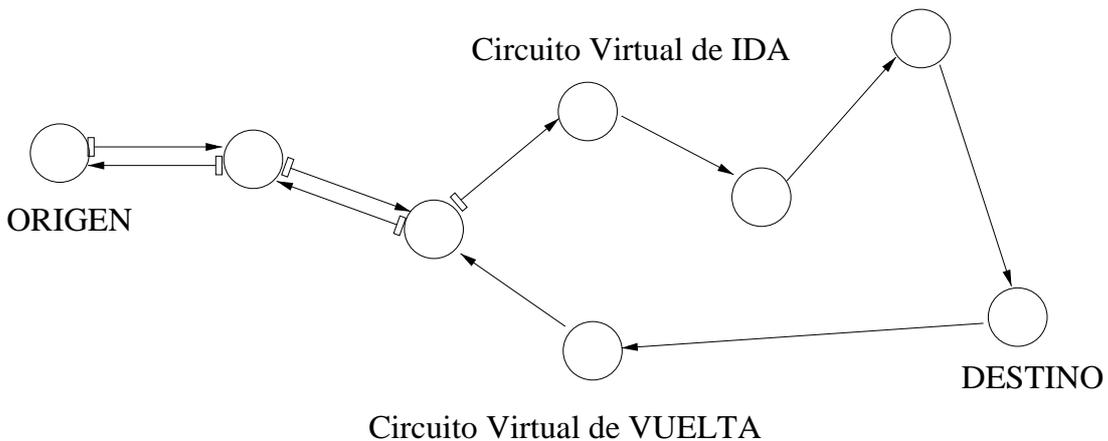


Figura 3.1: Ejemplo de circuito virtual bidireccional

### 3.4.3 Características de los circuitos virtuales

Los circuitos virtuales están caracterizados por los siguientes elementos, que son conocidos por todos los nodos incluidos en ellos:

**Tipo de circuito virtual** Indica si es un circuito virtual de tipo unidireccional o bidireccional.

**Identificador de circuito virtual** Todo circuito virtual tendrá asociado un Identificador único en la red. Los circuitos virtuales de ida y de vuelta compartirán el mismo identificador de circuito virtual.

**Identificador de sentido** Sirve para diferenciar los dos sentidos de la comunicación de los circuitos virtuales bidireccionales.

**Camino completo del circuito virtual** El camino que sigue el circuito virtual es conocido desde su establecimiento.

**Estados, tipos, de los enlaces y los nodos del circuito virtual** Todos son conocidos en detalle, sus evoluciones son seguidas.

### 3.4.4 Multiplexación de servicios

Es posible que cada usuario final desee tener varios servicios en su estación. Existen dos posibilidades:

- Utilización de un circuito virtual independientemente para cada servicio. Cada servicio tiene una dirección de usuario final diferente. AX.25 actúa de esta manera a nivel de enlace: cada servicio se encuentra en una dirección diferente.

Ventajas:

- Al abrirse circuitos virtuales diferentes, el encaminamiento es más óptimo ya que se vuelve a decidir el camino del circuito virtual, produciéndose un efecto de reparto de carga.
- Al no usar otros protocolos intermedios, la eficiencia a la hora de transportar información es óptima.

Inconvenientes:

- Se abre un nuevo circuito virtual cada vez que se quiere acceder a un nuevo servicio, esto supone una carga en la red y la reserva de recursos independientes que hubieran podido ser compartidos.
  - Cada circuito virtual abierto tendrá un procedimiento separado de control de flujo. Si los circuitos virtuales siguen el mismo camino estos procedimientos seguirán una evolución similar y simultánea, siendo más eficiente el tratamiento de la congestión si los servicios se multiplexaran dentro de un mismo circuito virtual.
- Utilización del mismo circuito virtual. Esta solución puede conseguirse con un protocolo de multiplexación sobre el nivel de red que ofrezca varios Puntos de Acceso a Servicio (S.A.P.).

Ventajas:

- No se abre un nuevo circuito virtual a cada vez que se quiere acceder a un nuevo servicio.
- Procedimiento de control de flujo unificado.

Inconvenientes:

- Encaminamiento no óptimo.
- Tara de información de las cabeceras del protocolo de multiplexación, que serán añadidas a cada n-pdu.

La primera solución es directamente aplicable, mientras que la segunda exige el diseño de un sencillo protocolo de multiplexación, que no vamos a especificar en este proyecto.

# Capítulo 4

## Modelo de nodo

### 4.1 Enfoque - Modelo funcional

El nodo se va a presentar funcionalmente: se describen un conjunto de subsistemas y como encajan entre ellos para dar una cierta funcionalidad. El modelo general del nodo se puede observar en la figura 4.1.

### 4.2 Subsistema de acceso a la red

Es el conjunto de sistemas que están en contacto con el interfaz de la capa de red: permiten el acceso a la red de los usuarios del nivel de transporte o superior.

Como podemos observar en la figura 4.2, son 3 los módulos que participan directamente en esta tarea.

**Máquina de control de flujo de acceso a la red** Limita el tráfico inyectado en la red, conforma el tráfico enviado a la máquina del circuito virtual (ver capítulo 6, apartado 6.1).

**Máquina de gestión de circuitos virtuales** Al recibir una petición de establecimiento/cerrado de circuito virtual, la máquina de gestión de circuitos virtuales tendrá por labor la creación/destrucción de este circuito virtual. Reservará/liberará los recursos necesarios para este circuito, especialmente creará/destruirá un proceso específico (“máquina de circuito virtual”) dedicado a la gestión de este circuito virtual particular.

**Encaminador** Es quien entrega las pds dirigidas a los usuarios finales: cuando este nodo es el nodo destino del circuito virtual, el encaminador sabe a qué usuario final entregar la información.

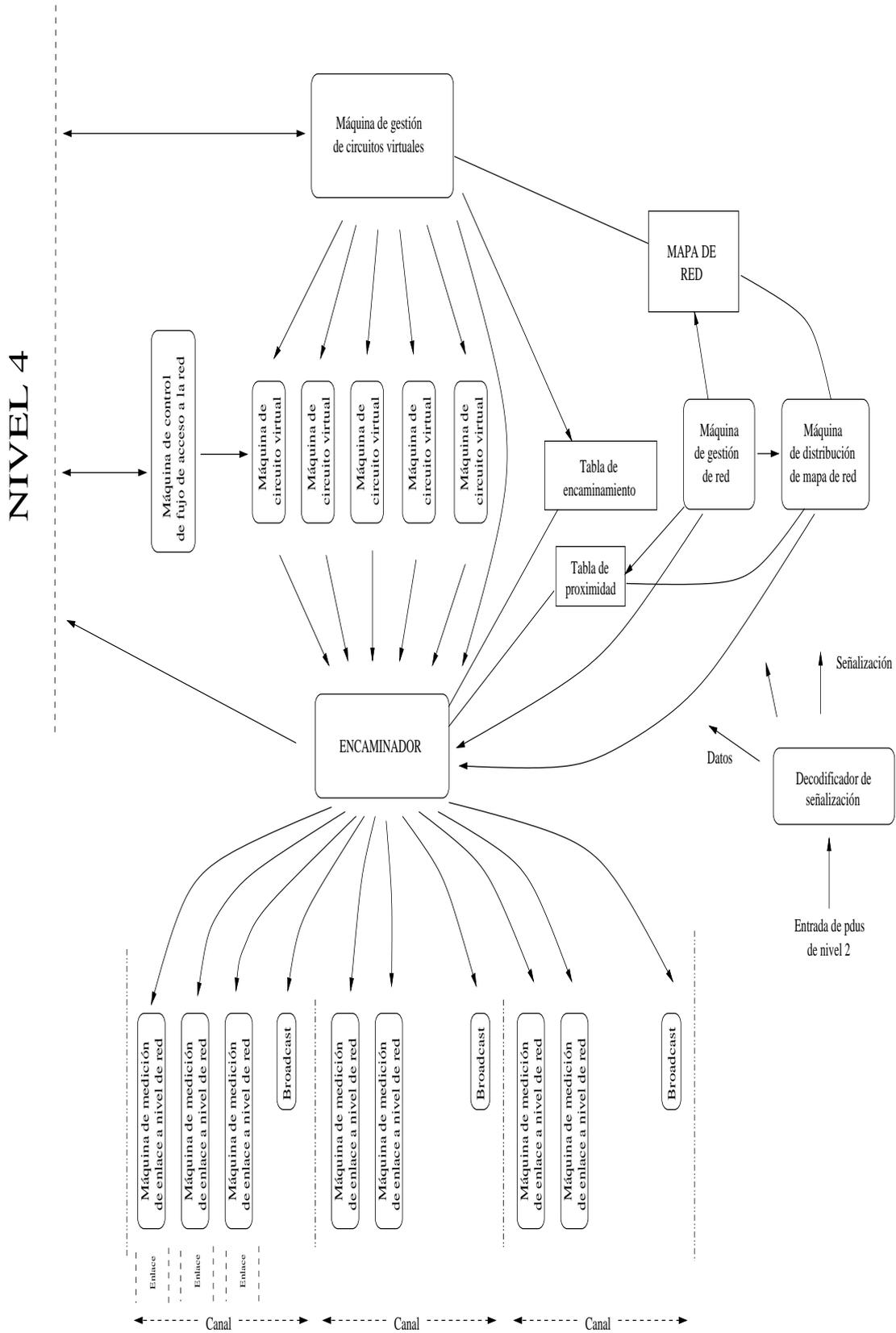


Figura 4.1: Modelo general del nodo

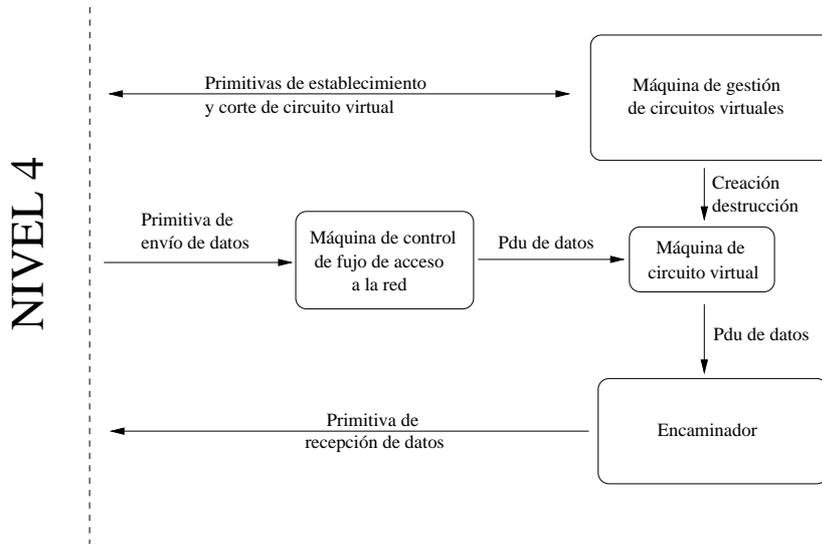


Figura 4.2: Subsistema de acceso a la red

### 4.3 Encaminador

En función de una tabla de encaminamiento, el encaminador decide por que camino enviar las pdus de datos o de señalización de los demás sistemas.

#### Formato de la tabla de encaminamiento

Id de C.V.	Sentido (ida/vuelta)	Canal de salida	Enlace de salida	Canal de entrada	Enlace de entrada	Usuario local

Tabla 4.1: Tabla de encaminamiento

La tabla de encaminamiento es una base de datos que tiene una entrada por cada sentido de un circuito virtual. Su objetivo es indicar por qué enlaces un circuito virtual entra y sale del nodo. Como veremos en el capítulo 7, el encaminamiento se decide al origen, por lo tanto *los elementos son creados y borrados por la máquina de gestión de circuitos virtuales* en el momento del establecimiento y cerrado de circuitos virtuales.

Como podemos observar en la tabla 4.1, cada elemento se compone de:

**Identificador de circuito virtual** El identificador del circuito virtual cuya ruta se indica.

**Sentido** Indica si este elemento se refiere al circuito virtual de ida o de vuelta.

**Canal de salida** Es posible que el nodo tenga acceso a varios puertos (o canales). Indica en qué puerto se encuentra el enlace utilizado por el circuito virtual para su salida del nodo.

**Enlace de salida** Enlace utilizado por el circuito virtual para su salida del nodo.

**Canal de entrada** Indica en qué puerto se encuentra el enlace utilizado por el circuito virtual para su entrada al nodo.

**Enlace de entrada** Enlace utilizado por el circuito virtual para su entrada al nodo.

**Usuario local** Contiene la referencia del usuario local al que deben ser entregadas las pdus (si el nodo no es el nodo destino del circuito virtual, esta campo está vacío).

## Encaminador

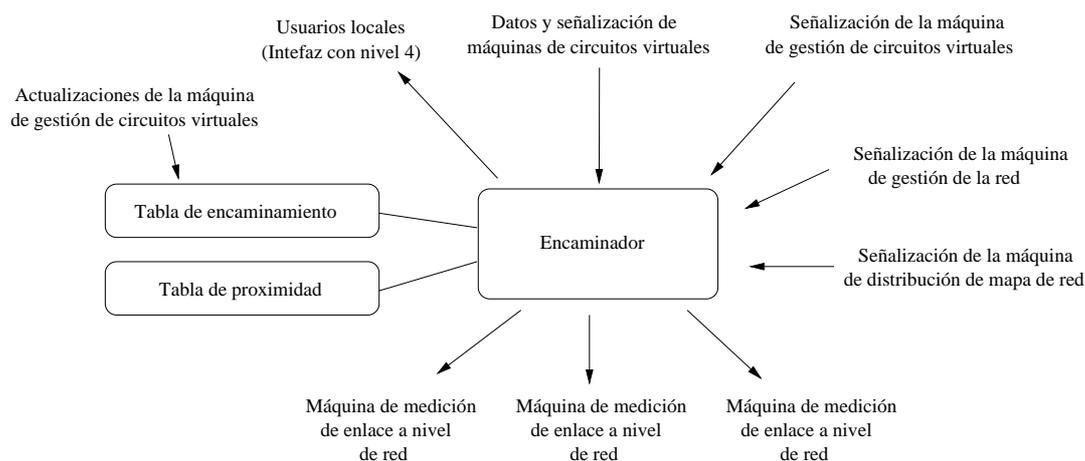


Figura 4.3: Máquina de encaminamiento

El encaminador es un punto de cruce que envía la información al lugar apuntado por la tabla de encaminamiento (exclusivamente a usuarios locales o a los canales radioeléctricos). Su localización es central, es el *distribuidor de salida*. En la figura 4.3 podemos observar sus conexiones, de las que destacan:

**Tabla de proximidad** El formato y descripción de esta tabla se describe en el apartado de *máquina de gestión de red* (página 29). Mediante esta tabla, el encaminador es capaz de mandar datos a nodos concretos, sin utilizar circuitos virtuales. Entre otros datos, en esta tabla figura la relación de convergencia nivel3-nivel2, es decir qué enlace (dirección de enlace AX.25 y canal) se debe de utilizar para alcanzar un nodo con una determinada dirección de red.

**Máquinas de medición de enlace a nivel de red** Existe una máquina de medición de enlace por cada enlace. Es el único punto de acceso para envío de datos a través de un enlace dado, al fin de que todo lo transmitido pueda ser medido. El encaminador decide a qué enlace enviar las pds (de datos o señalización) en función del circuito virtual, consultando la tabla de encaminamiento.

**Señalización de la máquina de gestión de circuitos virtuales** Los mensajes de apertura y de cerrado de circuitos virtuales son enviados al nodo vecino a través del encaminador, que ya conoce el enlace a utilizar puesto que la propia máquina de gestión de circuitos virtuales ha actualizado la tabla de encaminamiento con este dato.

## 4.4 Máquina de circuito virtual

Existe una máquina de circuito virtual para cada circuito virtual que pasa por el nodo, creada y destruida por la máquina de gestión de circuitos virtuales. El control de flujo interno es su principal función<sup>1</sup>, tiene capacidad de bloquear el tráfico del circuito virtual, e intercambiar señalización con los nodos de su circuito virtual.

En la figura 4.4 se puede observar su conexionado con los demás elementos. Cabe

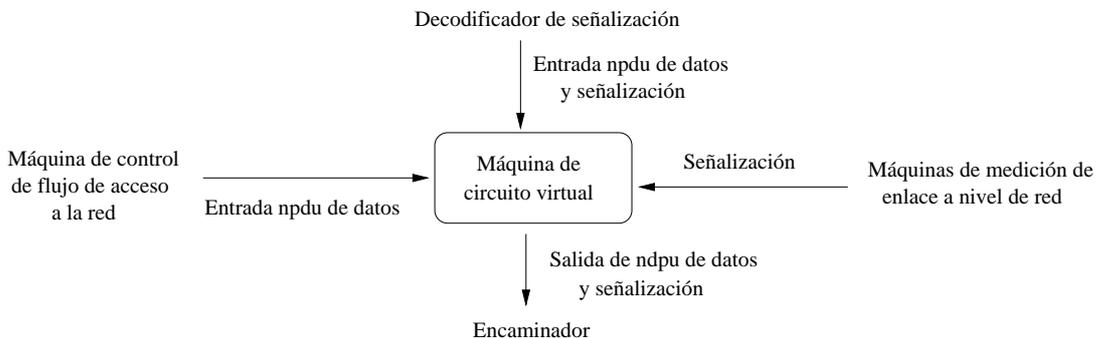


Figura 4.4: Máquina de circuito virtual

destacar la llegada de señalización de las máquinas de medición de enlace a nivel de red, que miden el nivel de congestión de cada enlace; con esta información, la máquina de circuito virtual toma decisiones respecto a las acciones de control de flujo.

<sup>1</sup>Ver el apartado 6.2, control de flujo interno.

## 4.5 Máquina de gestión de circuitos virtuales

Sus funciones son:

- *Creación* de los circuitos virtuales demandados por los usuarios locales, decidiendo su camino, dado el mapa de la red, comprobando que es viable el proceso, y que se lleva a cabo correctamente en el resto del circuito virtual.
- Recepción y propagación de las *señales de establecimiento* de circuito virtual proveniente de nodos vecinos.
- Añadir la nueva ruta a la tabla de encaminamiento.
- *Reserva de los recursos* necesarios para los circuitos virtuales que pasan por el nodo; especialmente, *creará una máquina de circuito virtual* para cada circuito virtual dedicada a la gestión de este circuito virtual particular.
- *Destrucción* de los circuitos virtuales cerrados por los usuarios locales, comprobando que se lleva a cabo correctamente en el resto del circuito virtual.
- Recepción y propagación de las señales de *cerrado* de circuito virtual.
- Borrar la ruta de la tabla de encaminamiento.
- Destruir los recursos ocupados, *destruir la máquina de circuito virtual*.

En la figura 4.5 podemos observar como se interconexiona para llevar a cabo estas tareas.

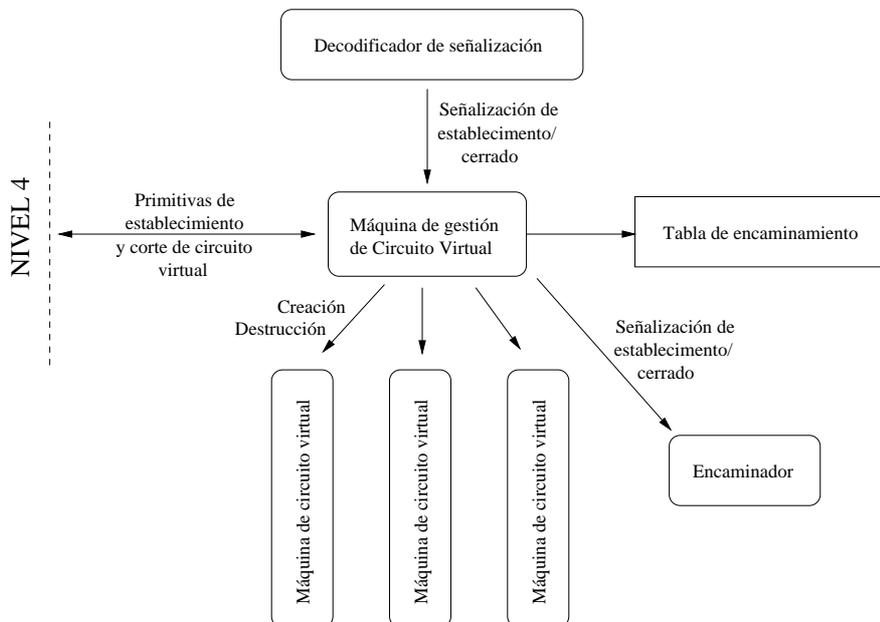


Figura 4.5: Máquina de gestión de circuitos virtuales

## 4.6 Máquinas de medición de enlaces a nivel de red

Como ya se ha dicho, cada enlace tiene asignado una máquina de medición de enlace. Toda la información mandada por un enlace debe pasar por su respectiva máquina de medición, siendo esta el único punto de acceso al enlace. En el capítulo 5 se abordarán en detalle los procedimientos de medición.

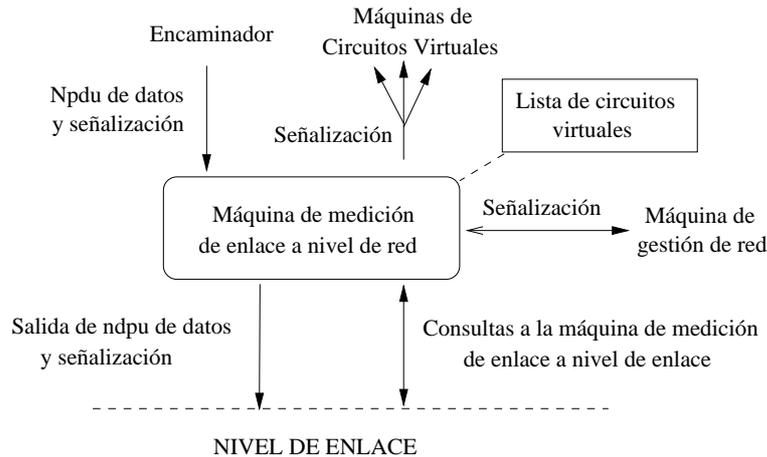


Figura 4.6: Máquinas de medición de enlaces a nivel de red

En la figura 4.6 se pueden observar las interacciones con los demás elementos, de los que cabe destacar:

**Máquina de medición de enlace a nivel enlace** Como se describe en el capítulo de mediciones (capítulo 5), se ha diseñado una máquina de mediciones cuyo marco de funcionamiento es interno al nivel 2, y realizando mediciones a bajo nivel.

**Máquina de gestión de red** Realiza consultas sobre magnitudes medidas: capacidad máxima del enlace, tasa binaria cursada. También le son comunicadas ciertas incidencias como la rotura o restablecimientos de enlaces.

**Máquinas de circuitos virtuales** Son informadas del nivel de congestión de sus enlaces. Para esta tarea, la máquina de medición de enlace a nivel de red debe tener una *lista en la que conste todos los circuitos virtuales que utilizan su enlace*. Ver apartado 6.2.15.

## 4.7 Máquina de gestión de red

La función de la máquina de gestión de red es crear y actualizar el mapa de la red y la tabla de proximidad; debe recibir la señalización de topología de red de los nodos vecinos, detectar nodos adyacentes, obtener y calcular los parámetros primarios de los enlaces de este nodo.

### 4.7.1 Tabla de proximidad

Dirección de red	Nº	Cj <sub>p</sub>	Fj <sub>p</sub>	Hj <sub>p</sub>	Pj <sub>p</sub>	Sentido	Tipo	Enlace	Canal	Estado

Tabla 4.2: Tabla de proximidad

La tabla de proximidad recoge toda la información que caracteriza el entorno del nodo: los nodos adyacentes, y los enlaces para llegar a ellos. Como podemos observar en la tabla 4.2, cada elemento (fila) de la tabla describe un enlace, con sus características (columnas), que son las siguientes:

**Dirección de red** Es la dirección de red del nodo al que apunta esta enlace.

**Número** Se trata de un identificador (número entero) que permite diferenciar enlaces que apuntan al mismo nodo pero a través de canales diferentes. Es “0” cuando no existe más de un enlace a este nodo, y se va incrementando a medida que aparecen enlaces alternativos para este mismo nodo.

**Cj<sub>p</sub>** Parámetro primario Cj<sub>p</sub> (capacidad máxima de este enlace en bits por segundo).

**Fj<sub>p</sub>** Parámetro primario Fj<sub>p</sub> (tasa binaria de la información transportada por el enlace, en bits por segundo)

**Hj<sub>p</sub>** Parámetro primario Hj<sub>p</sub> (el ancho de banda de canal no utilizado)

**Pj<sub>p</sub>** Parámetro primario Pj<sub>p</sub> (probabilidad de no error de paquete del enlace). No tiene sentido para enlaces con conexión hdlc.

**Sentido** Ida o vuelta. Los enlace hacia y desde un nodo no tienen las mismas características, y son diferenciadas mediante este indicador.

**Tipo** Especifica si se trata de un enlace *con* conexión hdlc o *sin* conexión hdlc.

**Canal** Canal dónde se encuentra el enlace a este nodo.

**Enlace** Dirección de enlace (ax.25) para alcanzar el nodo.

**Estado** Estado de la conexión (abierta o cerrada). Solo tiene sentido en enlaces con conexión.

### 4.7.2 Mapa de red

El mapa de red describe la topología de la red, haciendo abstracción de los niveles inferiores. Es un grafo en el que se describen los enlaces con:

1. Pareja (dirección de red de origen, dirección de red de destino)
2. Tipo de enlace (con o sin conexión hdlc)
3. Parámetros característicos ( $C_j$ ,  $F_j$ ,  $P_j$ )
4. Identificador de enlace: se trata del campo que se llama “número” en la tabla de proximidad. Sólo se utiliza cuando existen varios enlaces entre dos nodos.

### 4.7.3 Comunicación con otros elementos del nodo

En la figura 4.7 podemos ver los elementos con los que se comunica la máquina de gestión de red.

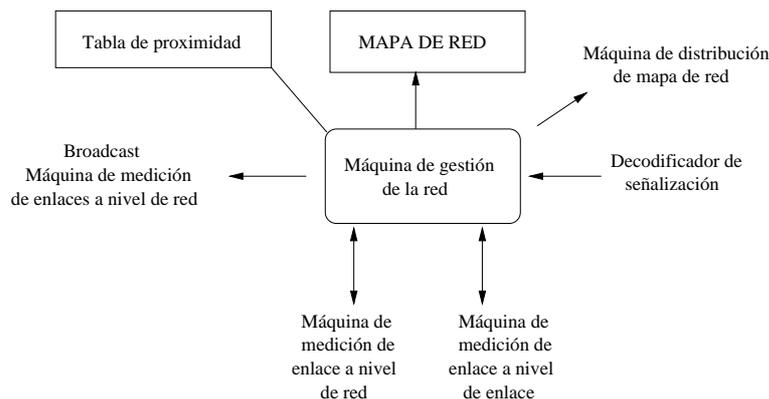


Figura 4.7: Máquina de gestión de red

Cabe destacar:

**Decodificador de señalización** Los mensajes enviados al canal por las máquinas de gestión de red y máquinas de distribución de mapa de red de otros nodos son recibidos por esta vía. Se reciben señalizaciones de detección de nodos adyacentes, señalización de medición de probabilidad de no error de paquete, y de mapas de red (enviado por máquinas de distribución de mapa de red).

**Broadcast** Se trata de un módulo que se encuentra en cada canal, junto con las máquinas de medición de enlaces a nivel de red, que permite difundir mensajes de señalización en el canal, usando tramas no numeradas dirigidas a un dirección de enlace predefinida para que los decodificadores de señalización de los nodos a la escucha sepan decodificarlas (como se indica en el punto anterior). Esta señalización se utiliza para mediciones de probabilidad de no error de paquete

(como describiremos en el capítulo de mediciones, capítulo 5), y para que los nodos adyacentes nos detecten.

**Máquinas de medición de enlace a nivel de red** Se reciben por esta vía mensajes que señalizan la rotura o restablecimientos de enlaces, y se realizan consultas sobre los parámetros primarios  $C_{j_p}$  y  $F_{j_p}$ .

**Máquinas de medición de enlace a nivel de enlace** Se realizan consultas a esta máquina con el objetivo de conocer el número de paquetes recibidos de una cierta estación, y el número de paquetes que hemos transmitido, el factor de ocupación del canal, el ancho de banda de canal no utilizado, datos relevantes en el cálculo de la probabilidad de no error de paquete  $P_j$ .

**Máquina de distribución de mapa de red** La máquina de gestión de la red detecta los cambios de topología y de parámetros de los enlaces locales, que plasma en el mapa de la red. Pero para una rápida distribución de estos cambios, la máquina de distribución de mapa de red es avisada inmediatamente mediante señalización interna.

## 4.8 Máquina de distribución del mapa de red

Este sistema es el de conexionado más sencillo : lee el mapa de red y la tabla de proximidad, es avisado de los cambios por la máquina de gestión de red. Propaga las actualizaciones a los nodos vecinos utilizando un formato definido en el protocolo de encaminamiento.

No obstante, a pesar de esta sencillez aparente, debe realizar una tarea compleja: controlar la difusión de la información de la topología de red, optimizando los recursos de la red y manteniendo la adecuada velocidad de propagación de esta información. Es pues una tarea de control de flujo y de encaminamiento, pero a nivel de difusión de información, no de comunicaciones entre dos nodos dados de la red. Básicamente deberá regular los siguientes aspectos:

- Velocidad a la que se envían las actualizaciones.
- Metodología de caché.<sup>2</sup>
- Política de prioridades.
- Encaminamiento de la difusión.

Debido a estas características especiales, no es evidente aplicar los resultados del estudio de control de flujo y encaminamiento de información de usuario, siendo incluso probable que el modelo de datagramas se ajuste mejor a este caso. Es otro plano que

---

<sup>2</sup>Almacenamiento en memoria de datos recibidos, y actualizaciones de estos mismos, para su posterior selectiva retransmisión.

requiere otro sistema de encaminamiento, pero que por razones de eficiencia deberá basarse en la topología ya obtenida en el mapa de la red.

Este capítulo no es abordado en el proyecto. No obstante es una parte importante, que deberá ser investigada en el futuro.

## 4.9 Decodificador de señalización

El decodificador de señalización compone la otra cara de la moneda: es el punto de entrada de todos los datos externos recibidos a través del nivel de enlace.

Si se recibe una n-pdu de datos, debe ser entregada a la máquina de circuito virtual a la que pertenece. Si se trata de una n-pdu de señalización, esta debe ser decodificada y entregada al módulo que deberá procesarla. Recapitulando los anteriores apartados, podemos realizar el siguiente reparto de señalización:

Destino de la señalización	Tipo de señalización	Origen de la señalización
Máquinas de circuitos virtuales	Control de flujo interno	Máquinas de circuitos virtuales
Máquina de gestión de circuitos virtuales	Establecimiento/corte de circuitos virtuales	Máquina de gestión de circuitos virtuales
Máquina de gestión de red	Actualización de mapa de red	Máquina de distribución de mapa de red
Máquina de gestión de red	Probabilidad de no error de paquete, detección de adyacencias	Máquina de gestión de red

# Capítulo 5

## Mediciones

### 5.1 Modificaciones de AX.25 v2.2

Algunas de las máquinas y algoritmos expuestos a continuación suponen modificaciones de la especificación original AX.25 versión 2.2. Se ha querido evitar al máximo esta práctica; en ningún caso se han aplicado “parches” o mejoras superficiales que perjudicarían la actual especificación, si no que se ha mejorado, completando partes incompletas y añadiendo funcionalidades, incorporando nuevas máquinas de estado que no perjudican ni alteran el funcionamiento del resto de las máquinas.

Resulta por ejemplo llamativo que las primitivas descritas en ax.25 v2.2 no tengan parámetros (valores transportados por las primitivas). Ello se debe a que estos parámetros no son estrictamente necesarios para la especificación, son implícitos. Pero en las nuevas máquinas introducidas, en algunos casos estos parámetros son imprescindibles, y por lo tanto se mencionan y describen explícitamente.

La inserción de nuevas máquinas hace necesaria la aparición de nuevas señales, que son generadas o recibidas por las máquinas actualmente implementadas en la norma. Es por lo tanto necesario modificar también estas máquinas, para que generen y reciban estas señales. Estas modificaciones no suponen grandes cambios, ni suponen ninguna tarea adicional.

### 5.2 Máquina de medición de enlace a nivel de enlace - MEASUREMENT MACHINE

#### 5.2.1 Ubicación

El modelo de AX.25 v2.2 divide el nivel de enlace en máquinas de estados finitos (máquina de enlace, segmentador, gestor de enlace, multiplexor de enlace), como se puede observar en la figura 5.1. La función de cada máquina se describe en “AX.25 Link Access Protocol for Amateur Radio, version 2.2”[3].

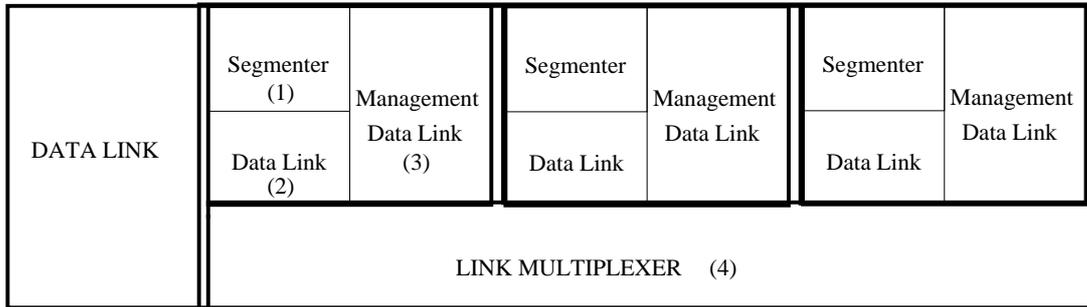


Figura 5.1: Máquinas de ax25 v2.2

Las máquinas de enlace de datos (2), segmentador (1), gestor de enlace (3), son específicas para cada enlace. Se crean cada vez que se crea un enlace ax.25 con otra estación. En cambio, el multiplexor (4) es único, dando servicio a todas las máquinas que se encuentran sobre él.

Existen mediciones de interés para el nivel de red que no pueden ser medidas a nivel de red: no se puede conocer el número de tramas ax.25 transmitidas porque el nivel de enlace realiza retransmisiones invisibles al nivel de red, no se pueden conocer las estadísticas de las estaciones recibidas porque el nivel de enlace no informa al nivel de red sobre tramas dirigidas a otras estaciones, y tampoco es posible saber si el canal tiene portadora de datos. En el estudio posterior, estas mediciones resultan útiles para obtener parámetros del canal, como la probabilidad de no error de paquete y el factor de ocupación del canal.

Esto nos lleva a ampliar la actual especificación ax.25 v2.2, añadiéndole estas facilidades de medición. Para ello vamos diseñar una máquina de medición, que tendrá por objetivo medir tres magnitudes:

- Número de paquetes transmitidos por cada máquina de enlace.
- Estaciones recibidas y número de paquetes de estas estaciones.
- Porcentaje de tiempo que el canal está ocupado con portadora de datos.

La segunda magnitud no puede medirse en máquinas específicas de cada enlace establecidos, ya que no tiene porque existir enlace establecido con todas las estaciones oídas (y por lo tanto no existen las máquinas correspondientes). Igualmente, no tiene sentido medir la tercera magnitud en las máquinas específicas de cada enlace. La primera magnitud puede medirse en las máquinas específicas de cada enlace o a nivel del multiplexor; la máquina de enlace envía los paquetes al multiplexor (mediante las primitivas LM-EXPEDITED-DATA Request o LM-DATA Request).

Para realizar una única máquina, la máquina de mediciones a nivel de enlace se va a encontrar al mismo nivel que el multiplexor. Sólo habrá una máquina de mediciones por canal físico, que medirá las magnitudes referidas a este canal. En la figura 5.2 vemos la ubicación de esta máquina, que bautizamos Measurement Machine (5) (o máquina de medición de enlace a nivel de enlace).

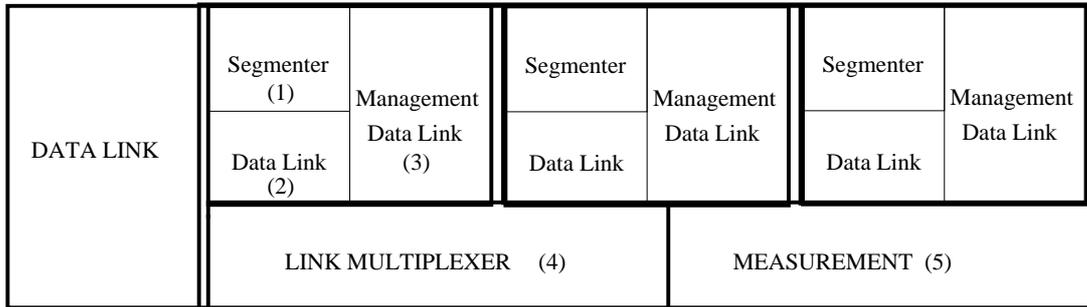


Figura 5.2: Ubicación de la máquina de medidas a nivel de enlace

### 5.2.2 Especificación

La siguiente especificación de Measurement Machine se hace siguiendo la estructura seguida en ax25 v2.2.

#### 5.2.2.1 Interacción con la máquina de estados de la capa física

**PH-BUSY Indication** Indicación de que el nivel físico detecta la presencia de una portadora de datos en el canal.

**PH-QUIET Indication** Indicación de que el nivel físico ya no detecta la presencia de la portadora de datos en el canal.

#### 5.2.2.2 Interacción con el multiplexor

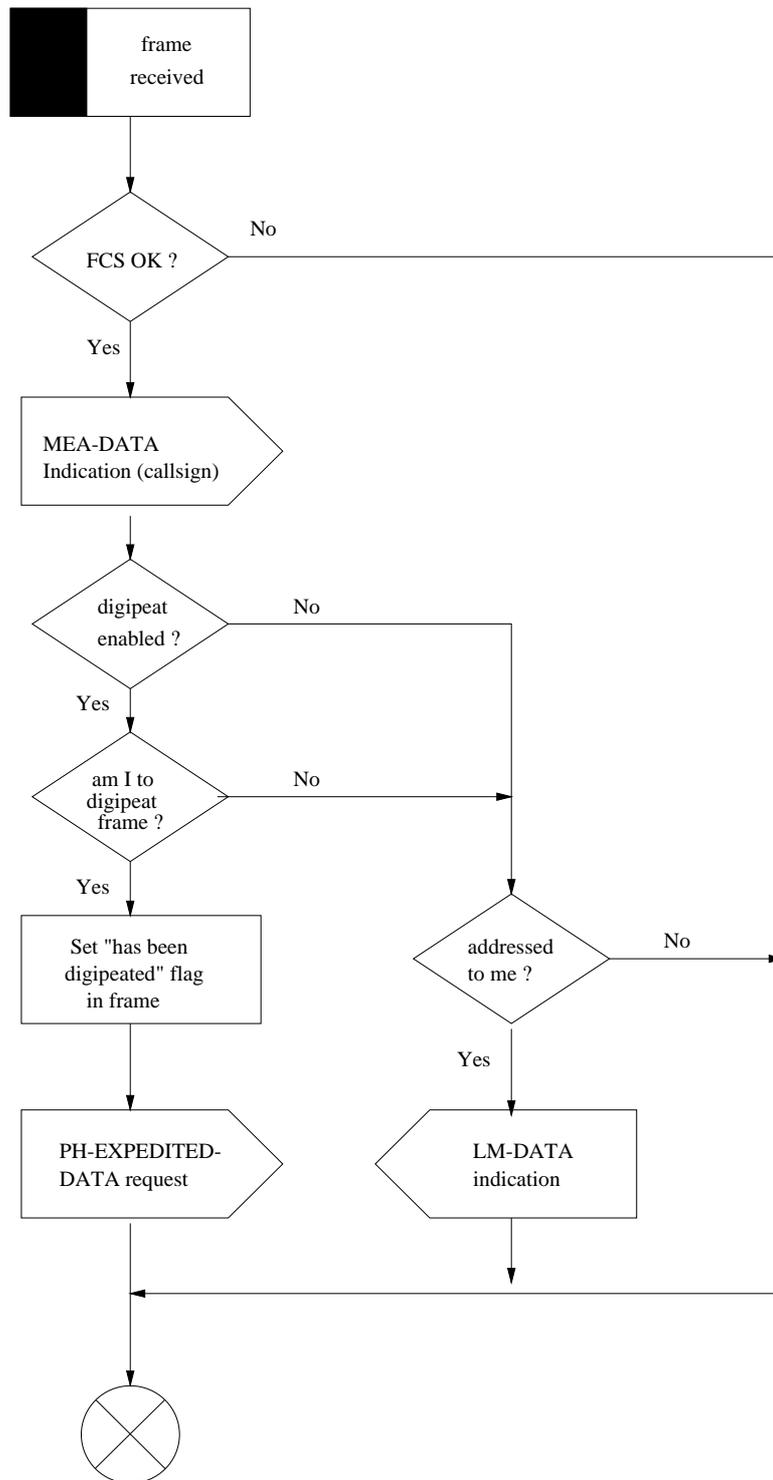
**MEA-DATA Indication(Callsign)** El multiplexor envía esta primitiva a la máquina de medición cuándo recibe un paquete libre de errores. El parámetro de esta primitiva es la dirección de nivel de enlace del paquete recibido.

Se modifica la máquina de estados del multiplexor, en concreto la subrutina “Frame received”, página 78 de la norma, tal y como puede observarse en la figura 5.3. El cambio es muy sencillo: se añade la generación de la señal MEA-DATA Indication entre las cajas de test “FCS Ok?” y “digipeat enabled?”, punto en el que se sabe que se ha recibido una trama AX.25 correcta (sin error de FCS). El multiplexor debe de todas las formas descodificar el campo destino de la trama AX.25, para saber si debe mandarla al nivel superior.

#### 5.2.2.3 Interacción con máquina de enlace

**LM-EXPEDITED-DATA Request** Envío de tramas no numeradas.

**LM-DATA Request** Envío de tramas numeradas.



Note: The LM-DATA indication is sent to the data link machine which is responsible for communications with the indicated remote (source) station.

Figura 5.3: Modificación del multiplexor - rutina frame received

La máquina de enlace de datos envía paquetes enviando estas dos primitivas al multiplexor<sup>1</sup>. La máquina de medición será sensible a estas primitivas, y contabilizará el número de paquetes transmitidos, y con qué dirección de enlace (cada máquina de enlace tiene una dirección de enlace diferente).

#### 5.2.2.4 Interacción con nivel de red

Las siguientes primitivas permiten al nivel de red realizar consultas a la máquina de mediciones:

**MEA-HEARD Request(Callsign)** Petición de cuantos paquetes provenientes de la dirección de enlace Callsign han sido oídos, y cuanto tiempo ha transcurrido desde el último paquete.

**MEA-HEARD Confirm(N,T)** Respuesta a la consulta anterior. N es el número de paquetes, T es el tiempo transcurrido. En caso de no estar Callsign en la Heard-List, se retorna N=0, T=0.

**MEA-SENT Request(Callsign)** Petición de cuantos paquetes con dirección de origen Callsign han sido transmitidos por el nodo.

**MEA-SENT Confirm(N)** Respuesta a la consulta anterior. N es el número de paquetes. En caso de no estar Callsign en la SentList, se retorna N=0.

**MEA-OCCUPATION Request** Petición del porcentaje de tiempo que el canal está ocupado (con portadora de datos), que llamamos “porcentaje de ocupación del canal”.

**MEA-OCCUPATION Confirm(OCi)** Respuesta a la consulta anterior.

#### 5.2.2.5 Operación interna de la máquina

**Primitivas MEA** (recibidas del nivel 3)

MEA-HEARD Request(Callsign)

MEA-SENT Request(Callsign)

MEA-OCCUPATION Request

**Primitivas MEA** (mandadas al nivel 3)

MEA-HEARD Confirm(N,T)

MEA-SENT Confirm(N)

MEA-OCCUPATION Confirm(OCi)

---

<sup>1</sup>No queda claramente expuesto en la máquina de estados de la máquina de enlaces (norma AX.25 v2.2), se sobreentiende que las señales con nombres de tipo de paquetes AX.25 (Información, control, no numerados) se mandan mediante estas primitivas.

**Primitivas MEA** (recibidas del multiplexor)  
 MEA-DATA Indication(Callsign)

**Primitivas LM** (recibidas de la máquina de enlace)  
 LM-EXPEDITED-DATA Request  
 LM-DATA Request

**Estados** 1 estado.

**Colas:** Ninguna.

**Códigos de error:** Ninguno.

**Variables:**

**Heard List** Lista de estaciones oídas. Un elemento por dirección AX.25 oída. Cada elemento se compone de 3 campos:  
 CallsignRx (Dirección AX.25) ; Nrx (número de paquetes recibidos) ;  
 LastHeard (instante de recepción del último paquete).  
 Los dos últimos campos se refieren a esta dirección AX.25 en particular.

**Sent List** Lista de indicativos de las máquinas de enlace del nodo que han transmitido paquetes. Cada elemento se compone de 2 campos:  
 CallsignTx (Dirección AX.25) ; Ntx (número de paquetes transmitidos por este indicativo).

*Las magnitudes de cuantificación del número de paquetes (transmitidos o recibidos) deben ser codificadas de forma que su incremento pueda ser calculado unívocamente. Por ejemplo, en caso de utilizar una codificación que tiene un límite superior<sup>2</sup>, al alcanzar este valor el contador puede ser puesto a cero y seguir incrementándose: el incremento real se deduce sin ambigüedad, siendo este:*

$$\text{valor anterior} - \text{valor máximo} + \text{valor actual}$$

**Occupation List** Lista de intervalos durante los cuales ha habido una portadora de datos en el canal (“ocupación del canal”). Un intervalo de ocupación de canal empieza en el momento en el que se recibe la primitiva PH-BUSY, y se acaba al recibir PH-QUIET. Cada elemento de la lista se corresponde con un intervalo de ocupación del canal, y se compone de 2 campos:  
 ti (instante de finalización del intervalo); Ti (duración del intervalo).  
 En la figura 5.4 se presenta una esquema resumen de la Occupation List.

---

<sup>2</sup>Es lo habitual en las variables utilizadas en los lenguajes de programación, ya que son codificadas con un numero fijo de bits. Por ejemplo un tipo “integer” codificado con dos octetos no puede superar el valor 65535.

- TBUSY Instante en el que se recibió el último PH-BUSY (principio de un intervalo de ocupación de canal).
- OC Último valor del porcentaje de ocupación del canal calculado.

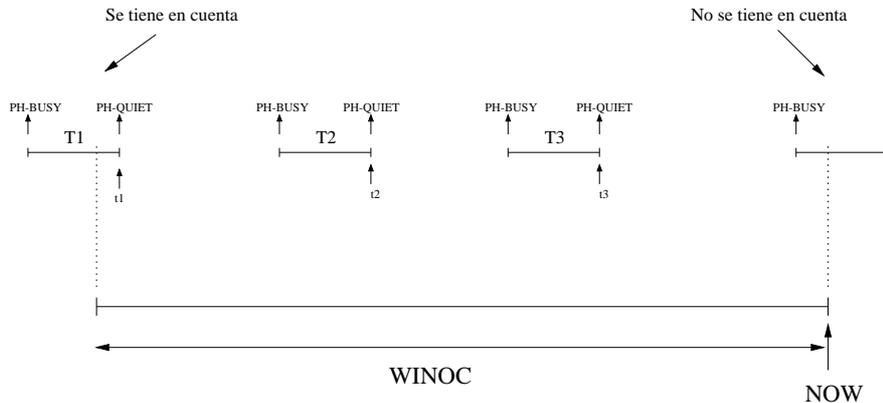


Figura 5.4: Occupation List

### Constantes

- THeard Los indicadores cuyo último paquete no se ha vuelto a oír desde hace más tiempo que THeard son borrados de la Heard List.
- WINOC Duración de la ventana tiempo durante la cual se realiza la medición de la ocupación del canal; es decir que es el intervalo de tiempo en el que se observa el canal para obtener OC.

### Temporizadores:

- THClear Periodo de limpieza de Heard List.
- TOC Periodo de medición de ocupación y limpieza de la Occupation List.

**Consejos:** Se corre el peligro de que la Occupation List se haga muy grande debido a oscilaciones y pérdidas de sincronismo de los dispositivos detectores de portadora de datos. Se aconseja definir un límite de tamaño, y si este límite se supera:

- Dividir la ventana WINOC en dos ventanas de tiempo:
  - WINOC1: empieza en el instante  $NOW - WINOC$ , y termina en  $NOW - WINOC + WINOC1$ . Es el intervalo de tiempo justo para que la Occupation List alcance su máxima capacidad.

- WINOC2: el resto del intervalo. Empieza en el instante  $NOW - WINOC + WINOC1$  y termina en el instante  $NOW$ .
- Calcular la ocupación dentro del intervalo WINOC1:  
$$OC = \min\left(1, \frac{\sum_{T_i \in WINOC1} T_i}{WINOC1}\right)$$
- No volver a calcular OC y no almacenar entradas hasta que WINOC no contenga el intervalo actual WINOC2. .

Cuánto mayor sea WINOC, más cálculos se hacen (peligro de saturación del nodo por falta de velocidad de procesamiento) y mayor es la Occupation List (peligro de saturación de la memoria). Por lo tanto WINOC debe ser elegido cuidadosamente en función de la velocidad y las características del canal.

#### 5.2.2.6 Especificación sdl

En las figuras 5.5 y 5.6 se puede ver la especificación sdl de la máquina de mediciones.

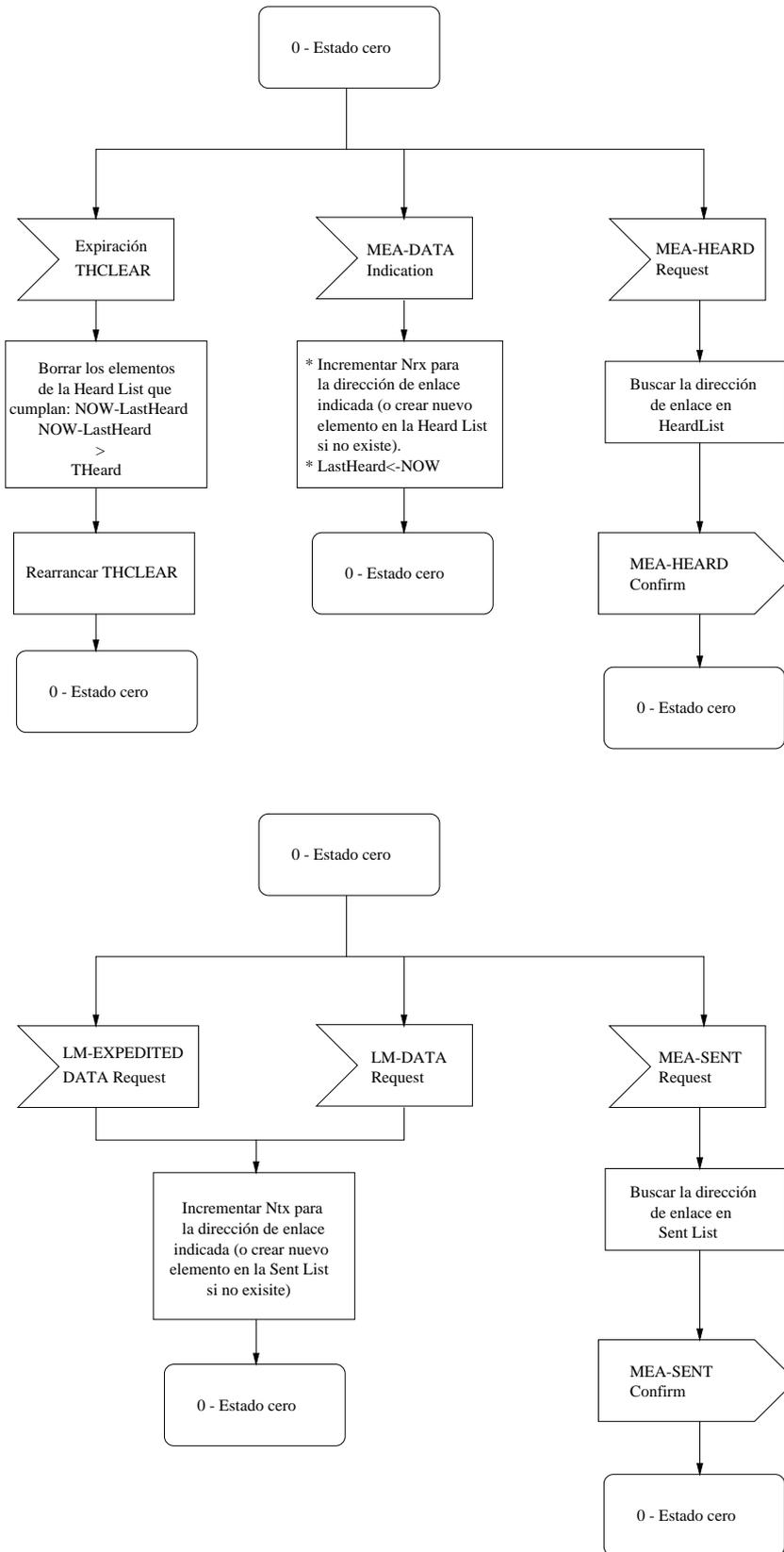


Figura 5.5: Especificación SDL de MEASUREMENT MACHINE

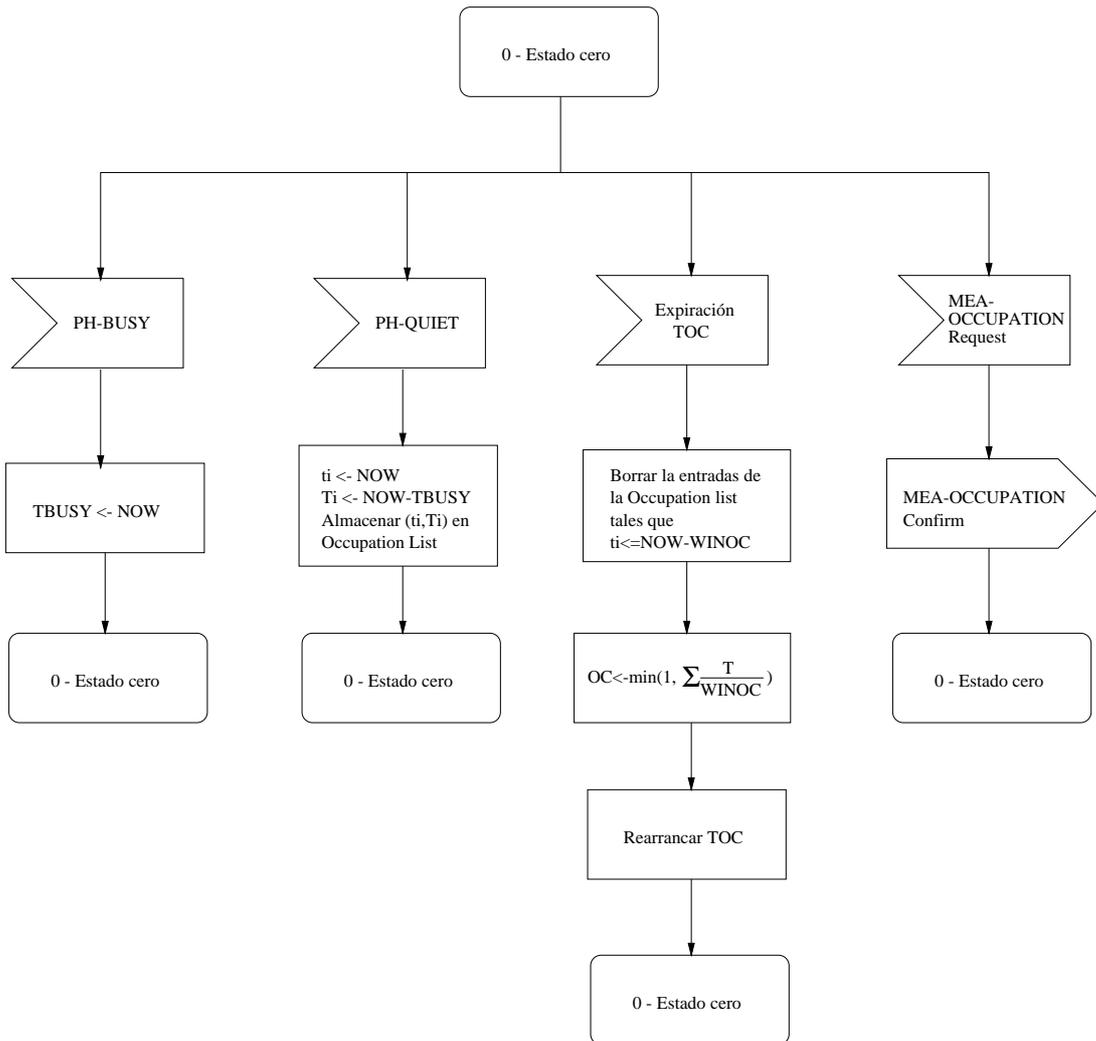


Figura 5.6: Especificación SDL de MEASUREMENT MACHINE - Parte de ocupación de canal.

## 5.3 Ampliación de AX.25 v2.2 para DL-DATA Confirm

La primitiva DL-DATA Confirm asiente el envío de las primitivas DL-DATA Request, indicando la pdu ha sido entregada correctamente a la estación con la que el nivel de enlace ha realizado una conexión.

En la actual norma AX.25 v2.2, no está implementado DL-DATA Confirm, a pesar de ser especificado en el capítulo de intercambio de primitivas (página 130 de la norma). En este capítulo analizamos esta carencia y proponemos una forma de solucionarla.

### 5.3.1 Funcionamiento y carencias del fragmentado en AX.25 v2.2

La dificultad de implementación del proceso generador de la primitiva DL-DATA Confirm radica en el funcionamiento del Segmentador.

El Segmentador actúa como un “filtro” entre el nivel de red y la máquina de enlace, interrumpiendo toda primitiva DL-DATA Request cuyo campo de información supera un tamaño determinado y generando varias primitivas DL-DATA Request de menor tamaño (segmentos).

La máquina de enlace de datos no detecta en absoluto este proceso, desconoce a qué pdu original pertenecen los segmento transmitidos, incluso desconoce si está transmitiendo segmentos o paquetes sin segmentar. Pero sólo ella sabe qué pdus son asentidas y cuáles no, ella gestiona el proceso de retransmisiones y movimiento de la ventana de transmisión hdlc. *Para implementar DL-DATA Confirm, la máquina de enlace de datos debe realizar un seguimiento de las pdus transmitidas y asentidas.*

Por otro lado, pueden haber varias primitivas DL-DATA Request pendientes de asentir: es necesario que las primitivas se diferencien con un identificador, para poder ser asentidas independientemente; *DL-DATA Request y DL-DATA Confirm tendrán como parámetro un identificador.*

### 5.3.2 Cambio de nombre de las primitivas

En el proceso de segmentación, el segmentador a partir de una primitiva con un determinado identificador debe generar otras primitivas (fragmentos) y asignarles otros identificadores. Además, es el segmentador quién tiene que generar el asentimiento DL-DATA Confirm de primitivas DL-DATA Request segmentadas; este asentimiento sólo se producirá cuando todos los fragmentos hayan sido asentidos por la máquina de enlaces. La operativa del segmentador se complica, y *para obtener mayor claridad se propone cambiar el nombre de las primitivas*, de la forma siguiente:

1. Todas las primitivas de transmisión y recepción de datos entre el nivel de red y el segmentador pasan del nombre DL-DATA al nombre SEG-DATA<sup>3</sup>.

---

<sup>3</sup>Esto incluye a la primitiva DL-DATA Confirm. Por lo tanto en toda rigurosidad la sección hubiera tenido que titularse *Ampliación de AX.25 v2.2 para SEG-DATA Confirm*.

2. Todas las primitivas de transmisión y recepción de datos entre el segmentador y la máquina de enlace empiezan su nombre por DL-DATA.

El punto 1 significa el siguiente cambio:

Antes	Ahora
DL-DATA Request	SEG-DATA Request (Id)
DL-DATA Indication	SEG-DATA Indication (Id)
DL-UNIT-DATA Indication	SEG-UNIT-DATA Indication
DL-UNIT-DATA Request	SEG-UNIT-DATA Request
DL-DATA Confirm	SEG-DATA Confirm (Id)

Con (Id) queremos destacar que ahora las primitivas SEG-DATA tienen un identificador explícito. Las primitivas SEG-UNIT-DATA no necesitan identificador, porque no son asentidas (envío no orientado a conexión), pero sí pueden ser segmentadas y por este motivo también se cambia su notación.

Estos cambios deben aplicarse en las descripciones SDL del segmentador, por simple sustitución del nombre de las señales, en las páginas 122, 123, 124 y 125 de la norma<sup>4</sup>.

El resumen del intercambio de primitivas se presenta en la figura 5.7. La notación Id e Id' se utiliza para destacar que los identificadores usados son independientes.

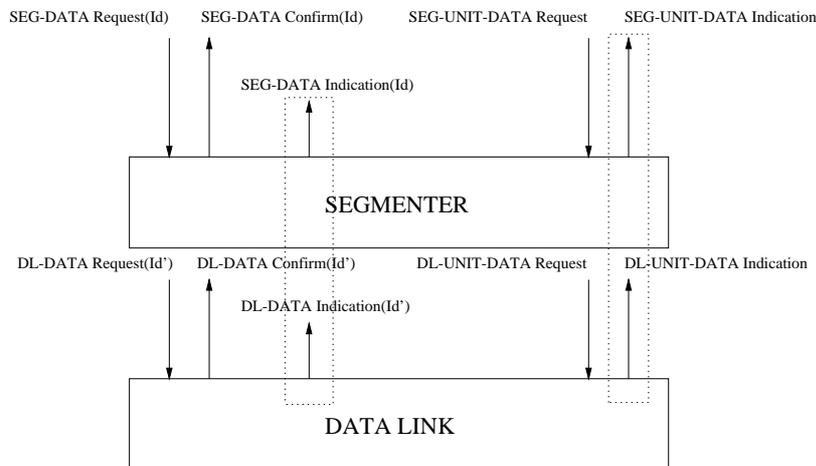


Figura 5.7: Esquema de las nuevas primitivas SEG

No se ha mencionado anteriormente por no complicar las descripciones, pero realmente el segmentador se divide en dos partes: la parte de segmentación y la parte de reensamblado. Las primitivas generadas y recibidas por la parte de reensamblado están rodeadas por una línea punteada en la figura 5.7. En la norma AX25 v2.2 esta parte tiene su propia máquina de estados, que no se ve afectada por la introducción

<sup>4</sup>Además, estos diagramas SDL van a tener que ser modificados, ver a continuación.

de las primitivas de asentimiento, sólo cambia el nombre de las primitivas DL-DATA Indication y DL-UNIT-DATA Indication mandadas al nivel de red.

### 5.3.3 Funcionamiento

#### 5.3.3.1 Segmentador

Se propone el siguiente comportamiento para el segmentador:

Cuando el segmentador reciba una primitiva que no va a ser fragmentada (tipo SEG-DATA Request, SEG-UNIT-DATA Request), se comportará transparentemente, convirtiendo la primitiva SEG- en DL-, manteniendo el mismo identificador. Igualmente ocurrirá con los asentimientos de primitivas no fragmentadas, DL-DATA Confirmation se convertirá transparentemente en SEG-DATA Confirmation.

Para cada primitiva SEG-DATA Request(Id) que se vaya a fragmentar, el segmentador generará los fragmentos DL-DATA Request(Id'), creará una tabla asociada a Id cuyos elementos serán Id' y el estado de asentimiento de cada fragmento (si este fragmento está asentido o no), como se muestra a continuación:

Id	
Identificador del segmento	¿Segmento asentido?
Id' <sub>1</sub>	SI
Id' <sub>2</sub>	SI
Id' <sub>3</sub>	NO
Id' <sub>4</sub>	NO

En resumen, el comportamiento de fragmentación es:

1. Cuando se fragmenta, el segmentador crea la tabla, almacenando en ella Id de la primitiva original e Id' de los fragmentos.
2. Cuando el segmentador reciba DL-DATA Confirm(Id') de uno de los segmentos, se marca este segmento como asentido en la tabla.
3. Cuando todos los segmentos de una tabla estén asentidos, el segmentador envía SEG-DATA Confirm(Id) y se borra la tabla.

En la figura 5.8, proponemos un nuevo diagrama sdl para el segmentador.

#### 5.3.3.2 Reensamblador

La máquina de reensamblado (páginas 123, 124, 125 de la norma) sólo cambia en la nomenclatura: las señales de salida DL-DATA Indication y DL-UNIT-DATA Indication se sustituyen respectivamente por SEG-DATA Indication y SEG-UNIT-DATA Indication, salvo en la página 125 en la que hay una errata (DL-DATA Indication en vez de DL-UNIT-DATA Indication, y por lo tanto la señal pasa a llamarse SEG-UNIT-DATA Indication).

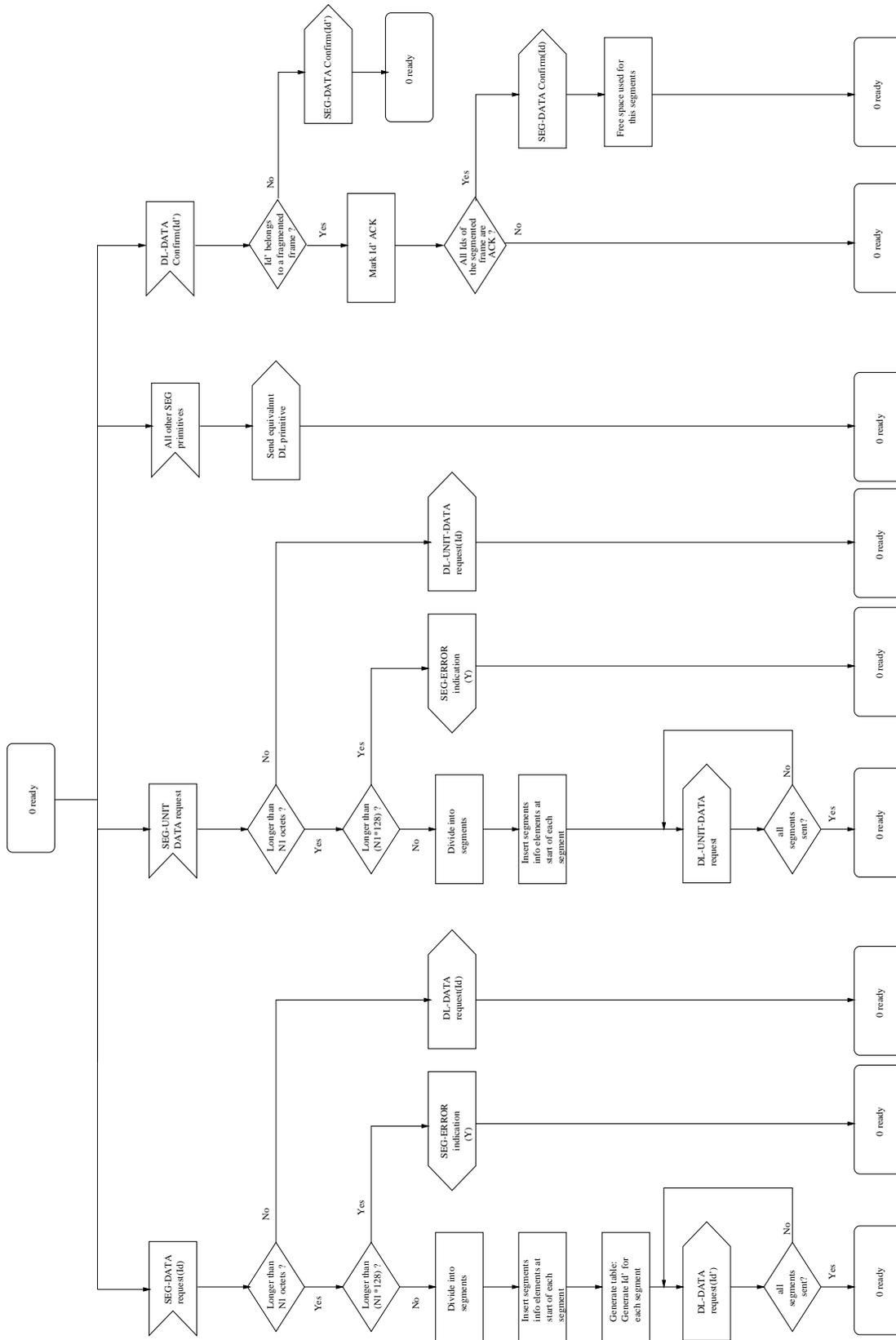


Figura 5.8: Diagrama SDL del segmentador

### 5.3.3.3 Máquina de enlace de datos

Gracias a las funciones desarrolladas por el segmentador, la máquina de enlace de datos sólo tiene que preocuparse de confirmar las primitivas DL-DATA Request, sean o no fragmentos.

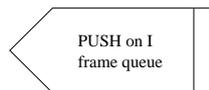
La máquina de enlace de datos encola los datos<sup>5</sup> de las primitivas DL-DATA Request en una cola First In First Out, que va vaciando mediante el procedimiento de ventana deslizante clásico de los sistemas hdlc. La ventana deslizante de transmisión marca exactamente que pdus han sido asentidas y cuales no, y por lo tanto una forma de generar los asentimientos DL-DATA Confirm(ld') es *asociar este proceso con el proceso de movimiento de la ventana deslizante de transmisión*.

Este seguimiento del movimiento de la ventana de transmisión se realiza comprobando las variaciones de la variable  $V(a)$ <sup>6</sup>, insertando una función de comprobación<sup>7</sup> después de los puntos en los que se asigna un valor a esta variable. Cuando  $V(a)$  ha cambiado, se envían las primitivas DL-DATA Confirm de las pdus asentidas.

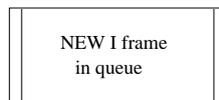
El seguimiento del identificador de las pdus asentidas se va a realizar mediante una cola First In First Out (que llamamos Id queue) en la que se van a meter los identificadores de las primitivas DL-DATA Request. Este proceso se realiza en el mismo momento de encolar los datos transportados por estas primitivas, ambos parámetros (identificadores y datos) entran sincronizadamente en sus respectivas colas.

El nivel de enlace entrega los datos en orden, de forma que la cola de datos por enviar nunca se desordena: si cada vez que un elemento de información es asentido se saca un elemento de la cola de identificadores, este elemento será el identificador de la primitiva correspondiente.

En la práctica, se va a sustituir el símbolo de “meter Trama de Información en la cola”:



Por una llamada a la función “Nueva Trama de Información en la cola”:



En la figura 5.9 vemos que esta nueva función almacena la parte de datos y el identificador en sus correspondientes colas.

En la figura 5.10 se representa la cola de identificadores, y la función de comprobación de variación de  $V(a)$ . Lo primero que se realiza es calcular la variación de  $V(a)$ ,

<sup>5</sup>La información transportada por las primitivas es en realidad un parámetro de estas. Es implícito, pero está ahí siempre.

<sup>6</sup>Último número de trama de la que se sabe con certeza su correcta recepción por el correspondiente.

<sup>7</sup>Que llamamos “Check npdu acknowledged”

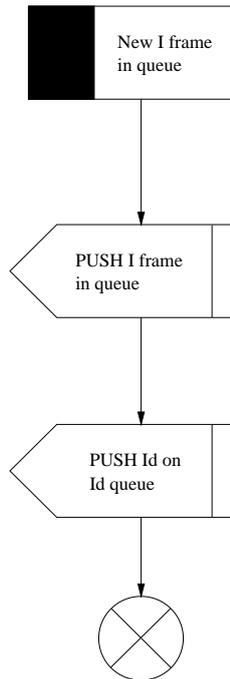


Figura 5.9: Función *New I frame in queue*

mediante la variable *OldVa* en la que se almacena el valor de *V(a)* la última vez que esta función fue llamada. A continuación se sacan tantos elementos como la variación de *V(a)* indica, y se generan las primitivas *DL-DATA Confirm(Id')* sacando *Id'* de la cola.

La inserción de la llamada a la función *Check npdu acknowledged* se realiza siempre en el punto siguiente a una instrucción de asignación de un valor a *V(a)* [*V(a) <- ??*]. En el diagrama SDL esto ocurre en las páginas 95 (rama *SREJ*), 96 (rama *REJ*), 99 (rama *RR*), 100 (rama *SREJ*) y 101 (rama *I*).

Esta cambio también podría hacerse en la función *Check I frame ack'd*, pero *V(a)* se actualiza en sus tres ramas. Para clarificar el diagrama, optamos por insertar una llamada a *Check npdu acknowledged* después de cada llamada a *Check I frame ack'd*, páginas 95 (rama *RR*), 96 (rama *I*).

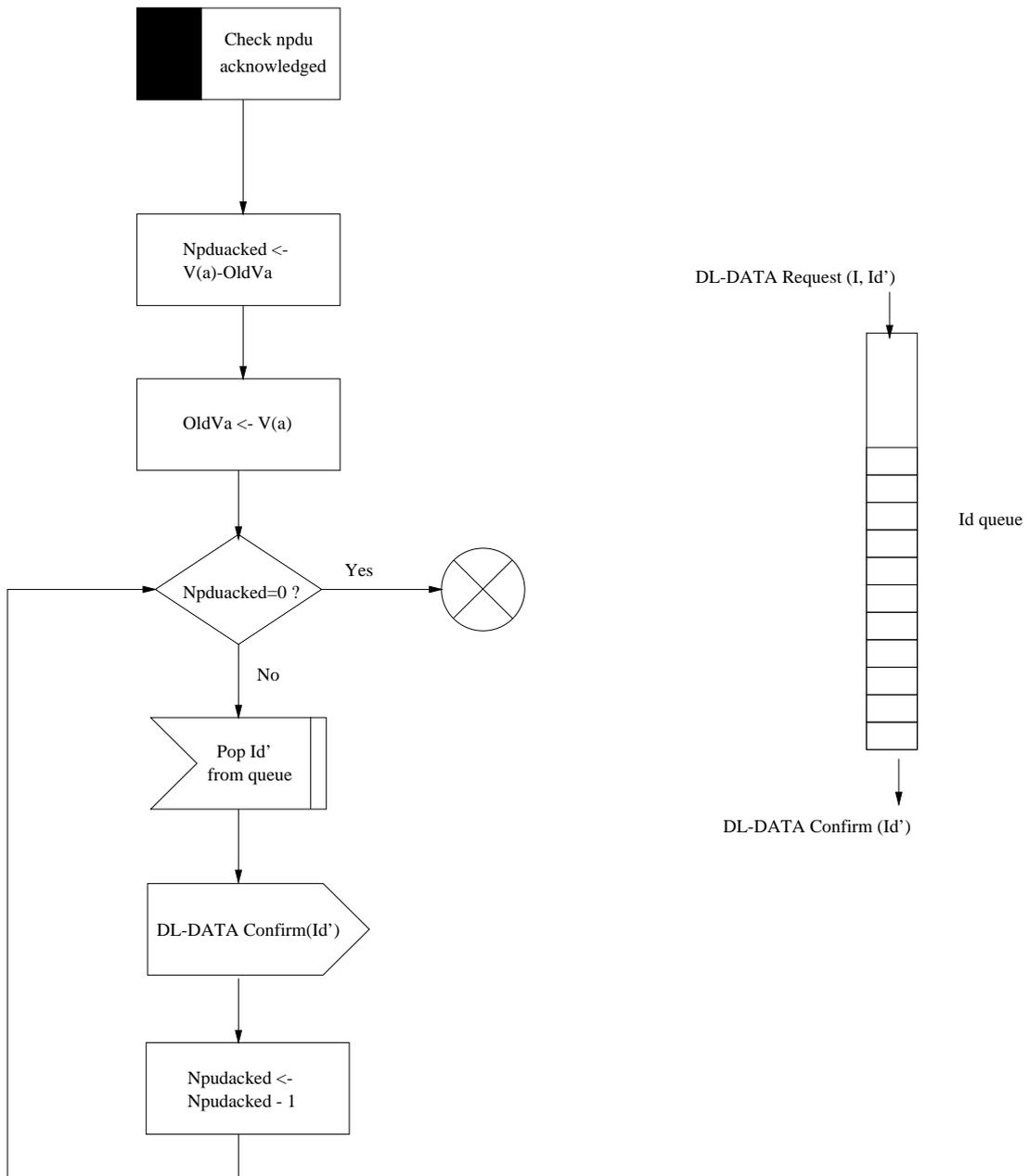


Figura 5.10: Función Check npdu acknowledged

## 5.4 Procedimiento general de medición

En la definición de los tipos de parámetros de la red (ver apartado 3.3), se explica que los parámetros primarios son transmitidos por los nodos en la red, y que su recepción permite obtener un grafo de la red con enlaces cuantificados por parámetros característicos, obtenidos a partir de los primarios.

Cada nodo debe por lo tanto realizar mediciones localmente, y obtener los parámetros primarios de sus enlaces, para luego difundirlos en la red. *Este apartado trata del procedimiento seguido para obtener estos parámetros primarios.*

Las magnitudes medidas, el tráfico, el ancho de banda disponible, la probabilidad de no error del enlace, pueden variar rápidamente, porque dependen de las condiciones de propagación del canal (variable en las frecuencias utilizadas), y de las condiciones de tráfico y congestión. Pero no es posible transmitir las cada vez que cambian, esto sería un gasto demasiado grande de ancho de banda; queremos optimizar los recursos. Una forma de facilitar la tarea de difusión<sup>8</sup> de estos parámetros, se hace suavizando las variaciones de las mediciones, mediante filtrado paso bajo.

Este filtrado paso bajo también es deseable para que los parámetros primarios obtenidos caractericen el comportamiento general de los enlaces, y no se vean demasiado influenciado por variaciones muy puntuales<sup>9</sup>.

El procedimiento general propuesto a continuación se divide en 4 fases: petición o recepción de la medida, filtrado paso bajo, cuantificación mediante umbrales, codificación, tal y como representamos en la figura 5.11.

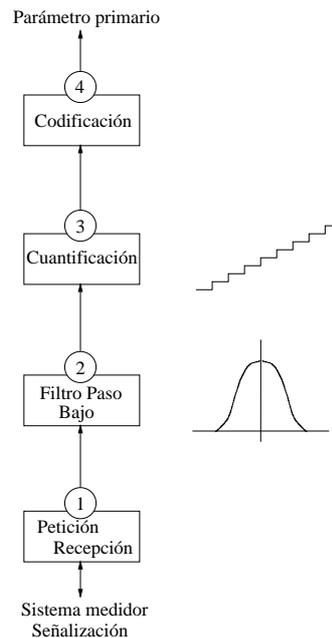


Figura 5.11: Las 4 fases del procedimiento general de medición

### 5.4.1 Fase 1 - Petición o recepción de medidas

Existen dos tipos de mediciones: las que son realizadas continuamente por sistemas del nodo (que llamamos *mediciones síncronas*), y las que sólo son medibles cuando ocurre un hito en el nodo (que llamamos *mediciones asíncronas*); este hito puede ser generado por varias causas, generalmente será la llegada de un elemento de información necesario para la obtención del parámetro primario.

En el caso de mediciones síncronas, se consultarán periódicamente al sistema medidor, con periodo  $T_{req}$  segundos, y se filtrará mediante un filtro digital, lo que requiere

<sup>8</sup>Llevada a cabo por la máquina de difusión de mapa de red.

<sup>9</sup>La labor de recopilación y filtrado de las medidas se lleva a cabo en la máquina de gestión de red, ver apartado 4.7.

almacenar las N salidas anteriores del filtro en una cola de N elementos (que llamamos *cola de tipo síncrono*).

Las mediciones asíncronas se almacenarán en el momento de su obtención, y se les aplicará un proceso de filtrado sencillo.

## 5.4.2 Fase 2 - Filtrado paso bajo de medidas

El siguiente paso consiste en realizar un filtrado paso bajo. El objetivo de este paso es suavizar las variaciones de las medidas, manteniendo la veracidad de las mismas.

Las colas de tipo síncrono serán procesadas mediante un filtro digital. Esto es posible porque se realizan muestreos periódicos de las mediciones, el filtrado digital se va a comportar como un filtrado analógico.

En cambio, las mediciones asíncronas no pueden ser procesadas de tal forma, por no existir tal periodicidad de mediciones. En efecto, si se filtraran digitalmente las medidas, convirtiendo estas medidas en una serie de números, se perdería la información de su antigüedad. Y si queremos que el valor obtenido al filtrar *represente lo mejor posible el valor actual de cantidad medida*, hay que darle más importancia a las mediciones recientes que a las mediciones antiguas; por ejemplo, no se trata igual la penúltima medición cuándo se ha realizado hace 10 segundos que cuándo se ha realizado hace 10 minutos.

### 5.4.2.1 Filtrado digital de mediciones síncronas

La obtención de las mediciones síncronas no consume recursos radioeléctricos, son realizadas en permanencia por sistemas medidores cuyas salidas vamos a muestrear; por lo tanto en este aspecto no hay limitación al período de muestreo (Treq) de estas mediciones. Pero sí existen otras limitaciones, debido a las propias características de la red, y las limitaciones de capacidad de proceso y de memoria del nodo; hay que escoger un valor de Treq adecuado a estas características.

**Orden de magnitud de Treq** Treq define el reloj de medición. De su valor va a depender mucho la velocidad de reacción del sistema. Un sistema al que se le exige reaccionar en milésimas de segundo, deberá realizar medidas a intervalos de este orden de magnitud.

Las redes de radio-paquete no requieren velocidades de reacción grandes, porque el objetivo principal no es la disponibilidad, sino la funcionalidad práctica a nivel de usuario final; los usuarios finales no exigen servicios de alta calidad, sino servicios que funcionen bien, con disponibilidades y calidades medianas. Es normal tener disponibilidades del orden del 80% o 90%<sup>10</sup>.

---

<sup>10</sup>Son estimaciones personales.

En este entorno, tiempos de respuesta del orden del minuto son razonables, y para obtener este resultado muestrear cada 10 segundos puede ser adecuado, como veremos a continuación.

**Límite superior de  $T_{req}$**   $T_{req}$  debe ser lo suficientemente bajo para que la salida del filtro no presente demasiado retardo, y esto depende de las características de la magnitud a medir; por ejemplo la presencia de escalones (mediciones que cambian bruscamente de valor y permanecen con este nuevo valor) hace aconsejable limitar el retardo, para que el sistema detecte estas variaciones a tiempo.

Como veremos en el apartado 5.6, los parámetros medidos son la capacidad disponible y la tasa binaria de cada enlace; ambas magnitudes presentan escalones. Los retardos máximos tolerables son del orden del minuto, por lo tanto  $T_{req}$  no debe superar este valor. Si por algún motivo el nodo no es capaz de garantizar esta cadencia, es necesario mejorar sus características técnicas.

**Límite inferior de  $T_{req}$**  Como veremos en el apartado 5.6, los sistemas medidores consultados utilizan una ventana de tiempo de duración  $T$ , durante la cual realizan el conteo de paquetes transmitidos; se trata de una integración en el intervalo  $T$ , un prefiltrado que hace que en intervalos  $T_{req}$  mucho más pequeños que  $T$ , las variaciones sean insignificantes.  $T$  es del orden de minutos, y con este orden de magnitudes no tendría sentido escoger  $T_{req}$  inferior a 1 segundo.

Además, si  $T_{req}$  es demasiado bajo, la frecuencia de muestreo es alta, y para conseguir que el filtro tenga un ancho de banda lo suficientemente estrecho hay que almacenar grandes cantidades de medidas, por lo tanto se aumentan las exigencias de capacidad de almacenamiento en memoria del nodo, a su vez que la capacidad de proceso (ya que cada  $T_{req}$  segundos es necesario recalcular la salida del filtro).

Otro factor a tener en cuenta es que las medidas (ancho de banda disponible y tasa de bits) se realizan paquete a paquete; es inútil realizar muestreos con periodos inferiores al tiempo de transmisión de un paquete, ya que no varía significativamente la magnitud medida. Es razonable escoger  $T_{req}$  varios ordenes de magnitud por encima del tiempo de transmisión de paquete. En condiciones habituales de radioaficionados, el tiempo de transmisión de un paquete es del orden de 100ms-1s (velocidades inferiores a 10Kbits, y con transceptores de tiempos de conmutación lentos).

**Valor de  $T_{req}$  escogido y respuesta al escalón** Teniendo en cuenta las consideraciones anteriores, se escoge  $T_{req}=10$  segundos<sup>11</sup>. Y se desea que en 50 segundos la repuesta al escalón unidad sea del orden de 0,8 (En aproximadamente 1 minuto se tiene el 80% de las variaciones de tipo escalón). Estos dos parámetros nos van a ayudar a fijar los parámetros del filtro digital.

---

<sup>11</sup>Hemos visto que varios parámetros del sistema pueden influir sobre un valor  $T_{req}$  óptimo. La elección tiene un grado de subjetividad, y puede ser objeto de experimentación. Por este motivo  $T_{req}$  es un parámetro del canal, modificable por el operador del nodo. Ver apéndice B página 128.

En el apéndice A se diseña el filtro digital.

### 5.4.2.2 Filtrado de mediciones asíncronas

El enfoque de filtrado digital tal y como se plantea en el apartado 5.4.2.1 no es utilizable directamente en el caso de las mediciones asíncronas. La llegada de mediciones asíncronas no es periódica, puede ser que la penúltima medición haya llegado hace tanto tiempo que no tenga sentido tenerla en cuenta, porque las condiciones medidas pueden haber variado completamente.

Por lo tanto, el filtrado debe tener en cuenta la antigüedad de las mediciones recibidas. Pueden diseñarse varios sistemas que tengan en cuenta este efecto, a continuación proponemos uno:

Nos basamos en un filtro digital paso bajo de primer orden, cuya transformada  $Z$  de su respuesta al impulso es:

$$H(z) = \frac{cte}{(1 - az^{-1})}$$

Este filtro presente un polo en  $z=a$ , con  $a \in ]0, 1[$ . Para que  $H(1) = 1$  es necesario que  $cte = 1 - a$ . La ecuación en diferencias de coeficientes constantes de este filtro es:

$$\begin{aligned} y[n] - ay[n-1] &= (1-a)x[n] \\ \Leftrightarrow y[n] &= (1-a)x[n] + ay[n-1] \end{aligned}$$

Proponemos un filtro de similares características, pero utilizando un  $a$  variable para ponderar la importancia del último resultado  $y[n-1]$  obtenido, en función del tiempo  $\Delta t$  que ha pasado desde la última medida. Es decir:

$$y = (1-a)x + ay_{-1}$$

La salida del filtro se calcula en el instante de la recepción de la medición  $x$ .  $y_{-1}$  es la salida del filtro calculada en el instante de la recepción de la anterior medición.

Para el cálculo de  $a$ , proponemos la siguiente expresión:

$$a = 1 - \min\left(1, \frac{\Delta t}{T_{asin}}\right)$$

Siendo:

$\Delta t$  el tiempo en segundos que ha pasado desde la recepción de la última medición.

$T_{asin}$  es el tiempo máximo en segundos de consideración de la salida de filtro. Cuando  $\Delta t > T_{asin}$ , el valor de  $a$  es 0, y por lo tanto  $y_{-1}$  no es tenido en cuenta, y toma el valor de la medición recibida ( $y = x$ ).

$a$  varía linealmente entre 1 (cuando  $\Delta t=0$ ) y 0 (cuando  $\Delta t = T_{asin}$ ).

Un valor razonable en un canal de 1200 baudios es  $T_{asin} = 1800\text{segundos}$ <sup>12</sup> (media hora), ya que como veremos en el apartado 5.5.1 las mediciones se realizan cada 10 minutos.

En resumen, hemos promediado la medición con la anterior salida del filtro, siendo la ponderación de esta última menor cuanto mayor sea la antigüedad de la última medición asíncrona recibida.

### 5.4.3 Fase 3 - Cuantificación mediante umbrales

El objetivo de la cuantificación es reducir las variaciones de las medidas en un rango de valores discretos, al fin de optimizar los recursos de la red utilizados en difundir los parámetros primarios, facilitando el trabajo de la máquina de distribución del mapa de red.

La cuantificación no afecta en absoluto la codificación del parámetro primario (este se sigue codificando con el mismo formato y la misma precisión), ni ninguno de los mecanismos de difusión y recepción de la información, y por lo tanto puede realizarse con total flexibilidad.

Se propone la utilización de un cuantificador de intervalos constantes, que tomará dos parámetros locales, que varían según el tipo de medida:

**Vmax:** Valor máximo que puede tomar la medición. En el caso de la probabilidad de no error ( $P_j$ ),  $V_{max}=1$ , en caso de las tasas de bits ( $C_j, F_j$ ),  $V_{max}=B_i$ , siendo  $B_i$  la tasa binaria nominal del canal utilizado.

**Nint:** El número de intervalos de cuantificación.

La longitud de los intervalos de cuantificación será  $\frac{V_{max}}{N_{int}-1}$ , y el valor cuantificado es igual a  $K \times \frac{V_{max}}{N_{int}-1}$  con  $K \in [0, 1, \dots, N_{int} - 1]$ .

La probabilidad de no error  $P_j$  se codifica en 32 niveles, es recomendable utilizar una cuantificación coherente con esta codificación, por ejemplo con  $N_{int} = 2^n; n \leq 5$ .

El efecto sobre las tasas de bits puede ser perjudicial si no se elige  $N_{int}$  lo suficientemente grande, porque se perdería precisión en la diferencia de  $C_j - F_j$ , valor utilizado para la decisión del encaminamiento (un enlace con  $C_j - F_j = 0$  no se utiliza jamás) y el control de flujo de acceso a la red (no se inyecta nunca tráfico en un enlace

<sup>12</sup>La elección de este parámetros tiene un grado de subjetividad, y puede ser objeto de experimentación. Por este motivo  $T_{asin}$  es un parámetro del canal, modificable por el operador del nodo. Ver apéndice B.

con  $C_j - F_j = 0$ ). Para evitar este efecto, se pueden escoger valores de  $N_{int}$  diferentes (primos entre si).

Los parámetros locales  $V_{max}$  y  $N_{int}$  asociados a los cuatro tipos de parámetros primarios se nombran:  $V_{maxPj}$ ,  $V_{maxCj}$ ,  $V_{maxFj}$ ,  $V_{maxHj}$ ,  $N_{intPj}$ ,  $N_{intCj}$ ,  $N_{intFj}$ ,  $N_{intHj}$ <sup>13</sup>.

#### 5.4.4 Fase 4 - Codificación

Esta fase consiste codificar el parámetro primario como se especifica en el apartado 3.3.3.

---

<sup>13</sup>“Valor máximo de la salida de los cuantificadores de los parámetros primarios  $P_j$ ,  $C_j$ ,  $F_j$  o  $H_j$ ” y “Número de intervalos de cuantificación de los cuantificadores de  $P_j$ ,  $C_j$ ,  $F_j$  o  $H_j$ ”. Se trata de parámetros locales del nodo asociado a cada canal. Ver apéndice B.

## 5.5 Medición de $P_j$

La medición de  $P_j$ , probabilidad de que un paquete atraviese sin error un enlace sin conexión hdlc, se realiza con la participación de los nodos origen y destino del enlace.

El proceso es simple: el nodo destino contabiliza el número de paquetes que recibe del nodo origen, mientras que este último transmite de vez en cuando señalización informando del número de paquetes que ha transmitido. El nodo destino por lo tanto conoce la magnitud  $\frac{\text{Número de paquetes recibidos por el nodo destino}}{\text{Número de paquetes transmitidos por el nodo origen}}$  que es precisamente  $P_j$ . Una vez conocido  $P_j$ , se filtra, siguiendo el procedimiento de filtrado de mediciones asíncronas del apartado 5.4.2.2:

*La medición de  $P_j$  se obtiene en el nodo destino.*

### 5.5.1 Procedimiento en el nodo origen de un enlace

El papel del nodo origen del enlace es simple: transmitir periódicamente con periodo  $T_{P_j}$ <sup>14</sup> segundos el número de paquetes transmitidos ( $N_{paqtx}$ ). La magnitud  $N_{paqtx}$  es independiente del enlace, se refiere al número total de paquetes transmitidos en el canal.

Esta operación se realiza con un solo proceso transmisor (lanzado por la máquina de gestión de red) asociado a cada canal. El periodo de transmisión  $T_{P_j}$  y el número de paquetes transmitidos  $N_{paqtx}$  son valores específicos de cada proceso, independientes de los procesos asociados a otros canales. El proceso transmisor realizará las siguientes operaciones con periodicidad  $T_{P_j}$ :

1. Se guarda el número de paquetes transmitidos en la anterior transmisión en la variable  $N_{paqtxlast}$ :  $N_{paqtxlast} = N_{paqtx}$ .
2. Se pide a la máquina de medición de enlace a nivel de enlace *del canal asociado* al proceso transmisor el número de paquetes transmitidos por la dirección de enlace asignada al nodo, mediante la primitiva MEA-SENT Request (Dirección de enlace del nodo).
3. Se recibe la respuesta a la solicitud, con la primitiva MEA-SENT Confirm (Número de paquetes transmitidos), y se almacena en la variable  $N_{paqtx}$ .
4. Con  $N_{paqtxlast}$  y  $N_{paqtx}$  se calcula  $N'_{paqtx}$ , el número de paquetes que se han transmitido durante el último periodo<sup>15</sup>.
5. Se calcula  $N_{paqacc}$ , el número acumulado de paquetes transmitidos:  

$$N_{paqacc} = N_{paqacc} + N'_{paqtx}$$

<sup>14</sup>“Periodo de transmisión del número de paquetes transmitidos en un determinado canal”. Se trata de un parámetro local del nodo asociado a cada canal. Ver apéndice B.

<sup>15</sup>Esta operación depende de la codificación del número de paquetes. Ver apartado 5.2.2.5.

6. Si  $N_{paqacc}$  no supera el umbral  $N_{paqtxmin}$  no se hace nada. El proceso reanuda en el punto 1 en el próximo periodo.
7. Si  $N_{paqacc}$  supera el umbral  $N_{paqtxmin}$ :
  - (a) Se lanza el proceso de transmisión de señalización de medición de probabilidad de error, con  $N_{paqtx}$  en su campo de información.
  - (b)  $N_{paqacc} = 0$ .

$N_{paqtxmin}$ <sup>16</sup> es el número mínimo de paquetes transmitidos en el canal para efectuar la transmisión del número de paquetes transmitidos. En el proceso de comparación del número de paquetes recibidos con el número de paquetes transmitidos pueden haber errores (por ejemplo debido a paquetes encolados) que falsean las medidas. Con  $N_{paqtxmin}$  se garantiza un valor mínimo de paquetes transmitidos, para limitar el error relativo de la medición.

Para un canal a 1200 baudios, se puede escoger por ejemplo  $N_{paqtxmin}=50$ .

**Proceso de transmisión de señalización de medición de probabilidad de error** El proceso de transmisión propiamente dicho es sencillo:

Se utiliza el módulo Broadcast asociado al canal, para transmitir una trama no numerada, que podrá ser captada por *todas* las estaciones presentes en el canal. Es fundamental que esta trama sea transmitida con la máxima prioridad, antes que cualquier otro paquete; en el caso contrario, el número de paquetes transmitidos no se corresponde con la realidad, y se introduce un error en la medición. Dependiendo de la implementación del nodo, puede ser incluso razonable enviar directamente la señalización al módulo de Broadcast sin pasar por el encaminador, para evitar pasar por sus colas.

### 5.5.2 Procedimiento en el nodo destino de un enlace

Cuando el decodificador de señalización recibe una trama de señalización de medición de probabilidad de error, envía la señal NM-Error Indication(Canal, Dirección AX.25,  $N_{paqtx}$ ) a la máquina de gestión de red. El procedimiento de medición es lanzado por la máquina de gestión de red cuando recibe esta señal, realizando las siguientes operaciones:

1. Se almacena  $N_{paqtx}$  en la variable  $R_{paqtx}$ .
2. Se pide a la máquina de medición de enlace a nivel de enlace *del canal indicado* el número de paquetes recibidos de este nodo, mediante la primitiva MEA-HEARD Request (Dirección AX.25).

<sup>16</sup>“Número mínimo de paquetes transmitidos para lanzar la señalización de medición de probabilidad de error de paquete en el nodo origen de un enlace.” Se trata de un parámetro local del nodo asociado a cada canal. Ver apéndice B.

3. Se recibe la respuesta a la solicitud, con la primitiva MEA-HEARD Confirm (Número de paquetes oídos, tiempo transcurrido), y se almacena “Número de paquetes oídos” en la variable  $N_{paqrx}$ .
4. Con  $N_{paqrxlast}$  y  $N_{paqrx}$  se calcula  $N'_{paqrx}$ , el número de paquetes de la estación origen recibidos desde la recepción de la penúltima primitiva NM-Perror Indication<sup>17</sup>.
5. Con  $R_{paqtxlast}$  y  $R_{paqtx}$  se calcula  $R'_{paqtx}$ , el número de paquetes que la estación origen ha transmitido desde la recepción de la penúltima primitiva NM-Perror Indication<sup>18</sup>. Si  $R'_{paqtx=0}$ , se para el proceso.
6. Se calcula la probabilidad de no error de paquete:  $P'_j = \frac{N'_{paqrx}}{R'_{paqtx}}$
7. Se almacena  $N_{paqrx}$  en  $N_{paqrxlast}$  :  $N_{paqrxlast} = N_{paqrx}$
8. Se almacena  $R_{paqtx}$  en  $R_{paqtxlast}$  :  $R_{paqtxlast} = R_{paqtx}$

---

<sup>17</sup>Esta operación depende de la codificación del número de paquetes. Ver apartado 5.2.2.5.

<sup>18</sup>Esta operación depende de la codificación del número de paquetes. Ver apartado 5.2.2.5.

## 5.6 Medición de Fj

La tasa binaria (o flujo de datos) transportada por un enlace es medida por su máquina de medición de enlace a nivel de red. Cada máquina de medición de enlace a nivel de red tiene asignado un único enlace, siendo esta un punto de paso obligatorio para cualquier dato que vaya a ser transmitido mediante este enlace; ella interacciona directamente con el nivel 2, mediante las primitivas SEG-DATA Request, SEG-DATA Confirm, SEG-UNIT-DATA Request.

La medición de Fj no se hace de forma continua, sino en intervalos periódicos de tiempo, pero esta característica es ocultada al resto de sistemas, ya que la primitiva de consulta a esta medición (NMEA-FJ Request) es contestada (NMEA-FJ Confirm) con el valor obtenido en el último periodo. Fj es por lo tanto una medición síncrona.

El procedimiento de medición no es el mismo para un enlace con conexión o sin conexión hdlc, se definen dos máquinas diferentes para cada caso, que describimos a continuación.

### 5.6.1 Enlaces con conexión hdlc

La medición de la velocidad en un enlace hdlc se realiza observando cuantas pdus han sido asentidas durante un intervalo de tiempo determinado.

Para llevar a cabo esta labor, se crea una tabla (tabla almacén de primitivas DL-DATA Request transmitidas) en la que se almacenan las propiedades de los paquetes transmitidos. Para cada paquete transmitido se crea un registro (fila de la tabla) con los siguientes datos (columnas):

**Identificador** El identificador de la primitiva SEG-DATA Request con la que se ha mandado la pdu al nivel de enlace.

**Longitud** El número de bits transportados (longitud en bits del campo de datos de la primitiva SEG-DATA Request).

**ACK** Indicador booleano. Verdadero si la primitiva ha sido asentido (SEG-DATA Confirm), falso si no lo ha sido.

**Tiempo** Indicador temporal de cuando ha llegado la primitiva a la máquina de medición de enlace a nivel de red. Sólo es necesario para medir el tiempo transcurrido, y no es necesario que el reloj esté en hora.

Cuando la máquina de medición de enlace recibe una pdu para transmitir (mediante la primitiva NMEA-DATA Request), genera un Identificador que no esté en la tabla, añade una fila con los datos mencionados, y envía al nivel de enlace la primitiva SEG-DATA Request (Identificador) correspondiente.

Cuando recibe del nivel de enlace la primitiva SEG-DATA Confirm (Identificador), marcará como verdadero el campo ACK, y anotará el instante de tiempo actual en una

Identificador	Longitud	ACK	Tiempo
4240	200	Falso	13h15min05s
4236	200	Falso	13h15min01s
4235	1024	Verdadero	13h14min50s
4234	256	Verdadero	13h14min10s

Tabla 5.1: Tabla almacén de primitivas DL-DATA Request transmitidas

variable (por motivos ajenos a la medición de  $F_j$ , ver apartado 5.7 de medición de  $C_j$ ). El procedimiento de medición es el siguiente:

Con un periodo de  $TF_j$ <sup>19</sup>, se realizan las siguientes operaciones:

1. Borrar las entradas de primitivas *asentidas* más antiguas que  $TF_jwin$ <sup>20</sup> ( $NOW - Tiempo \geq TF_jwin$ ).
2. Calcular Nbits, el número de bits transportados durante el intervalo  $TF_jwin$ , sumando las longitudes de las primitivas asentidas.
3. Almacenar en el registro interno la tasa binaria medida:  $F_j = \frac{Nbits}{TF_jwin}$ .

En las figuras 5.12, 5.13, 5.14 se puede ver el diagrama SDL del módulo de medición de  $F_j$  y  $C_j$ .

## 5.6.2 Enlaces sin conexión hdlc

En un enlace sin conexión hdlc, el nivel 2 no realiza el control de flujo de ventana deslizante. En el transmisor no es posible conocer si una pdu ha sido entregada, ni cuando ha sido entregada. El diseño de AX25 v2.2 es tal que el nivel de red puede colapsar por completo el transmisor, no existiendo ningún mecanismo de control de flujo, en ninguno de los puntos. Las tramas son encoladas en la cola del multiplexor hasta el momento en el que les toque el turno ser transmitidas a este enlace; en ese momento la cola es vaciada por completo en la máquina PHY, que no realiza ninguna acción de control de flujo<sup>21</sup>.

Ante esta situación, desde el transmisor resulta difícil la medición del flujo real transportado por el enlace, porque el flujo saliente del nivel de red o del nivel de enlace no se corresponden con el flujo real transportado por el enlace.

En cambio, desde el receptor (nodo destino del enlace) la medición de  $F_j$  se realiza fácilmente, porque los datos llegan a la tasa que los ha podido transportar el canal:

<sup>19</sup>“Periodo de medición de  $F_j$ ”. Se trata de un parámetro local del nodo asociado a cada canal. Ver apéndice B.

<sup>20</sup>“Ventana de medición de  $F_j$ ”. Se trata de un parámetro local del nodo asociado a cada canal. Ver apéndice B.

<sup>21</sup>Salvo interrupciones temporales del transmisor si este está más de 10 minutos transmitiendo sin parar, pero no es realmente control de flujo ya que no evita el colapso de la cola del transmisor, sino que es una medida preventiva para dar oportunidades de transmitir a otras estaciones del mismo canal.

*En enlaces sin conexión hdlc,  $F_j$  se mide en el receptor.*

La medición de la velocidad se realiza observando las pdus que han llegado durante un intervalo de tiempo determinado.

Para llevar a cabo esta labor, se crea una cola en la que se almacenan la longitud del campo de datos y el instante de llegada de las primitivas SEG-UNIT-DATA Indication. El procedimiento de medición es el siguiente:

Con un periodo de  $TF_j^{22}$ , se realizan las siguientes operaciones:

1. Borrar (sacar) de la cola los elementos más antiguos que  $TF_{jwin}^{23}$ .
2. Calcular Nbits, el número de bits recibidos durante el intervalo  $TF_{jwin}$ , sumando las longitudes de los paquetes recibidos.
3. Almacenar en el registro interno la tasa binaria medida:  $F_j = \frac{Nbits}{TF_{jwin}}$ .

---

<sup>22</sup>“Periodo de medición de  $F_j$ ”. Se trata de un parámetro local del nodo asociado a cada canal. Ver apéndice B.

<sup>23</sup>“Ventana de medición de  $F_j$ ”. Se trata de un parámetro local del nodo asociado a cada canal. Ver apéndice B.

## 5.7 Medición de $C_j$

### 5.7.1 Análisis general

Todos los enlaces presentes en un mismo canal radioeléctrico comparten un recurso finito: el canal tiene un ancho de banda limitado. El ancho de banda de canal disponible en cada momento depende del uso que hace de él cada enlace.

Por lo tanto, no se puede definir la capacidad de un enlace como un ancho de banda fijo disponible asignado a este enlace, porque en la realidad cada enlace comparte el canal, y su capacidad varía en función del tráfico de cada uno de ellos. Una definición más apropiada de la capacidad  $C_j$  de un enlace sería: tasa máxima de datos que el enlace podría transportar si las condiciones del canal y el comportamiento del resto de los enlaces se mantuvieran estacionarios.

Para la consecución de la medición de tal magnitud se han analizado las siguientes vías de trabajo:

1. Medición del tiempo de ida y vuelta: Se trata del sistema utilizado en la red flexnet e inp (ver capítulo 2).  
Este sistema supone transmitir periódicamente paquetes, y medir el tiempo que tardan en ir y volver. Para que la medida sea exacta, se debe presuponer una cierta simetría del enlace. Esta premisa no es cierta en Jnet, y por lo tanto se descarta la opción.
2. Transmisión de ráfagas: la capacidad del canal ( $C_j$ ) es la tasa de datos transmitida ( $F_j$ ) medida en situaciones en las que *hay paquetes encolados en el nivel de enlace*; dicho de otra forma, la tasa de datos ( $F_j$ ) no es igual a la capacidad del canal ( $C_j$ ) porque hay momentos en los que el enlace está inactivo (no hay datos pendientes de transmitir por este enlace). A la situación de tener paquetes encolados a nivel de enlace la llamamos “ráfaga”, y para realizar las mediciones en situación de “ráfagas”, se ha pensado en dos posibilidades:
  - (a) Provocar ráfagas de paquetes: se envían paquetes de prueba al nivel de enlace a una tasa superior que la capacidad del canal, y se mide  $F_j$ .  
El principal problema de esta posibilidad es que los nodos consumen tráfico del canal (un recurso escaso que hay que optimizar), y que la propia medición perturba la medida (se consume capacidad del enlace, por lo tanto la capacidad de este es menor).
  - (b) Detectar ráfagas de paquetes: no introducir paquetes de prueba, pero detectar cuando un enlace envía ráfagas. Se han estudiado dos criterios de detección de ráfagas:
    - i. Medir la tasa de envío de los paquetes al nivel de enlace:  
se determina un umbral de velocidad de entrega de paquetes del nivel de red al nivel de enlace a partir del cual se considera que los paquetes

son enviados en ráfaga. El principal problema es calcular este umbral. Si se le fija a la capacidad nominal del interfaz ( $B_j$ ), es probable que en muchas situaciones no llegue a superarse (por ejemplo en el caso de un nodo con un canal de entrada de menor capacidad que el canal de salida).

ii. Observar la cola en el nivel de enlace.

Se trata de detectar los momentos en los que hay paquetes por enviar en la cola de la máquina de nivel de enlace, y obtener  $C_j$  realizando mediciones sobre estos paquetes. Esta es la vía seguida para enlaces con conexión hdlc.

3. Evaluación de la longitud media de la cola. Se trata de aplicar la teoría de colas, al fin de evaluar la capacidad del sistema a partir de  $F_j$  y la longitud media de la cola. Esta opción es analizada en el apéndice D.
4. Medición del porcentaje de tiempo que el canal está ocupado. De esta magnitud se puede derivar una estimación del ancho de banda disponible en el canal. Es la opción elegida para enlaces sin conexión hdlc.

Formalmente  $C_j$  es una medición síncrona, al igual que  $F_j$ , aunque como veremos a continuación la máquina de medición de enlaces a nivel de red la va a obtener asíncronamente. Para responder a la peticiones síncronas de la máquina de gestión de red (primitivas NMEA-CJ Request, NMEA-CJ Confirm), la máquina de medición de enlaces utilizará un registro donde almacena el último valor medido de  $C_j$ <sup>24</sup>.

## 5.7.2 Enlaces con conexión hdlc

La técnica empleada consiste en detectar las ráfagas, medir su longitud  $N_{raf}$  en bits y el tiempo  $T_{raf}$  que tardan en llegar correctamente al otro extremo del enlace, y calcular la capacidad  $C_j = \frac{N_{raf}}{T_{raf}}$ . Esta medición se va a realizar simultáneamente a la medición de  $F_j$ , por una única máquina. Se van a necesitar tres nuevas variables:

$T_{raf\ first}$  : instante de llegada del primer paquete de la ráfaga.

$T_{raf\ last}$  : instante de llegada del último asentimiento de un paquete de la ráfaga.

$N_{raf}$  : longitud en bits de la parte de la ráfaga que ha llegado correctamente al destino del enlace.

A continuación describimos la operativa de la máquina, y en las figuras 5.12, 5.13, 5.14 se puede ver el diagrama SDL.

### 5.7.2.1 Detección de ráfagas

El primer paso consiste en detectar y delimitar las ráfagas. Esta tarea, como hemos destacado anteriormente, se puede realizar observando la cola del nivel de enlace. En

<sup>24</sup>Salvo si  $F_j > C_j$ , ver apartado 5.7.2.4.

concreto, como la tasa de transporte de información es marcada por la máquina de enlace de datos, es su cola la que debe ser observada.

No obstante, gracias a la implementación del asentimiento SEG-DATA Confirm, es posible realizar una tarea equivalente a nivel de red, y obviamente optaremos por esta solución ya que permite no modificar AX.25 v2.2.

La delimitación de las ráfagas se realiza definiendo dos estados en la máquina de mediciones:

**Estado-0 Esperando principio de ráfaga** Es el estado inicial de la máquina. Cuando llega la primitiva NMEA-DATA Request a la máquina de medición de enlaces a nivel de red (solicitud de transmisión de pdu), se comprueba si la primitiva anterior ha sido asentida (campo ACK de la tabla almacén de primitivas DL-DATA Request transmitidas, ver tabla 5.1).

Si la primitiva anterior no ha sido asentida, se considera que ha empezado una nueva ráfaga, formada por la primitiva actual y la anterior:

- $T_{raf\ first}$  toma el valor del instante de llegada de la primitiva anterior (campo TIEMPO de la tabla almacén de primitivas DL-DATA Request transmitidas).
- Se cambia al estado 1.

**Estado-1 Esperando fin de ráfaga** Se considera que una ráfaga termina cuando han sido asentidas todas las primitivas de la tabla almacén de primitivas DL-DATA Request (campo ACK). Esta comprobación se realiza al recibir un asentimiento SEG-DATA Confirm.

Cuando una primitiva es asentida:

- Se almacena en el registro interno  $T_{raf\ last}$  el instante actual.
- Sea  $N$  la longitud de la pdu asentida:  $N_{raf} = N_{raf} + N$ .
- Si todas las primitivas han sido asentidas (=fin de ráfaga):
  - Almacenar en el registro interno la capacidad medida:
 
$$C_j = \frac{N_{raf}}{NOW - T_{raf\ first}}$$
 siendo  $NOW$  el instante actual, medido de la misma forma que  $T_{raf\ first}$ .
  - Se cambia al estado 0.

### 5.7.2.2 Ráfagas largas

Como se puede observar, el procedimiento de medición de  $C_j$  es asíncrono, y sólo se conoce su nuevo valor hasta que termina una ráfaga. Este comportamiento puede ser problemático: un aumento súbito del tráfico puede generar una ráfaga muy larga, tan larga que solo se acabaría cuando el tráfico sea regulado por mecanismos de control de flujo interno de la red. Durante este intervalo de tiempo, el nodo no habría actualizado

$C_j$ , a diferencia de  $F_j$  que se actualiza periódicamente. Es importante que  $C_j$  y  $F_j$  estén sincronizados, porque  $C_j - F_j$  representa el ancho de banda disponible, y se utiliza en el algoritmo de control de acceso a la red. Valores no sincronizados pueden provocar que los nodos inyecten demasiado tráfico en la red.

Por este motivo, se va a forzar la actualización de  $C_j$  durante las ráfagas largas. En estado-1, se lanza un temporizador con periodo  $TF_{jwin}$ , definido en el apartado 5.1, que permitirá actualizar  $C_j$ : 
$$C_j = \frac{N_{raf}}{T_{raflast} - T_{raf first}}$$

### 5.7.2.3 Ráfagas cortas

En el caso opuesto de ráfagas muy cortas, puede darse el caso que la medida obtenida no sea acorde a la realidad. Por este motivo, se define el parámetro  $N_{rafmin}$ <sup>25</sup>. Si en el proceso de cálculo de  $C_j$  se verifica que  $N_{raf} \leq N_{rafmin}$ , no se calcula  $C_j$  (se mantiene el valor anterior).  $N_{rafmin} \geq 0$ , y por lo tanto no se calcula jamás el valor de  $C_j$  si ninguna de las pds de la ráfaga han sido asentidas ( $N_{raf} = 0$ ).

### 5.7.2.4 Medición de $C_j$

El filtrado de  $C_j$  se produce en la máquina de gestión de red, con la función de filtrado definida en el apéndice A. La máquina de medición de enlaces a nivel de red se limita a responder a las primitivas periódicas de petición de la medición  $C_j$  con el valor almacenado en su registro interno. No obstante,  $F_j$  y  $C_j$  son medidos mediante algoritmos diferentes, y al fin de evitar incongruencias del tipo  $F_j > C_j$ , en el registro se almacena siempre  $\max(F_j, C_j)$ .

<sup>25</sup>“Longitud mínima de ráfaga”. Se trata de un parámetro local del nodo asociado a cada canal. Ver apéndice B.

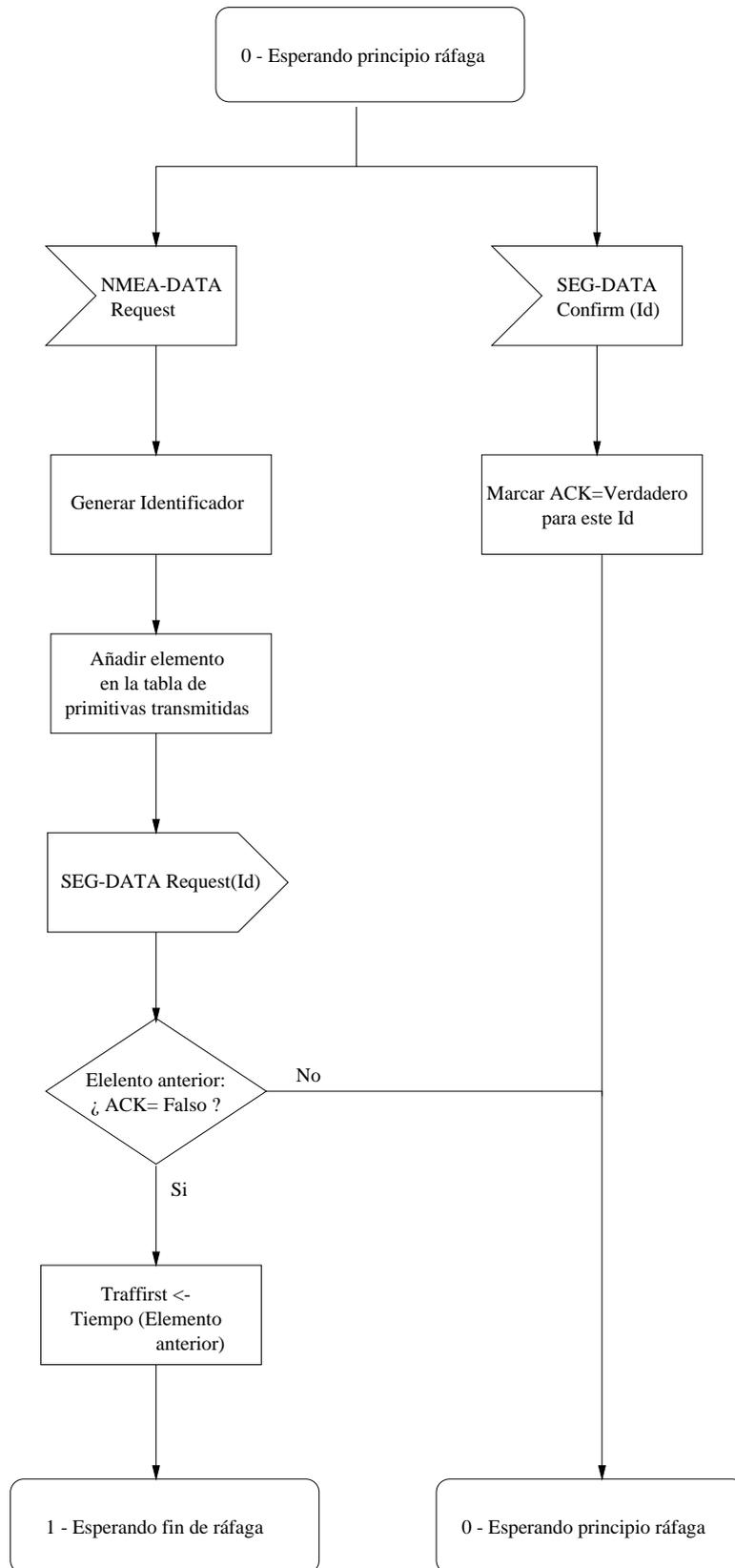


Figura 5.12: Máquina de medición de Cj y Fj - Estado 0

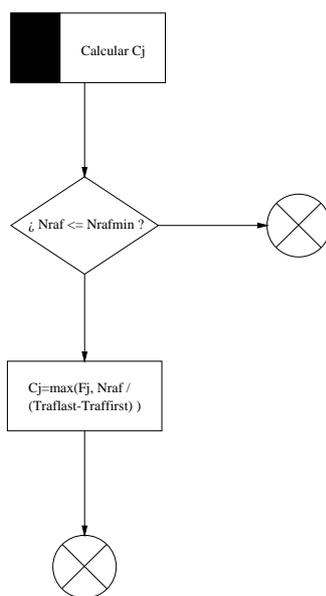
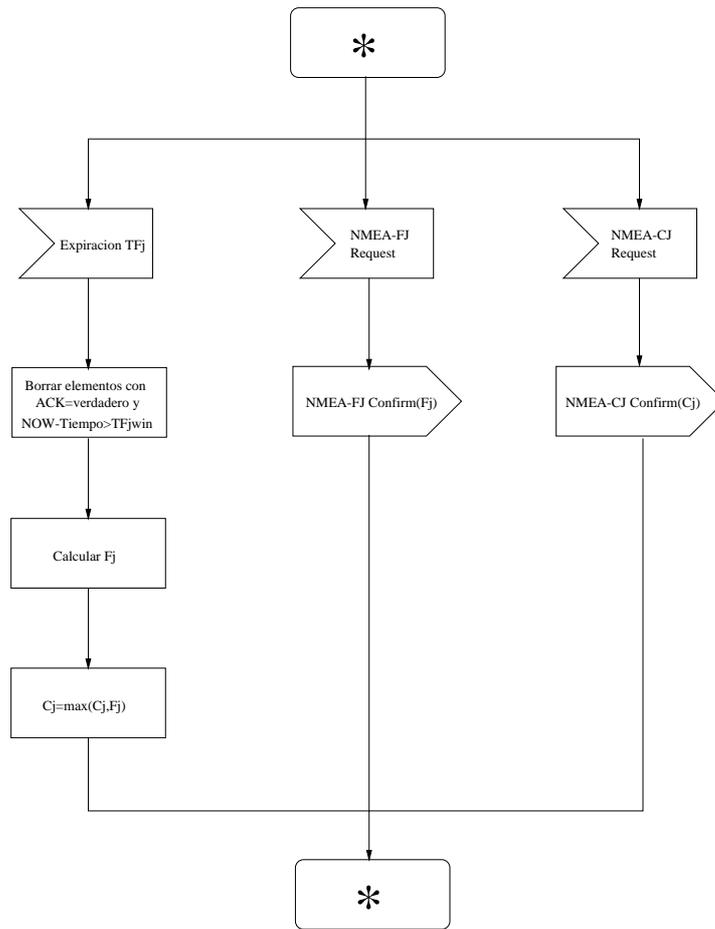


Figura 5.13: Máquina de medición de Cj y Fj

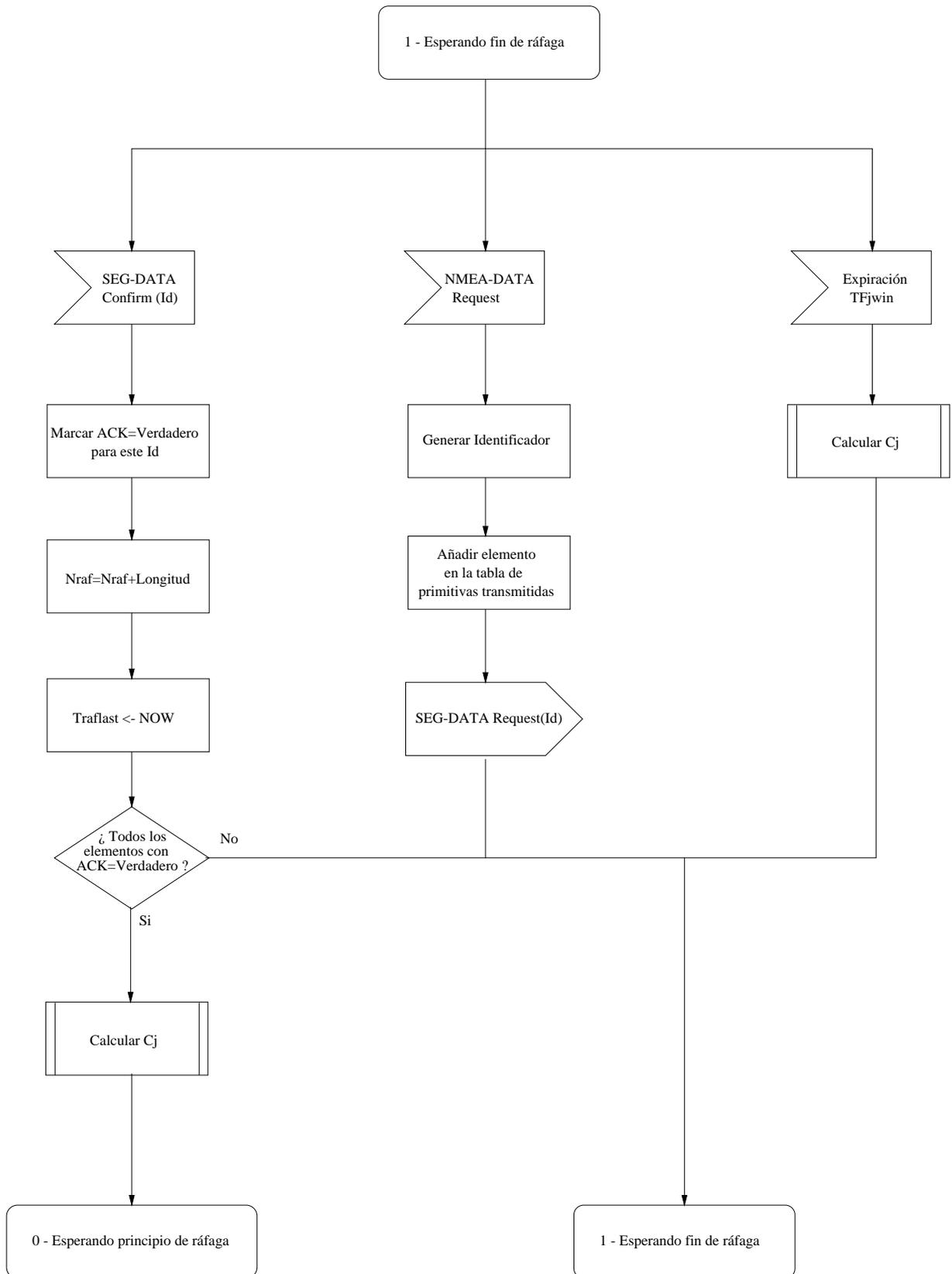


Figura 5.14: Máquina de medición de Cj y Fj - Estado 1

### 5.7.3 Enlaces sin conexión hdlc

Por lo motivos expuestos en el apartado de medición de  $F_j$  (5.6.2), en enlaces sin conexión hdlc no es posible detectar ráfagas desde el nivel de red, porque no se conoce que pds han sido entregadas y cuales no. Debido al diseño de AX.25 v2.2, esta detección solo sería posible a nivel PHY, y se sale del marco de estudio de este proyecto.

Una alternativa es calcular  $C_j$  a partir de  $F_j$  y de la longitud media de la cola (apéndice D), pero tampoco es posible por el mismo motivo: la cola se distribuye entre el nivel 2 y el nivel 1, medirla supone modificar ambas capas.

Es por lo tanto necesario un *enfoque* alternativo. La arquitectura que ofrece Jnet es muy flexible, permite que el nodo realice cualquier tipo de medición (parámetro primario) útil a la obtención del parámetro característico del enlace. Es incluso posible que el nodo origen y el nodo destino midan independientemente parámetros primarios de naturaleza diferente, de forma que el procesado de ambas medidas permita obtener un parámetro característico del enlace. Queremos destacar que esta flexibilidad da un importante grado de libertad, y que profundizando en el análisis siempre se podrían obtener enfoques y algoritmos de medición más eficaces, o más exactos.

En este enfoque propuesto, se asocia la capacidad de un enlace sin conexión hdlc con la capacidad disponible en el canal radioeléctrico. La capacidad disponible en el canal puede asimilarse con intervalos de tiempo en los que no existe portadora alguna en el canal. Estos intervalos de tiempo están disponibles para transmitir tramas no numeradas, y por lo tanto son asimilables a capacidad de canal. Que las transmisiones realizadas durante estos intervalos de tiempos se conviertan en capacidad efectiva del enlace (es decir de cuantos bits por segundo se consiguen transportar realmente) depende de la probabilidad de error de este enlace.

El porcentaje de ocupación del canal (OC) es medido por la máquina de medición de nivel de enlace (MEASUREMENT MACHINE, ver apartado 5.2), y la probabilidad de error también es conocida (ver apartado 5.5).

La capacidad del canal es:

$$C_j = F_j p + P_j p \times B_i \times (1 - OC_i) \times a \quad (5.1)$$

Siendo:

$F_j p$  El parámetro primario  $F_j p$  (tasa de bits actualmente transportada por el canal).

$P_j p$  El parámetro primario  $P_j p$  (probabilidad de no error de paquete).

$B_i$  Capacidad bruta del canal (en bps); por ejemplo 1200 bps, 9600 bps, 19200 bps.

$OC_i$  Porcentaje de tiempo que el canal está ocupado.

$a$  Relación de eficiencia del nivel de enlace. Se trata de la relación:

$$\begin{aligned}
 a &= \frac{\text{Número de bits de datos de usuario por paquete ax.25}}{\text{Número de bits total por paquete ax.25}} \\
 &= \frac{\text{Número de bits de datos de usuario por paquete ax.25}}{\text{Número de bits de datos de usuario por paquete ax.25} + \text{tara nivel de enlace}}
 \end{aligned}$$

Esta relación permite expresar la relación entre la capacidad del canal y la capacidad real ofrecida por el nivel de enlace. Depende de la longitud media de los paquetes. Para una longitud de 256 octetos de usuario, en modo de ventana no extendida, tenemos:

$$\begin{aligned}
 a &= \frac{8 \times 256}{8 \times 256 + [\text{flags}]2 \times 8 + [\text{address}]14 \times 8 + [\text{control}]8 + [\text{pid}]8 + [\text{fcs}]16} \\
 &= 0.93
 \end{aligned}$$

En la ecuación  $C_j = Fj_p + Pj_p \times B_i \times (1 - OC_i) \times a$  cabe destacar lo siguiente:

Se suma  $Fj_p$  a la capacidad del canal  $Pj_p \times B_i \times (1 - OC_i) \times a$ . Esto se debe a que  $(1 - OC_i)$  es el porcentaje de tiempo que el canal está libre, y que por lo tanto  $Pj_p \times B_i \times (1 - OC_i) \times a$  es la capacidad teórica que el nivel de enlace puede ofrecer además del tráfico que ya transporta actualmente; es por lo tanto necesario sumar  $Fj_p$  para obtener la capacidad total que el enlace ofrece.

### 5.7.3.1 Procedimiento

La máquina de gestión de red del nodo *origen* del enlace solicita con periodo  $T_{req}$  a la máquina de medición de enlace a nivel de enlace la tasa de ocupación del canal  $OC_i$ , y obtiene el parámetro primario  $Hj_p$ , que llamamos “ancho de banda disponible de canal visto desde el transmisor”:

$$Hj_p = B_i \times (1 - OC_i) \times a$$

A continuación  $Hj_p$  se procesa mediante los procedimientos generales descritos en el apartado 5.4.

Queremos destacar que  $Hj_p$  se mide en el nodo *origen*, y no en el nodo destino como  $Pj_p$  y  $Fj_p$ . De haberse hecho en el nodo destino, es probable que  $OC_i$  fuera diferente<sup>26</sup> y  $Hj_p$  hubiera sido el “ancho de banda disponible de canal visto desde el receptor”. No obstante, en enlaces unidireccionales con equipos que sólo pueden transmitir (ver apartado 3.1) y no recibir,  $Hj$  no puede medirse en el nodo *origen*, debe medirse en el nodo destino.

La función característica  $N$  que permite que cada nodo calcule la capacidad  $C_j$  del canal es:

$$\begin{aligned}
 C_j &= N(Fj_p, Pj_p, Hj_p) \\
 N(Fj_p, Pj_p, Hj_p) &= Fj_p + Pj_p \times Hj_p
 \end{aligned}$$

<sup>26</sup>Debido a que el nodo origen y nodo destino pueden no recibir el mismo tráfico. Ver la descripción del problema de la estación oculta, apartado 1.2.

# Capítulo 6

## Control de flujo

Jnet utiliza simultáneamente dos técnicas complementarias de control de flujo: el control de acceso a la red, y el control de flujo interno (o “nodo a nodo”). Ambas técnicas son descritas en las publicaciones “Data Networks”[5] y “Protocols & Techniques for Data Communication Networks”[11].

### 6.1 Control de acceso a la red

#### 6.1.1 Objetivos

En Jnet, no se especifica un protocolo de acceso a la red para usuarios finales. En las referencias posteriores, con “usuario” del nivel de red nos referimos al nivel de transporte o a cualquier otro sistema que se comuniquen con el nodo a través de las primitivas de la capa de nivel de red de Jnet; el interfaz con el usuario es el conjunto de estas primitivas. Esto no excluye la utilización de protocolos específicos de acceso al nodo, para ello bastará diseñar un sistema traductor que transforme la interacción con el usuario final mediante estos protocolos en primitivas del nivel de red.

La función del control de acceso a la red es limitar el tráfico inyectado en la red por los usuarios, actuando directamente o inmediatamente después del interfaz con estos. En Jnet, esta función es realizada por el subsistema de acceso a la red, mediante la Máquina de control de flujo de acceso a la red (ver capítulo 4, apartado 4.2).

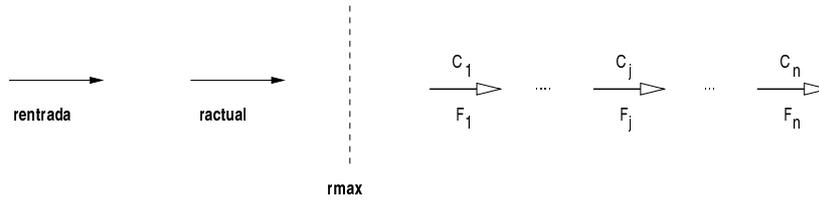
El control de flujo de acceso a la red se realiza independientemente para cada circuito virtual. El subsistema de acceso a la red conoce los parámetros característicos de los enlaces que forman cada circuito virtual, y en función de ellos realiza su tarea.

En la figura 6.1 representamos la situación:

El nodo conoce los parámetros característicos de los enlaces utilizados por el circuito virtual  $(C_1, C_2, \dots, C_n, F_1, F_2, \dots, F_n)$ .

**rentrada** es el tráfico que el usuario ofrece al nodo para este circuito virtual.

Figura 6.1: Esquema representativo



**ractual** es el tráfico actualmente inyectado en el circuito virtual.

**rmax** es el máximo tráfico inyectable en el circuito virtual.

El subsistema de acceso a la red debe:

- Calcular  $r_{max}$  en función de  $(C_1, C_2, \dots, C_n, F_1, F_2, \dots, F_n)$ .
- Limitar el tráfico inyectado al circuito virtual, tal que  $r_{actual} \leq r_{max}$ .

### 6.1.2 Control de flujo de Gallager

El control de flujo de acceso a la red se realiza una vez que el circuito virtual ha sido establecido. Como vemos en el capítulo 7 (encaminamiento), se elige el camino que tenga el mayor ancho de banda disponible. Es por lo tanto un criterio de maximización de tráfico.

Pero un nivel de red no puede centrarse en la maximización del tráfico: ello supondría una tendencia a utilizar toda la capacidad de los enlaces de la red, es decir una tendencia natural al colapso de los recursos. La meta del control de flujo es evitar esta situación de colapso, y para ello utiliza criterios cuya finalidad no es utilizar al máximo la capacidad de los enlaces.

Jnet usa el control de flujo de Gallager, que viene enmarcado dentro de una teoría conjunta de encaminamiento y control de flujo óptimo (ver [5]). En nuestro caso el control de flujo interviene sobre un circuito virtual ya prefijado y sin multiencaaminamiento (no se reparte el tráfico de un circuito virtual en varios caminos), y la aplicación de la teoría conjunta de Gallager da un algoritmo de control de flujo que se ajusta a la problemática. Seguiremos el enfoque del artículo [12], por ser más claro y más adaptado a este caso.

Se definen las siguientes variables de interés:

$D_{Tj}(F_j, C_j)$  Número medio de bits presentes en el enlace  $j$ . Utilizaremos la expresión simplificada para M/M/1 :

$$D_T(F_j, C_j) = \frac{F_j}{C_j - F_j} + d_j \times F_j$$

$C_j, F_j$  Son la capacidad y la tasa de datos transportados por el enlace  $j$ , es decir sus parámetros característicos.

$d_j$  Es el retardo de propagación del canal o de procesamiento del nodo. En la práctica es un parámetro desconocido, salvo para enlaces por satélite. Es interesante tenerlo en cuenta en Jnet, en vista a realizar experimentos con satélites de radioaficionados. No obstante, se trata de un caso aislado, y se ha decidido no incluir  $d_j$  en la lista de parámetros característicos de los enlaces.

$\frac{\delta D_{Tj}}{\delta r_{actual_i}}(F_j, C_j)$  Retardo incremental del enlace  $j$ . Es el incremento de  $Dt$  debido al incremento del tráfico del circuito virtual  $i$ .

$\frac{\delta D_T}{\delta r_{actual_i}} = \sum_{j=1}^N \frac{\delta D_{Tj}}{\delta r_{actual_i}}$  Retardo incremental total del circuito virtual  $i$  (suma de los retardos incrementales de los enlaces que participan en el circuito virtual  $i$ ).

Aunque dimensionalmente los retardos incrementales no sean tiempos, sino un incremento del número de bits, el comportamiento de estas variables define el retardo real de los enlaces; definen la sensibilidad del retardo del enlace a los incrementos de tráfico.

El control de flujo de Gallager consiste en limitar el tráfico inyectado de forma que el retardo incremental total no supere un umbral  $L_0$ . Además, hace variable este umbral en función del punto de trabajo ( $r_{actual}$ ), de forma que se consigue el punto de trabajo  $r_{max}$  óptimo, y que hace equitativo el reparto de la red (ver [12]).

En algoritmo de Gallager modificado por Thaker G. y Cain J. [12] consiste en calcular  $r_{max}$  tal que  $L_0(r_{max}) = \frac{\delta D_T}{\delta r}(r_{actual})$ .

### 6.1.3 Retardo incremental total del circuito virtual

El flujo de un enlace puede descomponerse de la forma:

$$F_j = r_{actual} + K_j$$

Siendo  $K_j$  el tráfico que no proviene del circuito virtual estudiado. Por lo tanto:

$$D_{Tj}(F_j, C_j) = \frac{r_{actual} + K_j}{C_j - r_{actual} - K_j} + (r_{actual} + K_j) \times d_j$$

Vamos a suponer que  $\frac{\delta K_j}{\delta r_{actual}} = 0$ . Por lo tanto:

$$\begin{aligned} \frac{\delta D_{Tj}(F_j, C_j)}{\delta r_{actual}} &= \frac{1 \times (C_j - F_j) - F_j \times (-1)}{(C_j - F_j)^2} + d_j \\ &= \frac{C_j}{(C_j - F_j)^2} + d_j \end{aligned}$$

En consecuencia el retardo incremental total del circuito virtual es:

$$\frac{\delta D_T}{\delta r_{actual}} = \sum_{j=1}^N \frac{C_j}{(C_j - F_j)^2} + \sum_{j=1}^N d_j$$

### 6.1.4 Función umbral del retardo incremental total

En este apartado definimos la función  $L(r_{actual})$  umbral superior del retardo incremental del circuito virtual ( $\frac{\delta D_T}{\delta r_{actual}}$ ). Utilizamos la definición dada en “Interactions Between Routing and Flow Control Algorithms” [12], en la que, a diferencia del análisis de “Data Networks” [5],  $L(r_{actual})$  no tiende a infinito cuando  $r_{actual}$  tiende a cero. Esto hace de ella una curva aplicable a casos reales, sin peligro de desbordamientos, y con puntos característicos que permiten obtenerla de forma práctica. No obstante, el análisis realizado en [12] sólo se aplica a redes cuyos enlaces tienen todos la misma capacidad. A continuación aplicaremos la teoría, adaptándola a nuestro caso particular.

$L(r_{actual})$  se define mediante 2 puntos característicos:

**(0,Lo):** Lo es el máximo absoluto de  $L(r_{actual})$ , define el techo de algunos parámetros de los enlaces. Es posible relacionar Lo con el máximo factor de utilización  $\rho_m$  de un enlace, mediante la relación:

$$Lo = \frac{d(\rho_m)}{C}$$

$d(\rho_m)$  es una función que se representa en la figura 6.3. De peculiar interés es el enlace de menor capacidad del circuito virtual,  $C_{min} = \min(C_j)$ , ya que en él nos interesa particularmente limitar el factor de utilización máximo  $\rho_m$ . En este caso:

$$Lo = \frac{d(\rho_m)}{C_{min}}$$

No obstante hay que destacar que el factor de utilización máximo  $\rho_{mj}$  en el resto de los enlaces es superior a  $\rho_m$ , en efecto:

$$\begin{cases} d(\rho_m) = C_{min} \times Lo \\ d(\rho_{mj}) = C_j \times Lo \Rightarrow \rho_m \leq \rho_{mj} \\ C_{min} \leq C_j \end{cases}$$

Para garantizar un factor de utilización máximo en todos los enlaces, hay que elegir  $Lo = \frac{d(\rho_m)}{C_{max}}$ . Pero ello supone una fuerte limitación del factor de utilización de los

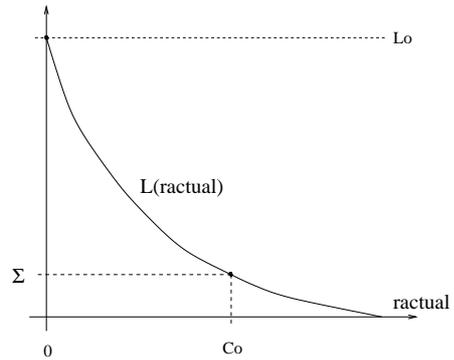


Figura 6.2: Perfil de L(r)

enlaces de baja capacidad, y puesto que estos son los enlaces que limitan el tráfico en el circuito virtual, la tasa binaria inyectada en este resultaría mínima. En concreto para el enlace de menor capacidad:  $d(\rho_{mj}) = \frac{C_{min}}{C_{max}} L_o$ . No es nuestro interés limitar el factor de utilización de todos los enlaces, sino el del enlace de menor capacidad, ya que este es crítico por ser el cuello de botella del circuito virtual.

En Jnet elegimos el siguiente valor  $\rho_m = 0.8 \Rightarrow d(\rho_m) = 25$ . Por lo tanto:

$$L_o = \frac{d(\rho_m)}{C_{min}} = \frac{25}{C_{min}}$$

**(Co,ε):** Es el segundo punto característicos. Representa la situación en la que la capacidad ofrecida por el sistema de control de flujo es máxima. En Jnet, se decide que la capacidad máxima ofrecida es:

$$C_o = C_{min}$$

Y sólo se ofrece en la situación del retardo incremental mínimo posible:

$$\frac{\delta D_T}{\delta r_{actual}} \text{mínimo} \Leftrightarrow F_j = 0 \Rightarrow \frac{\delta D_T}{\delta r_{actual}} = \sum \frac{1}{C_j} + \sum d_j$$

Es decir:

$$L(C_o) = \varepsilon = \sum \frac{1}{C_j} + \sum d_j$$

En una red sin enlaces por satélite, el segundo término no aparece.

Nota: Debe cumplirse que  $L_o > \varepsilon$ . En caso contrario se escoge  $\varepsilon = L_o$ , la curva  $L(r_{actual})$  es un segmento horizontal.

### Cálculo de L(r)

La ecuación de la función umbral del retardo incremental total puede ser expresada de la forma:

$$L(r) = \frac{A}{B+r} + D$$

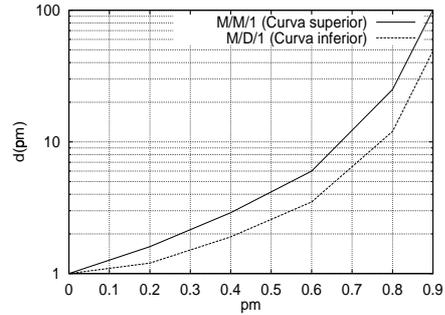


Figura 6.3:  $d(\rho_m)$

Tenemos 3 constantes  $A, B, D$  a definir, y necesitamos por lo tanto 3 puntos de la curva. Anteriormente hemos definido 2 puntos ( $L(0) = L_o$ ,  $L(C_{min}) = \varepsilon$ ). El tercer punto es:

$$L\left(\frac{C_{min}}{10}\right) = L_o - \frac{L_o - \varepsilon}{2}$$

Se desea que  $L(r)$  decrezca rápidamente, para hacer más equitativo el reparto de la red (penalizando los tráficos de mayor caudal). La fijación del tercer punto permite dar este perfil decreciente acentuado (en la décima parte de  $C_{min}$ ,  $L(r)$  ha disminuido la mitad, la otra mitad se recorre en las 9 décimas partes restantes).

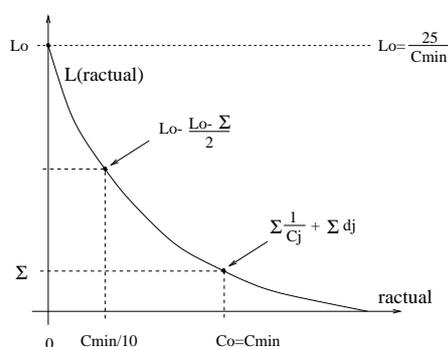


Figura 6.4: Los 3 puntos característicos de  $L(r)$

Partimos de estas 3 ecuaciones:

$$L(r) = \frac{A}{B+r} + D$$

$$L(0) = \frac{25}{C_{min}} = \frac{A}{B} + D = L_o \quad (1)$$

$$L(C_{min}) = \varepsilon = \frac{A}{B+C_{min}} + D \quad (2)$$

$$L\left(\frac{C_{min}}{10}\right) = L_o - \frac{L_o - \varepsilon}{2} = \frac{A}{B + \frac{C_{min}}{10}} + D = K \quad (3)$$

En el apéndice E despejamos  $A, B, D$ , y obtenemos:

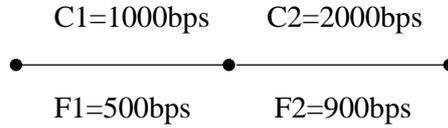
$$B = \frac{C_{min}}{8} \quad (5)$$

$$A = \frac{9C_{min}}{64}(L_o - \varepsilon) \quad (6)$$

$$D = \frac{9\varepsilon - L_o}{8} \quad (7)$$

### 6.1.5 Ejemplo

Analizaremos un caso sencillo, un circuito virtual formado por dos enlaces:



$$L_o = \frac{25}{C_{min}} = \frac{25}{1000} = 0.025$$

$$\varepsilon = \sum \frac{1}{C_j} = \frac{1}{1000} + \frac{1}{2000} = 0.0015$$

Los parámetros de la curva  $L(r)$ :

$$B = \frac{C_{min}}{8} = \frac{1000}{8} = 125$$

$$A = \frac{9C_{min}}{64}(L_o - \varepsilon) = \frac{9000}{64}(0.025 - 0.0015) = 3.3046$$

$$D = \frac{9\varepsilon - L_o}{8} = \frac{9 \times 0.0015 - 0.025}{8} = -0.0014$$

El retardo incremental:

$$\frac{\delta D_T}{\delta r} = \sum_{j=1}^N \frac{C_j}{(C_j - F_j)^2} = \frac{1000}{(1000 - 500)^2} + \frac{2000}{(2000 - 900)^2} = 0.00565$$

Control de flujo:

$$L_o(r_{max}) = \frac{\delta D_T}{\delta r} \Leftrightarrow \frac{A}{B + r_{max}} + D = \frac{\delta D_T}{\delta r} \Leftrightarrow r_{max} = \frac{A}{\frac{\delta D_T}{\delta r} - D} - B.$$

$$r_{max} = 341bps$$

### 6.1.6 Aplicación a Jnet

El método de control de flujo de Gallager/Thaker/Cain no es eficiente en una red de radio-paquete, debido a los retardos y a la escasez de ancho de banda.

El retardo incremental  $\frac{\delta D_T}{\delta r}$  es función del tráfico inyectado  $r_{actual}$ , porque  $r_{actual}$  está incluido en el término  $F_j$ . Al producirse un incremento de tráfico  $\Delta r_{actual}$  dentro del límite permitido por  $r_{max}$ ,  $F_j$  se incrementará, y en consecuencia también lo hará  $\frac{\delta D_T}{\delta r}$ . Cambiará el punto de trabajo, y en consecuencia también  $r_{max}$ . Si el usuario no utiliza control de flujo a nivel de transporte (por ejemplo TCP), su tasa binaria será siempre la impuesta por el nivel de red,  $r_{max}$ , que cambiará hasta converger en un punto de equilibrio.

La principal problemática de este método reside en que frente a cambios instantáneos de la demanda del usuario, la evaluación de  $r_{max}$  no es óptima y se corrige mediante el proceso de realimentación descrito en el párrafo anterior. En Jnet el procedimiento de procesamiento y difusión de los parámetros primarios es orientado al ahorro de

ancho de banda (con filtrados, cuantificaciones, difusiones controladas), es lento.  $r_{max}$  tardaría demasiado en converger hacia el valor óptimo, esto significa que *el método de control de flujo de Gallager/Thaker/Cain en el caso de Jnet no previene convenientemente la congestión de la red*, puesto que durante el (largo) tiempo de convergencia  $r_{max}$  es superior al valor óptimo, el que se alcanza al equilibrio.

Esta problemática se soluciona modificando el método, añadiéndole condiciones preventivas que lo hagan converger rápidamente  $r_{max}$  al valor óptimo:

El valor de  $r_{max}$  se obtiene en función de  $\frac{\delta D_T}{\delta r}$ , que a su vez es función de  $(C_j, F_j)$ . Pero  $F_j$  cambia con el tráfico inyectado. Vamos a prever la variación de  $F_j$  suponiendo que:

- Con el tráfico actualmente inyectado  $r_{actual}$  el tráfico actual de los enlaces es  $F_j$ .
- Con un tráfico  $r_{actual} + \Delta r$ , el tráfico de los enlaces sería  $F_j' = F_j + \Delta r$ .

Se trata de calcular  $\Delta r$  tal que:

$$\frac{\delta D_T}{\delta r}(r_{actual} + \Delta r) = L(r_{actual} + \Delta r) \quad (8)$$

El flujo máximo autorizado  $r_{max}$  es  $r_{actual} + \Delta r$ , siendo  $\Delta r$  la solución a la anterior ecuación.

6.1.6.1 Cálculo de  $r_{max}$

En este apartado especificamos el procedimiento de cálculo de  $r_{max}$ .

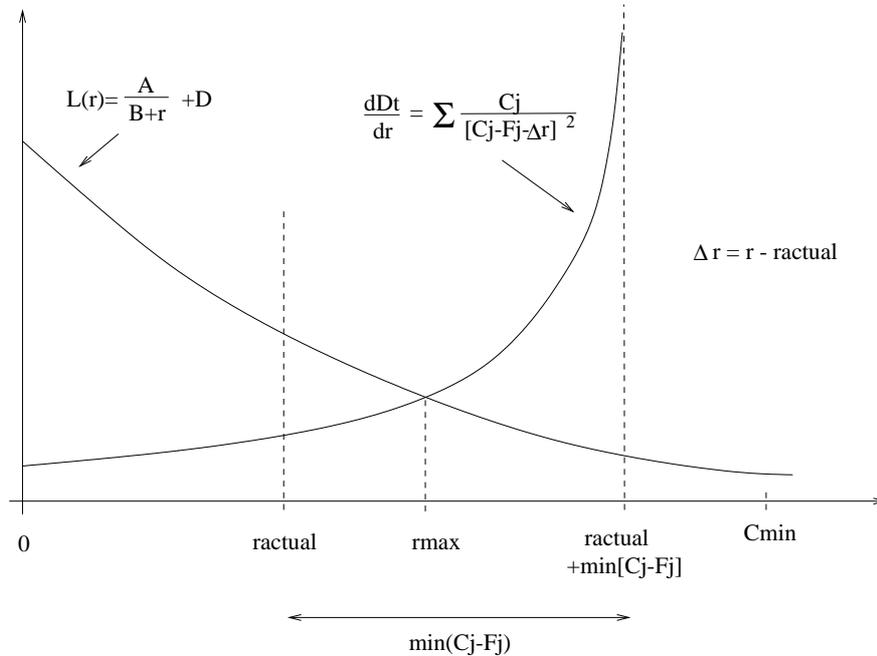


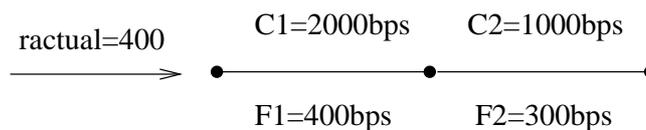
Figura 6.5: Cálculo de  $r_{max}$

En la figura 6.5 representamos gráficamente la resolución de  $r_{max}$ . Se trata de hallar la intersección de las curvas  $L(r) = \frac{A}{b+r} + D$  y  $\frac{\delta D_t}{\delta r}(r) = \sum \frac{C_j}{(C_j - F_j - (r - r_{actual}))^2}$ , siendo  $\Delta r = r - r_{actual}$ .

Es fácilmente demostrable que la intersección existe y es única, puesto que  $L(r)$  es acotada y decreciente, mientras que  $\frac{\delta D_t}{\delta r}(r)$  es creciente y  $\lim_{r \rightarrow r_{actual} + \min(C_j - F_j)} (\frac{\delta D_t}{\delta r}(r)) = \infty$ .

El intervalo de trabajo es  $[0, r_{actual} + \min(C_j - F_j)[$ , puesto que en ningún caso se debe superar el mínimo caudal disponible  $C_j - F_j$  de los enlaces.

Finalmente, también hay que recalcar que *en la práctica* es posible el caso  $C_{min} < r_{actual} + \min(C_j - F_j)$ , debido a los retardos de la red (esto nos permite reducir el extremo superior del intervalo de búsqueda de soluciones al mínimo de estos dos valores). Por ejemplo en la siguiente configuración:



$$\min(C_j - F_j) = C_2 - F_2 = 1000 - 300 = 700 \Rightarrow r_{actual} + \min(C_j - F_j) = 400 + 700 = 1100$$

$$C_{min} = 1000 \Rightarrow C_{min} < r_{actual} + \min(C_j - F_j)$$

Este caso es posible por ejemplo porque el tráfico  $r_{actual} = 400$  aún no ha llegado al enlace 2, o bien porque el parámetro primario aún no ha llegado al nodo origen.

La resolución de la ecuación de  $r_{max}$  se hace mediante un procedimiento numérico muy simple, consistente en recorrer el intervalo  $[0, \min(C_{min}, \min(C_j - F_j) + r_{actual})]$  hasta que  $L(r) < \frac{\delta D_T}{\delta r}$ , mediante los siguientes pasos:

- Se divide el intervalo  $[0, \min(C_{min}, \min(C_j - F_j) + r_{actual})]$  en 100 subintervalos:

$$\Delta x = \frac{\min(C_{min}, \min(C_j - F_j) + r_{actual})}{100}$$

- Realizar bucle con variable contador  $i=0$  incrementándose hasta  $i=99$ ; salir del bucle si:

$$\sum \frac{C_j}{(C_j - F_j - [i \times \Delta x - r_{actual}])^2} \geq \frac{A}{B + i \times \Delta x} + D$$

- $r_{max} = i \times \Delta x$

### Notas sobre posibles desbordamientos de cálculo

- Como  $L(r)$  es decreciente, sólo puede desbordarse en  $L(0)$  (el máximo).  $L(0) = \frac{d(\rho_m)}{C_{min}} = \frac{25}{C_{min}}$ . Siguiendo la codificación propuesta en el apartado 3.3.3, los dos valores más bajos posibles de  $C_{min}$  son 10 bps y 0 bps. El primer valor no causa problemas: el algoritmo sólo debe ejecutarse si  $C_{min} \neq 0$ . Si  $C_{min} = 0$  no se ejecuta el algoritmo, y se fija  $r_{max} = 0$ .
- $\frac{\delta D_T}{\delta r}$  está acotada por  $L(r)$  en el intervalo  $[0, (i-1) \times \Delta x]$ , por lo tanto no puede desbordarse si  $L(r)$  no se desborda. El único punto con peligro es  $r_{max} = i \times \Delta x$ , ya que en él  $\frac{\delta D_T}{\delta r} > L(r)$ .

6.1.6.2 Dinámica y comportamiento frente a cambios

Vamos a estudiar mediante un ejemplo el comportamiento dinámico del sistema de control de acceso a la red de Jnet, observando como varían las curvas  $\frac{\delta D_T}{\delta r}$  y  $L(r)$  en dos situaciones: cambio de  $r_{actual}$ , y cambio de  $F_j$ .

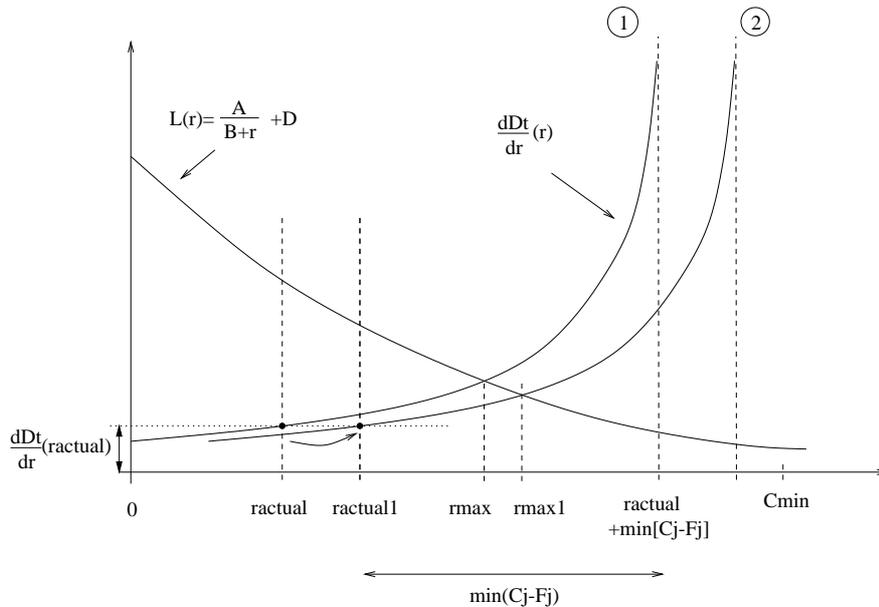


Figura 6.6: Cambio de tráfico

En la figura 6.6 representamos un cambio “instantáneo” del tráfico introducido en la red. En el instante del cambio, no se detecta ningún cambio en los parámetros primarios del tráfico de la red. La curva  $L(r)$  no cambia, pero sí lo hace la curva  $\frac{\delta D_T}{\delta r}(r)$  (paso de la curva (1) a la curva (2)) con una translación horizontal igual a la variación del tráfico. Por lo tanto también cambia la intersección de las dos curvas, el valor de  $r_{max}$  es diferente.

Pero hay que destacar que esta variación es transitoria e inexacta, puesto que aún no se tiene en cuenta el aumento del retardo incremental causado por el cambio de tráfico. Pasada la situación transitoria, es muy probable que el valor  $r_{max}$  sea muy parecido al inicial, al incrementarse  $\frac{\delta D_T}{\delta r}(r_{actual})$  el punto de trabajo  $(r_{actual}, \frac{\delta D_T}{\delta r}(r_{actual}))$  tiende a encontrarse sobre la curva inicial. Este ejemplo ilustra un caso en el que es peor recalcular  $r_{actual}$ , ya que se tiene la certeza de estar frente a una situación transitoria. En conclusión:

No se va a calcular de nuevo  $r_{max}$  cuando cambie  $r_{actual}$ .

En cambio, no es posible realizar ninguna afirmación sobre las causas de las variaciones de los parámetros  $C_j$  y  $F_j$  de los enlaces del circuito virtual, que pueden ser provocadas por el aumento de  $r_{max}$ , por causas externas, o ambas a la vez. Por este

motivo:

$r_{max}$  se calcula de nuevo cuando cambie  $C_j$  o  $F_j$ .

### 6.1.7 Medición de $r_{actual}$ y regulación del tráfico

#### Regulación del tráfico

Destacamos un punto al que hay que prestar atención: los parámetros primarios  $C_j$  y  $F_j$  representan tasas de bits transportadas por los niveles de enlace. En el análisis anterior, hemos considerado que el tráfico entregado  $r_{actual}$  al nivel de red iba a poder ser transportado por el circuito virtual dados sus parámetros primarios. En la práctica no es así, porque el nivel de red introduce una tara de información en cada paquete entregado al nivel de enlace (cabeceras del nivel de red). El flujo de bits entrante es menor que el flujo de bits saliente.

Se define la relación de eficiencia del nivel de red  $J_{net}$ , mediante la relación:

$$b = \frac{\text{Número de bits de datos de usuario por paquete } J_{net}}{\text{Número de bits total por paquete } J_{net}}$$

$$= \frac{\text{Número de bits de datos de usuario por paquete } J_{net}}{\text{Número de bits de datos de usuario por paquete } J_{net} + \text{tara nivel de red}}$$

$b$  depende de la implementación del protocolo  $J_{net}$ .

*Por lo tanto, la tasa de información entregada a  $J_{net}$  por el usuario (nivel de transporte) debe ser limitada al valor  $r'_{max} = r_{max} \times b$ .*

La limitación del tráfico de entrada a la red se realiza con un algoritmo de “leaky bucket” (cubo agujereado), que *modificamos* para funcionar con pds de tamaño variable.

Las constantes y variables utilizadas en el algoritmo son:

$T = \frac{1}{r_{max}}$ : tiempo de bit (s/bit).

$K$  : bloque de bits (bits). Se trata del tamaño de los bloques liberados por el algoritmo, es decir el tamaño del “agujero del cubo”. El valor de  $K^1$  debe ser del orden o un poco superior al tamaño medio de los paquetes. Por ejemplo  $K=500$ .

$C$  : Crédito de bits (bits). Límite de bits transmisibles en el instante actual.

$L$  : Tamaño de la pdu a transmitir (bits).

<sup>1</sup>“Tamaño de bloque de leaky bucket para acceso a la red”. Se trata de un parámetro local del nodo. Ver apéndice B.

El algoritmo es el siguiente:

1.  $C=0$
2. Sacar NPDU de la cola de entrada (= primitiva N-DATA Request proveniente del nivel de transporte).  $L$ =longitud del campo de datos.
3. Esperar ( $T \times K$ )
4.  $C=C+K$
5. ¿  $C>L$  ?  
 Si:
  - (a) Entregar los datos al nivel de enlace (= primitiva SEG-DATA Request).
  - (b)  $C=C-K$
  - (c) Sacar NPDU de la cola de entrada (= primitiva N-DATA Request proveniente del nivel de transporte).  $L$ =longitud del campo de datos.
  - (d) Volver al punto 5.

No:

- (a) Volver al punto 3.

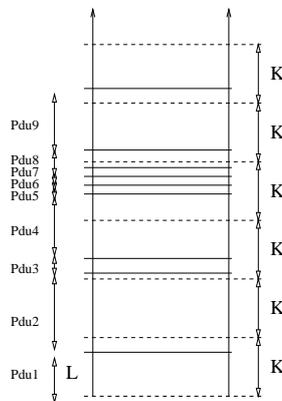


Figura 6.7: Leaky bucket

En la figura 6.7 representamos el funcionamiento del algoritmo. Se representan los paquetes en líneas continuas, y el incremento de créditos en líneas discontinuas.

### **Medición**

La medición de  $r_{actual}$  se realiza a la salida del módulo de regulación de tráfico. Por lo tanto, se garantiza que  $r_{actual} \leq r_{max}$ . No se requiere ningún filtrado especial, pudiéndose realizar la medición en una ventana de tiempo del orden de 1 minuto.

Con estas puntualizaciones, cerramos el capítulo de control de acceso a la red. En el apéndice F se encuentra el código fuente en lenguaje C de la implementación del algoritmo de resolución numérica de  $r_{max}$ .

## 6.2 Control de flujo interno

### 6.2.1 Objetivos y necesidades

El control de flujo interno actúa en el transporte de información de usuarios que se produce entre los nodos de la red, con la misma finalidad que el control de acceso a la red: evitar la congestión, y si esta se produce procesarla de forma que sus consecuencias sean mínimas.

Hemos observado que el control de acceso a la red utiliza un mecanismo preventivo potente, pero lento, porque utiliza la señalización de mapa de red (parámetros característicos  $C_j$  y  $F_j$ ) difundida por necesidad de optimización del ancho de banda mediante un mecanismo lento. Se trata de un *mecanismo poco reactivo, insuficiente* para reaccionar frente a acontecimientos rápidos generadores de congestión, como ocurre frecuentemente en entornos radioeléctricos (interferencias radioeléctricas, fenómenos naturales que deterioran las instalaciones, variaciones de las condiciones de propagación de las ondas, etc.).

El control de flujo interno va a suplir estas carencias. Para ello, necesita utilizar un mecanismo de señalización más rápido que el provisto por el mapa de red, un mecanismo no de difusión, sino *dirigido* directamente a los nodos que puedan tener un papel activo en la disminución de la congestión y de sus efectos perjudiciales. ¿Cuál es este mecanismo? ¿Cómo encamina la señalización? A estas dos preguntas contestamos a continuación.

### 6.2.2 Modelo elegido: back pressure

El sistema de control de flujo interno sigue el modelo de *back pressure* (ver características y ventajas de este sistema en “Data Networks” [5]). Cada circuito virtual se verá sometido independientemente a un algoritmo de back pressure, siendo la señalización necesaria transportada dentro del mismo circuito virtual. Esto soluciona en gran medida el problema de encaminamiento de la señalización, puesto que se va a utilizar una única ruta predefinida, la misma que sigue la información de usuario dentro del circuito virtual. El control de flujo se hace en cada nodo, dentro de cada máquina de circuito virtual (asignada a un solo circuito virtual, ver apartado 4.4).

No obstante, existe una problemática: el modelo de back pressure requiere un mecanismo de señalización hacia atrás. Y como se ha descrito en el apartado 3.4.2, los circuitos virtuales pueden tener enlaces unidireccionales que hacen imposible la comunicación directa hacia atrás.

La solución a esta problemática pasa por utilizar el circuito virtual de vuelta cuando estemos utilizando un circuito virtual bidireccional. En caso de circuitos virtuales unidireccionales, sólo será posible el control de flujo interno cuando todos los enlaces que lo componen sean con conexión hdlc. Por lo tanto hay que tener cuidado en los circuitos virtuales unidireccionales con enlaces sin conexión hdlc, ya que pueden congestionar la red; es un aspecto sujeto a futuras modificaciones y mejoras.

Por cada sentido de circuito virtual, se establece una topología de encaminamiento de señalización, en la que tendrán una gran importancia los bucles formados entre el circuito virtual de ida y de vuelta, llegándose a una clara clasificación de tipos de nodos y de su papel según su posición.

### 6.2.3 Topología

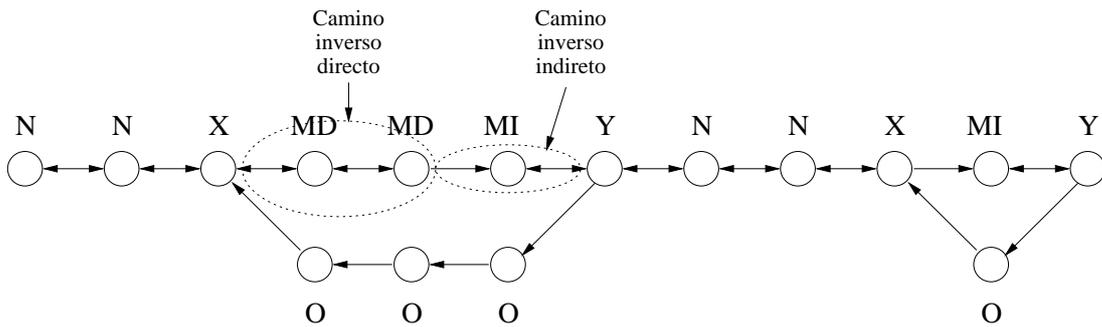


Figura 6.8: Tipos de nodos

Se definen los siguiente tipos de nodos:

**N** Nodo que se encuentra en el circuito virtual de ida y de vuelta.

**M** Nodo que sólo se encuentra en el circuito virtual de ida. Se definen dos tipos:

**MD** Nodo M que pertenece al camino inverso directo.

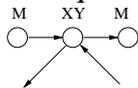
**MI** Nodo M que pertenece al camino inverso indirecto.

**O** Nodo que sólo se encuentra en el camino de vuelta.

**X** Nodo N, cuyo nodo siguiente es de tipo M.

**Y** Nodo N, cuyo nodo anterior es de tipo M.

**XY** Nodo que es a la vez nodo X y nodo Y.



Los bucles están formados por nodos X, M, Y, O. Se definen dos segmentos de especial interés:

**Camino inverso directo** Esta formado por los nodos M capaces de alcanzar el nodo X anterior utilizando los enlaces utilizados por el circuito virtual directo, pero en sentido opuesto. En la figura 6.8 representamos esta situación; el “poder utilizar los enlaces en sentido opuesto” se marca con doble flecha, y significa:

- → Que el enlace es con conexión hdlc. La existencia de un enlace hdlc implica la posibilidad de enviar datos en sentido opuesto, aunque este sentido no sea utilizado por el circuito virtual para enviar datos de usuario.
- ← Que existe un enlace sin conexión hdlc hacia atrás.

Pueden darse diferentes combinaciones basadas en este esquema. En caso de posible elección entre enlace con conexión hdlc y sin conexión hdlc, se elegirá el enlace con conexión hdlc (para evitar pérdidas en la señalización).

**Camino inverso indirecto** Se trata del conjunto de nodos M que han quedado “aislados” del nodo X anterior, se encuentran a partir del nodo M que no tiene posibilidad de enviar datos a su nodo anterior.

Este modelo topológico es *fundamental*, porque cada tipo de nodo tiene un comportamiento y características diferentes. En el momento del establecimiento del circuito virtual bidireccional, el nodo debe automáticamente incluirse en una de estas categorías. Es por lo tanto necesario que en el momento del establecimiento los nodos conozcan las características del circuito virtual: identificadores de los nodos (para poder diferenciar los nodos, e identificar los nodos X e Y que pertenecen a ambos sentidos del circuito virtual) y el tipo de enlaces (con o sin conexión hdlc).

En general, será necesario un procedimiento de establecimiento de circuito virtual en 3 etapas:

1. Establecimiento del circuito virtual de ida: los nodos participantes son informados de las características de este circuito virtual.
2. Establecimiento del circuito virtual de vuelta: los nodos del circuito virtual de vuelta son informados de las características del circuito virtual de vuelta y de ida.
3. Los nodos del circuito virtual de ida son informados de las características del circuito virtual de vuelta.

La topología del circuito virtual de vuelta se define con los mismos criterios, y generalmente será diferente (por ejemplo los nodos O pasan a ser nodos M). Los procedimientos de control de flujo interno del circuito virtual de ida y de vuelta son los mismos, pero se realizan de forma independiente.

#### 6.2.4 Procedimientos generales

El sistema de control de flujo por back pressure consiste en la propagación hacia atrás de una onda de aviso de congestión, que ordena a todos los nodos que la reciben parar sus transmisiones (en el caso de Jnet, son las transmisiones de un circuito virtual determinado, no todas). Con la vuelta a la normalidad, se propaga hacia atrás otra onda informando que el tráfico puede de nuevo cursarse.

Por este motivo, la señalización de control de flujo se divide en dos fases: señalización de principio de congestión, y señalización de fin de congestión.

Más adelante (apartado 6.2.5.2) definiremos exactamente el *modo pressure*. El nodo tiene un comportamiento claro relativo al transporte de datos de usuario: en modo pressure, no transmite datos de usuario en el circuito virtual determinado; si no está en modo pressure, transmite normalmente los datos de usuario.

### 6.2.4.1 Back pressure fuera de bucles

El procedimiento general de la señalización de control de flujo entre nodos N, X, Y, XY se llama *back pressure directo*.

La fase de señalización de principio de congestión se compone de una señal de aviso de principio de congestión que llamamos “BACK PRESSURE DIRECTO”. Cuando un nodo recibe esta señal, inmediatamente la propaga hacia atrás (la envía al nodo anterior). El aviso de congestión debe ser asentido, mediante la señal “ACK BACK PRESSURE DIRECTO”, sea cual sea el tipo de enlace (con o sin conexión hdlc).

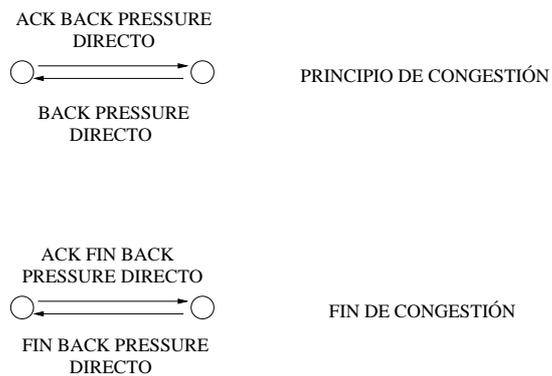


Figura 6.9: Back pressure directo

La fase de señalización de fin de congestión es similar a la fase de principio de congestión, con dos señales: “FIN BACK PRESSURE DIRECTO” y “ACK FIN BACK PRESSURE DIRECTO”.

### 6.2.4.2 Back pressure en bucles

En un bucle, llamaremos *nodo generador de señal de congestión* al primer nodo que informa a los nodos del bucle de una nueva congestión. Si es un nodo Y, es posible que esta noticia de congestión venga de un nodo posterior al bucle. Si se trata de un nodo MD o MI, entonces es que ha detectado una congestión en el enlace a su siguiente nodo. El nodo X no informa al resto del bucle de las congestiones que detecta, no hay necesidad alguna puesto que el bucle se encuentra después de él.

En las congestiones detectadas por los nodos MD, es decir en el camino inverso directo, se sigue el procedimiento de back pressure directo.

Pero en el camino inverso indirecto, debe utilizarse un procedimiento diferente, que llamamos *back pressure indirecto*.

La fase de señalización de principio de congestión por *back pressure indirecto* se compone de 5 etapas, de forma que el mensaje pueda dar una vuelta al bucle:

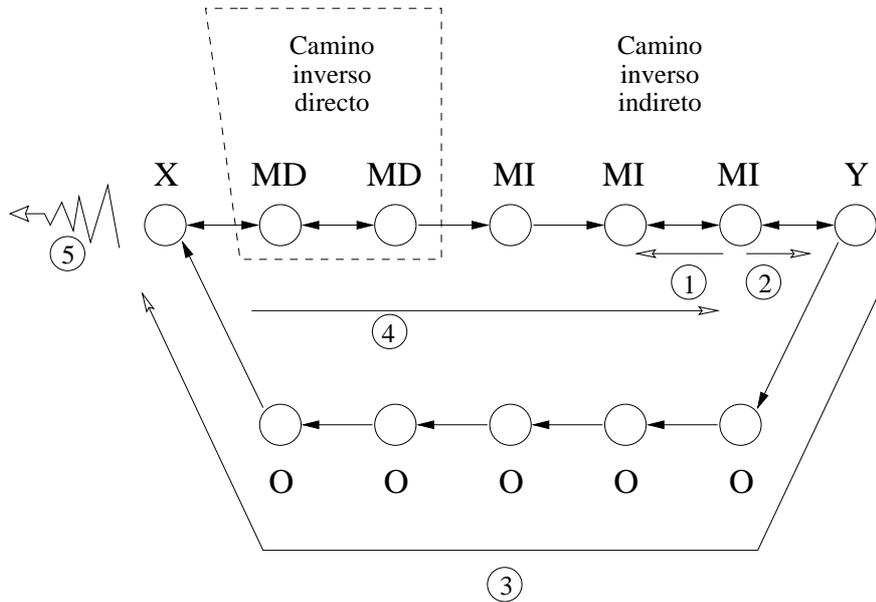


Figura 6.10: Back pressure indirecto

1. Back pressure directo por el camino inverso indirecto. Aunque sepamos que por esta vía no se consigue alcanzar al nodo X, es posible avisar algunos nodos MI anteriores utilizando los enlaces que nos comunican con ellos; para ello, se sigue el procedimiento de back pressure directo. Cuando un nodo MI recibe la señal de back pressure directo y no puede propagarla hacia atrás, se limita a mandar el asentimiento y tomar las acciones correspondientes al aviso de congestión.
2. El *nodo generador de señal de congestión* envía una señal que llamamos “FORWARD BACK PRESSURE INDIRECTO”, que se propaga en sentido del circuito virtual de ida, hasta llegar al nodo Y.
3. Al recibir la señalización anterior, el nodo Y genera una señal que llamamos “BACK PRESSURE INDIRECTO”, que se propaga por el circuito virtual de vuelta a través de los nodos O, hasta el nodo X.
4. Al recibir la señalización anterior, el nodo X genera una señal “ACK BACK PRESSURE INDIRECTO”, que se propaga por el circuito virtual de ida a través de los nodos MD y MI hasta llegar al *nodo generador de señal de congestión*.
5. El nodo X, además, propaga a los nodos anteriores al bucle el aviso de congestión (utilizando el procedimiento adecuado al tipo del nodo anterior).

La fase de señalización de fin de congestión es idéntica, pero con otros nombres:

1. Fin de congestión directo.
2. “FORWARD BACK PRESSURE INDIRECTO FIN DE CONGESTIÓN”.

3. “BACK PRESSURE INDIRECTO FIN DE CONGESTIÓN”.
4. “ACK BACK PRESSURE INDIRECTO FIN DE CONGESTIÓN”.
5. Propagación a los nodos anteriores del mensaje de fin de congestión.

Nota: Si en el bucle no hay nodos MI, sólo se sigue el procedimiento de back pressure directo.

### 6.2.4.3 Multienvío

En el caso de back pressure en bucles, la señal (2) “Forward back pressure indirecto” (o “forward back pressure indirecto fin de congestión”) es enviada varias veces por el *nodo generador de señal de congestión*, de forma que la probabilidad de que el mensaje dé una vuelta al bucle sea mayor que un valor  $A$ <sup>2</sup>. Para ello, el *nodo generador de señal de congestión* debe calcular la probabilidad de error del bucle, a partir de los parámetros primarios  $P_j$  de los enlaces (se considera que  $P_j = 1$  para enlaces con conexión hdlc). Los nodos intermedios (MI, Y, O, X, MD) retransmiten todas las copias de la señalización que reciben.

El procedimiento de back pressure directo también es sometido a multienvío. Pero esta vez, tanto el nodo generador de la señal (“back pressure directo” o “fin back pressure directo”) como el nodo receptor (el nodo que genera las señales de asentimiento “ack back pressure directo” o “ack fin back pressure directo”) realizan multienvío, transmitiendo la señalización el número de veces necesario para que llegue al extremo del enlace con una probabilidad mayor que  $A$ . A diferencia del procedimiento de back pressure indirecto, este procedimiento es local: la recepción de varias copias de una misma señal (es decir referida al mismo identificador de congestión) no debe generar varias copias de asentimientos, ni tampoco varias señales propagadas; el número de señales generadas es independiente del número de señales recibidas (con una basta, y si se reciben más debe hacerse caso omiso de ellas).

### 6.2.4.4 Temporizadores

El *nodo generador de señal de congestión* (en el procedimiento de back pressure indirecto) o el generador de la señal “back pressure directo” o “fin back pressure directo” (procedimiento back pressure directo) reinicia el proceso de envío (multienvío) de esta señalización si no recibe la señal de asentimiento correspondiente pasado un cierto tiempo definido por un temporizador.

Para cada circuito virtual, y para cada sentido de la comunicación (circuito virtual de ida y de vuelta) se definen dos temporizadores:

T1: Temporizador para procedimiento de back pressure directo.

T2: Temporizador para procedimiento de back pressure indirecto.

---

<sup>2</sup>“Probabilidad mínima de no pérdida de señalización en un bucle”. Se trata de un parámetro local del nodo. Ver apéndice B.

Se define localmente al nodo su valor inicial en dos variables T1INIT y T2INIT<sup>3</sup>. T1 y T2 se van ajustando al valor estimado de ida y vuelta (“Round Trip Time”), mediante el algoritmo de Phil Karn [13] o alguno equivalente (por ejemplo otros algoritmos utilizados en el protocolo TCP).

#### 6.2.4.5 Identificador de congestión

Pueden producirse congestiones en distintos nodos de un circuito virtual, y en diferentes instantes de tiempo. Es necesario poder diferenciar estas congestiones, el apartado anterior es un claro ejemplo: para estimar el tiempo de ida y vuelta, hay que estar seguro a qué congestión se refiere el asentimiento recibido. El *nodo generador de señal de congestión* debe ser capaz de reconocer los asentimientos a las ondas de back pressure generadas por él.

Esta diferenciación se hace mediante un *identificador de congestión*, generado por el *nodo detector de la congestión*. Este identificador debe ser único en el circuito virtual.

### 6.2.5 Optimización : ocultación de congestiones

#### 6.2.5.1 Introducción

En un circuito virtual susceptible a la congestión, es probable que varios enlaces se congestionen, generando una cantidad no despreciable de señalización de back pressure. Para optimizar los recursos, Jnet limita la señalización de back pressure ocultando ondas de back pressure a partes del circuito virtual donde esta información no es de utilidad.

La idea es la siguiente:

- Cuando un nodo se congestiona, genera una onda de back pressure hacia atrás en el circuito virtual. Si a su vez, este nodo congestionado recibe una onda de back pressure de un nodo que le sigue (que llamamos “nodo posterior”), anunciando otra congestión, la propagación de esta información no es de interés para el resto de nodos puesto que ya han tomado medidas con respecto a la congestión.
- Cuando este nodo se descongestiona, no puede enviar la onda de final de congestión hasta que todos los nodos posteriores se hayan descongestionado.
- Puede ocurrir que en el momento de la congestión, ya hubiera nodos posteriores congestionados. Esto debe tenerse en cuenta, puesto que todos los nodos que hayan recibido su onda de “principio de congestión” están esperando recibir su onda de “final de congestión”, no pudiéndose ocultar esta información. Debe ser por lo tanto un ocultamiento selectivo.

---

<sup>3</sup>“Valor inicial del temporizador de retransmisión por back pressure directo / indirecto”. Se trata de un parámetro local del nodo. Ver apéndice B.

Vemos por lo tanto que esta optimización va a suponer una mayor complejidad, justificada debido a la escasez de ancho de banda, o simplemente debido a la voluntad de eficiencia de este proceso en Jnet. Para poder llevarla a cabo, los nodos necesitan memorizar las congestiones recibidas con anterioridad, realizar su seguimiento, y tener diferentes estados de trabajo en los que se comportarán de forma diferente.

### 6.2.5.2 Listas e indicadores

En cada nodo y por cada sentido de circuito virtual se definen 3 listas:

1. *Lista de nodos congestionados*: Lista en la que se almacenan los identificadores de congestión de los nodos posteriores congestionados.
2. *Lista de nodos MD/MI congestionados*: En el caso de los nodos MD, lista en la que se almacenan los identificadores de congestión de los nodos MD posteriores congestionados del mismo bucle. En el caso de los nodos MI, lista de nodos MI posteriores que tienen comunicación con nosotros en modo back pressure directo.
3. *Lista de nodos congestionados antes que nosotros*: Lista los identificadores de congestión correspondientes a nodos posteriores que estaban ya congestionados en el momento en el que nuestro nodo se congestiona. También se la llama *lista translúcida*.

A su vez, se definen 3 indicadores, que definen el modo del nodo:

1. Un nodo está en *MODO PRESSURE* cuando la lista de nodos congestionados tiene elementos.
2. Un nodo MD está en *MODO PRESSURE DIRECTO* cuando la lista de nodos MD/MI congestionados tiene elementos.
3. Un nodo está en *MODO TRANSLÚCIDO* cuando la lista de nodos congestionados antes que nosotros tiene elementos.

### 6.2.5.3 Estados de los nodos

Se definen 3 estados de funcionamiento. Estos están asociados a cada circuito virtual, pudiendo un nodo estar en estados diferentes de funcionamiento según el circuito virtual.

**Estado CONGESTIONADO** El nodo tiene un enlace del circuito virtual congestionado.

**Actividad** Bloquear los avisos de congestión.

**Descongestión** Cuando el enlace congestionado se descongestiona, el nodo:

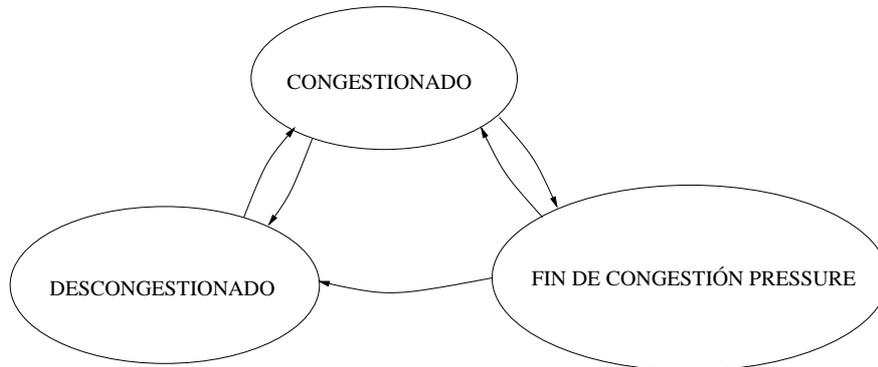


Figura 6.11: Diagrama de estados de control de flujo interno

1. Se borra de la *Lista de nodos congestionados*.
2. ¿ Está el nodo en modo PRESSURE ?
  - NO            Generar la onda back pressure de fin de congestión, y pasar al *estado descongestionado*.
  - SI            Pasar el *estado de fin de congestión pressure*.

**Estado FIN DE CONGESTIÓN PRESSURE** El nodo está a la espera de salir del *modo pressure* para pasar el estado descongestionado.

**Actividad** Bloquear los avisos de congestión.

**Congestión** Si uno de los enlaces utilizados por el circuito virtual se congestiona:

1. El nodo se añade a la *Lista de nodos congestionados*.
2. Pasar al *estado congestionado*.

**Final de pressure** Si el nodo sale del *modo pressure*:

1. Generar la onda back pressure de fin de congestión.
2. Pasar al *estado descongestionado*.

### Estado DESCONGESTIONADO

**Actividad** El nodo propaga toda la señalización de control de flujo.

**Congestión** Si uno de los enlaces utilizados por el circuito virtual se congestiona:

1. Copiar la *lista de nodos congestionados* a la *lista de nodos congestionados antes que nosotros*.
2. Generar el *identificador de congestión*.
3. El nodo se añade a la *Lista de nodos congestionados*.
4. Generar la onda back pressure de principio de congestión.
5. Pasar al *estado congestionado*.

### 6.2.6 Optimización: anulación de ondas de back pressure

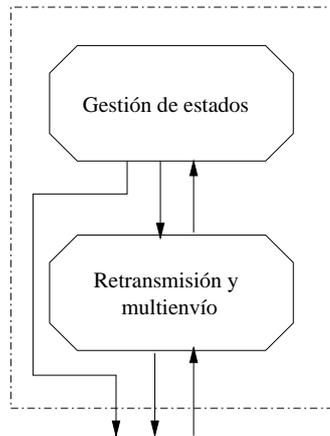
En el proceso de control de flujo por back pressure, existen dos “frentes de ondas”: el frente de onda de aviso de congestión, y el frente de onda de aviso de fin de congestión. Es posible que debido a la casualidad y a la corta duración de la congestión, el último frente alcance al primero. En tal caso, Jnet para ambos frentes, no los propaga.

Este caso se detecta con facilidad: el nodo recibe una señal de “Fin back pressure” con un identificador de congestión que no pertenece a la lista de nodos congestionados.

Cuando esto ocurre, el nodo almacena el identificador de la señal “Fin back pressure” en una lista, que llamamos *lista de back pressure anulados*, y no propaga hacia atrás esta señal. Posteriormente, llega la señal de “back pressure” (principio de congestión) correspondiente a esta misma congestión (para saberlo se comprueba si pertenece a la lista de back pressure anulados), que el nodo tampoco propaga hacia atrás.

### 6.2.7 Modelado del control de flujo interno mediante máquinas

En este apartado modelamos las características generales anteriormente descritas, al fin de poder dar una descripción más detallada del comportamiento de cada tipo de nodo.



La tarea se divide y estructura mediante dos máquinas, la *máquina de gestión de estados* y la *máquina de retransmisión y multienvío*. La máquina de retransmisión y multienvío recibe la señalización de los nodos vecinos, la procesa y se la comunica a la máquina de gestión de estado (por ejemplo eliminando señales duplicadas debido al multienvío de nodos vecinos). La máquina de gestión de estados cambia su estado en función de las informaciones que recibe de la máquina de retransmisión y multienvío, y reacciona enviándole señalización para los nodos vecinos, que esta última se encarga de entregar mediante retransmisiones y multienvío.

### Máquina de retransmisiones y multienvío

Sus funciones son:

- En recepción:
  - Filtrar las señales duplicadas, debido a la multitransmisión de los nodos vecinos.
  - Gestión de las señales de asentimiento.
    - \* Medición de tiempo de ida y vuelta (rtt).
    - \* Comprobar que una señal es asentida, y borrar los temporizadores correspondientes.
- En transmisión
  - Retransmisión.
  - Multitransmisión.
  - Generación de los asentimientos

La máquina de retransmisiones y multienvío *no tiene estados*. Esto le permite realizar el seguimiento de los temporizadores, del multienvío, de los asentimientos, independientemente del estado de funcionamiento del nodo.

### Máquina de gestión de estados

La máquina de gestión de estados:

- Controla la propagación de las ondas de back pressure.
- Actualiza las listas.
- Efectúa los cambios de estados del nodo.

En los diagramas SDL, salvo si se especifica lo contrario, las señales recibidas y transmitidas por la máquina de gestión vienen y van de/a la máquina de retransmisiones y multienvío.

### 6.2.8 Nodos N

En las figuras 6.12, 6.13, 6.14 están los diagramas sdl de la máquina de gestión de estados de los nodos tipo N. Se puede observar que no genera las señales de asentimiento (lo hace la máquina de retransmisión y multienvío).

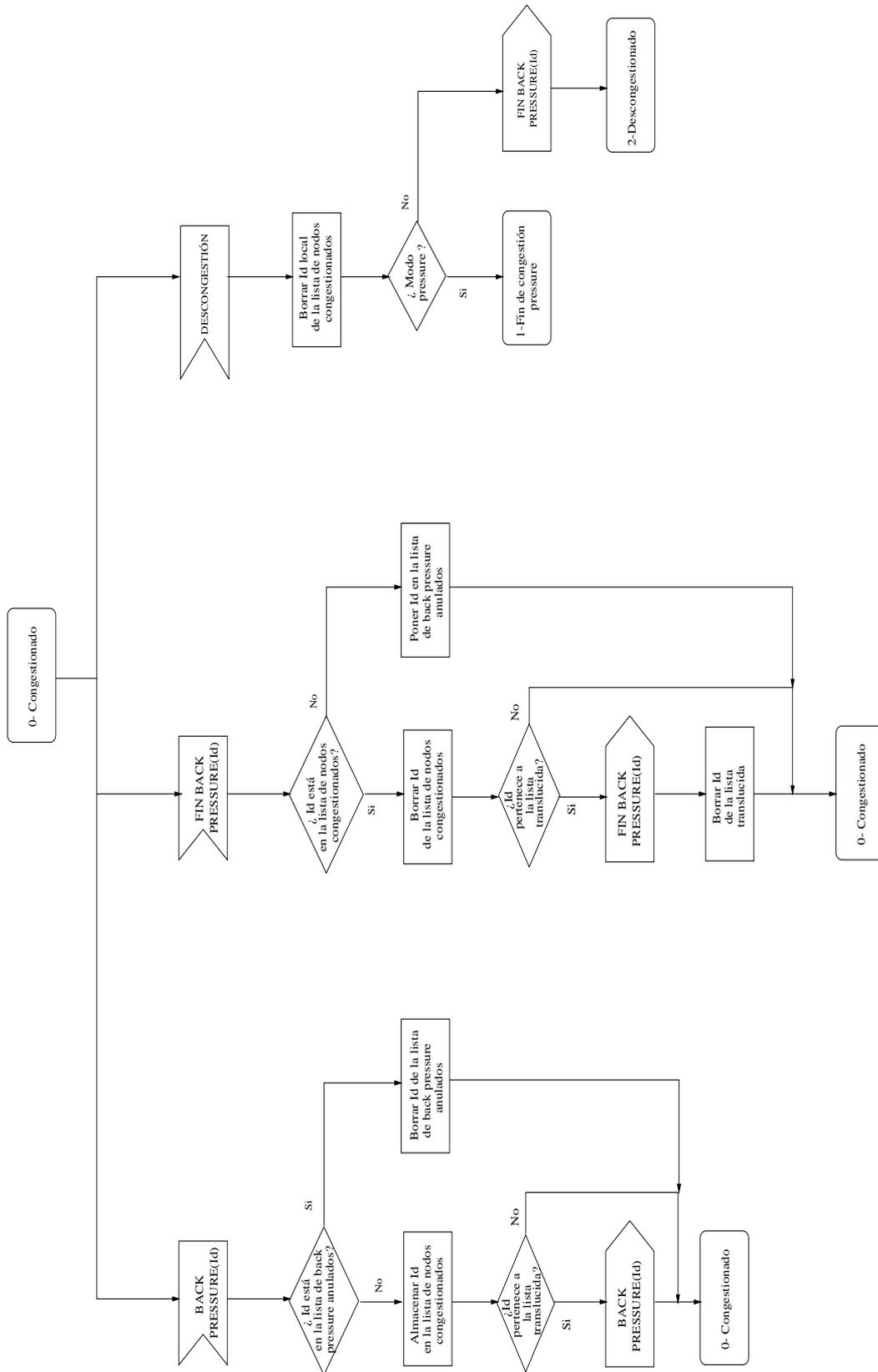


Figura 6.12: Máquina de gestión de estados, nodos tipo N, estado congestionado

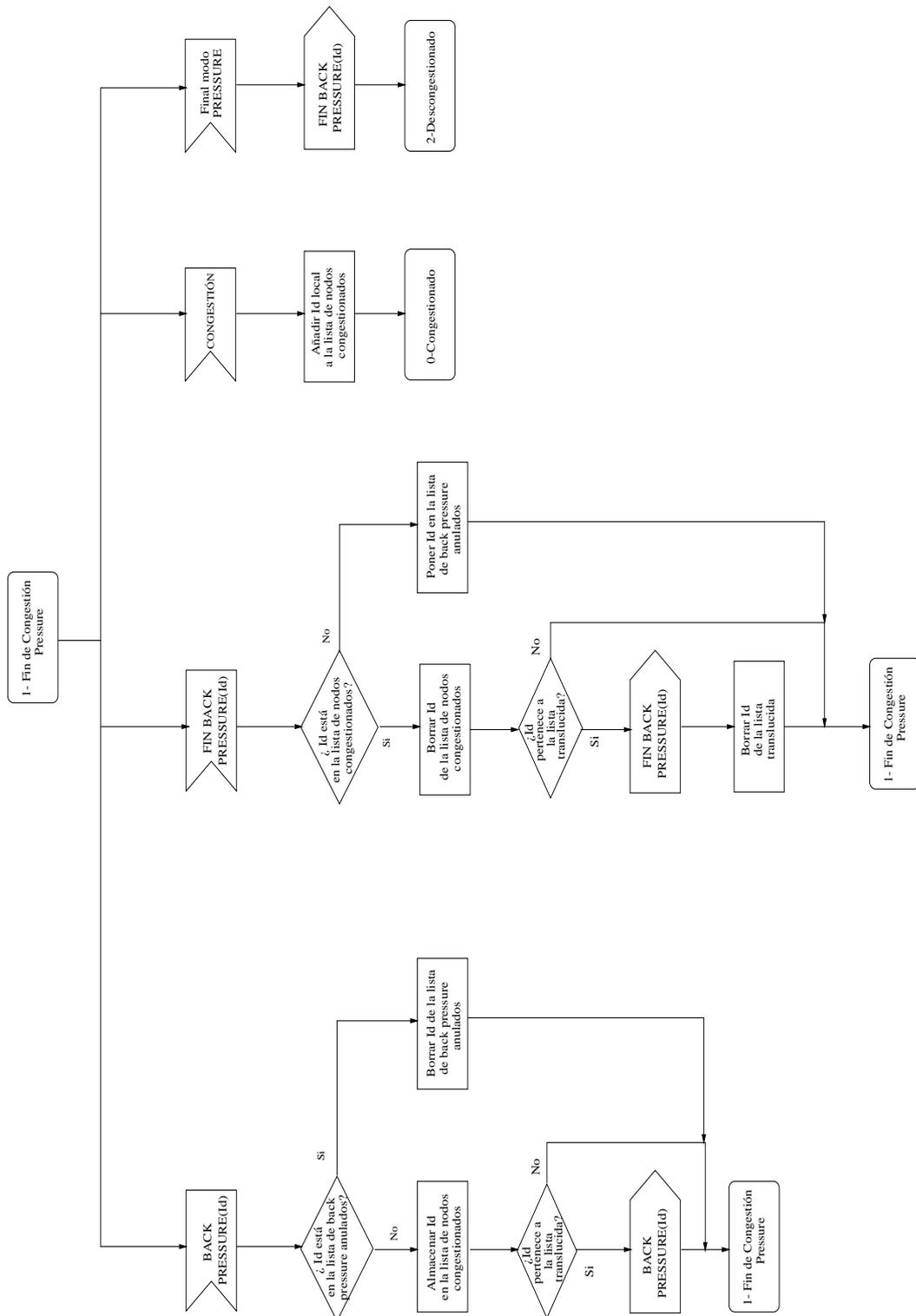


Figura 6.13: Máquina de gestión de estados, nodos tipo N, estado fin de congestión pressure.

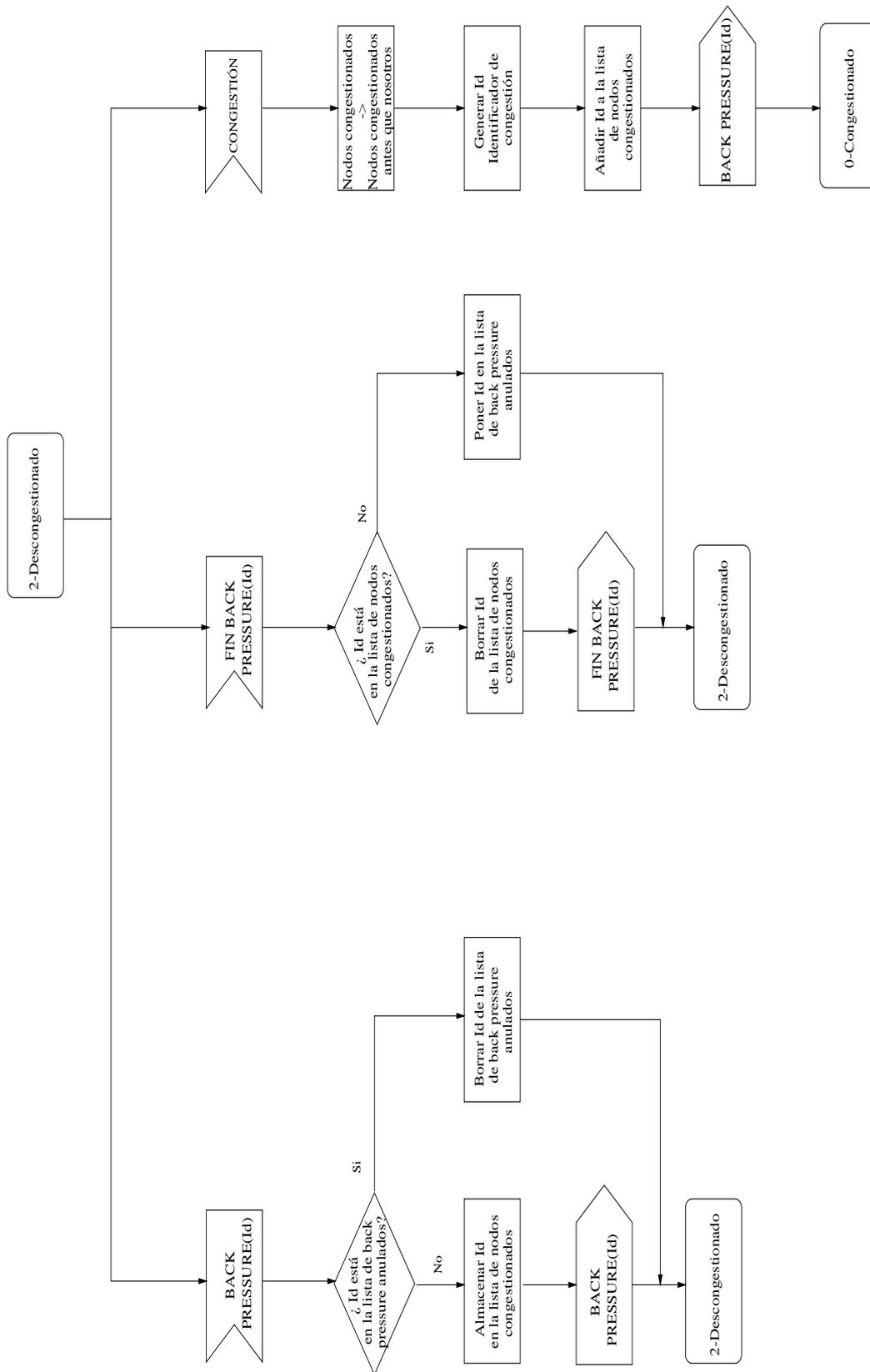


Figura 6.14: Máquina de gestión de estados, nodos tipo N, descongestionado.

### 6.2.9 Nodos X

En las figuras 6.15, 6.16, 6.17 están los diagramas sdl de las máquina de gestión de estados de los nodos tipo X. Observamos matices con respecto a la anulación de ondas de back pressure: los frentes anulados en el nodo X no son propagados hacia atrás, pero sí se envían las señales de asentimiento *ack back pressure indirecto* y *ack back pressure indirecto fin de congestión*, porque estas señales no son suprimibles (las espera el *nodo generador*).

### 6.2.10 Nodos Y

En las figuras 6.18, 6.19, 6.20 están los diagramas sdl de la máquina de gestión de estados de los nodos tipo Y. Aunque en el diagrama no se compruebe, lógicamente la emisión de señales de back pressure directo sólo se realiza si existe enlace directo hacia el nodo anterior.

De especial interés es el tratamiento de la señal *forward back pressure indirecto* y *forward back pressure indirecto fin de congestión*: son propagadas, pero no incluidas en las listas de nodos congestionados porque vienen de nodos *anteriores* y por lo tanto no influyen en el estado de back pressure de este nodo.

No aparecen las señales *ack back pressure indirecto* y *ack back pressure indirecto fin de congestión* porque son procesadas por la *máquina de retransmisión y multienvío* (nota: cuando estas señales llegan hasta un nodo Y es porque Y es el *nodo generador de señal de congestión*)

### 6.2.11 Nodos MD

En las figuras 6.21, 6.22, 6.23, 6.24 están los diagramas sdl de la máquina de gestión de estados de los nodos tipo MD.

Los nodos del bucle MI (si existen) son siempre posteriores a los nodos MD, y su estado de congestión es recibido mediante el procedimiento de back pressure indirecto mediante las señales *ack back pressure indirecto* y *ack back pressure indirecto fin de congestión*. Al ser nodos posteriores, forman parte de la *lista de nodos congestionados*, y por lo tanto contribuyen al *estado pressure* del nodo. Por este motivo, que un nodo MD esté en estado pressure no es motivo suficiente para que bloquee las señales de back pressure directo de los nodos MD posteriores: puede darse el caso por ejemplo que ninguno de los nodos MD posteriores esté congestionado, y que lo esté algún nodo MI; en tal caso el nodo MD estaría esperando a que se descongestionaran estos nodos MI para salir del modo back pressure fin de congestión, lo que no aporta nada, sino un ralentizamiento del proceso.

Por este motivo se ha diseñado el *modo pressure directo* para los nodos MD, y la lista de nodos MD congestionados. El final del modo pressure directo define en instante de salida del *estado fin de congestión pressure*.

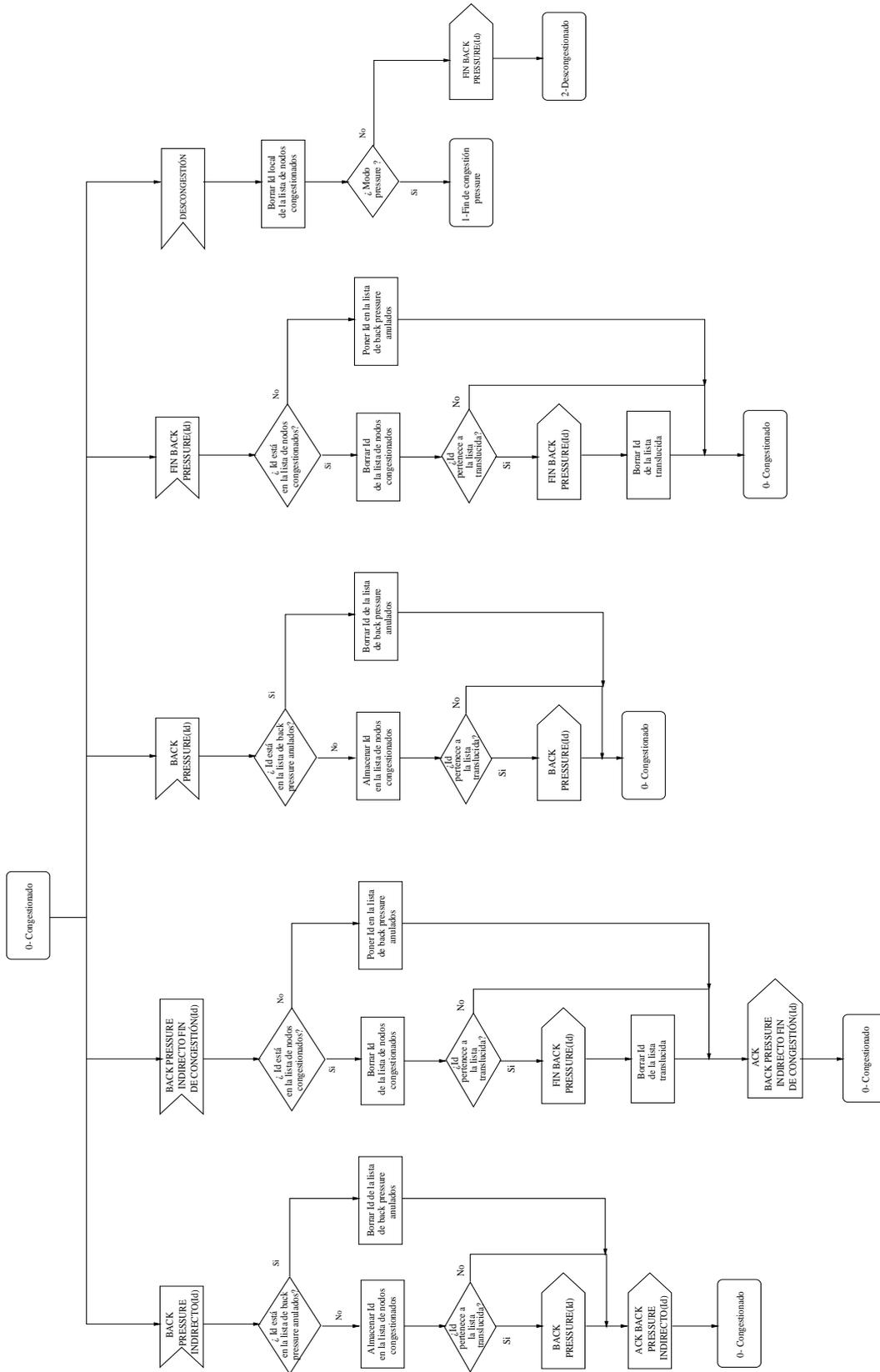


Figura 6.15: Máquina de gestión de estados, nodos tipo X, congestionado.

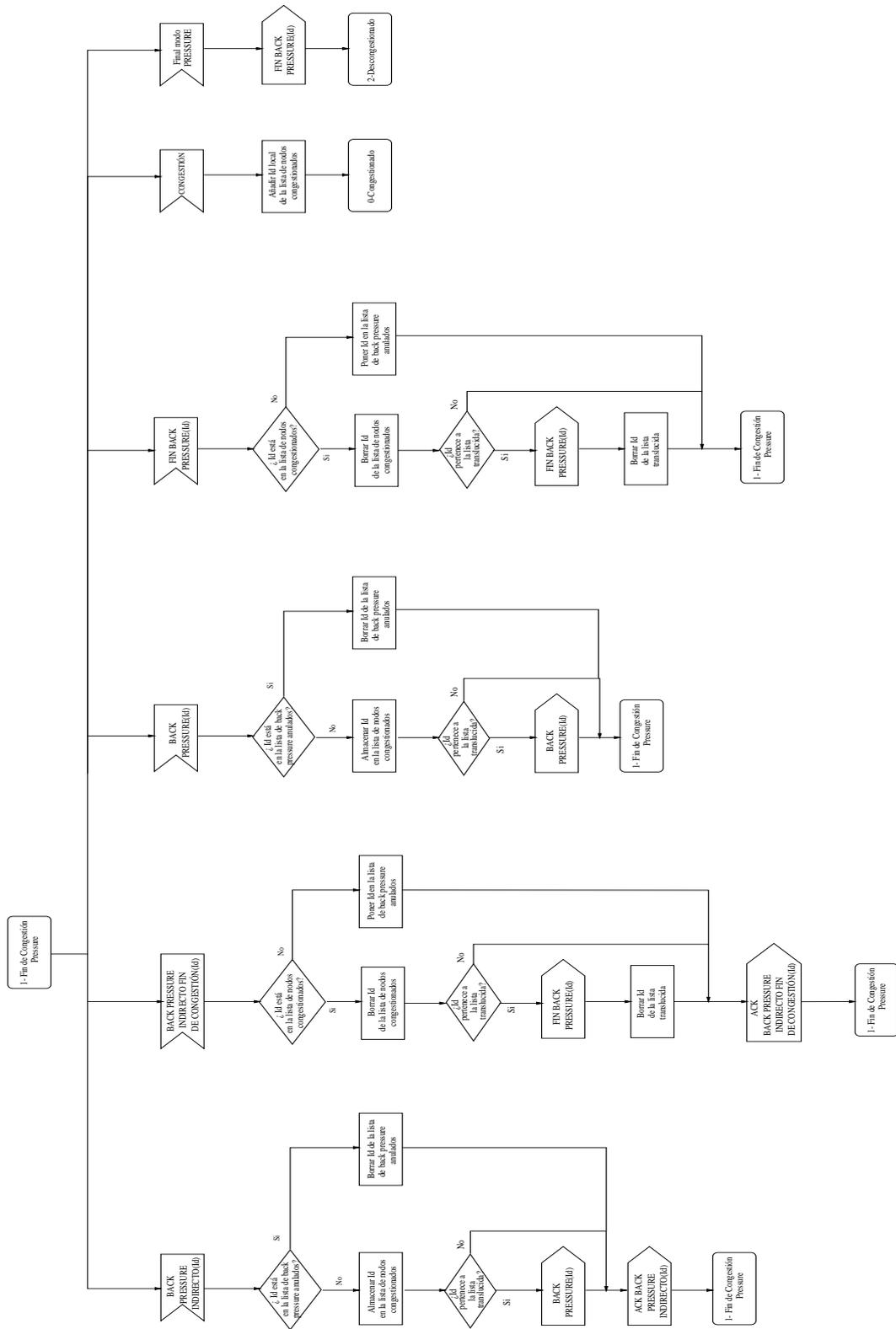


Figura 6.16: Máquina de gestión de estados, nodos tipo X, estado fin de congestión pressure.

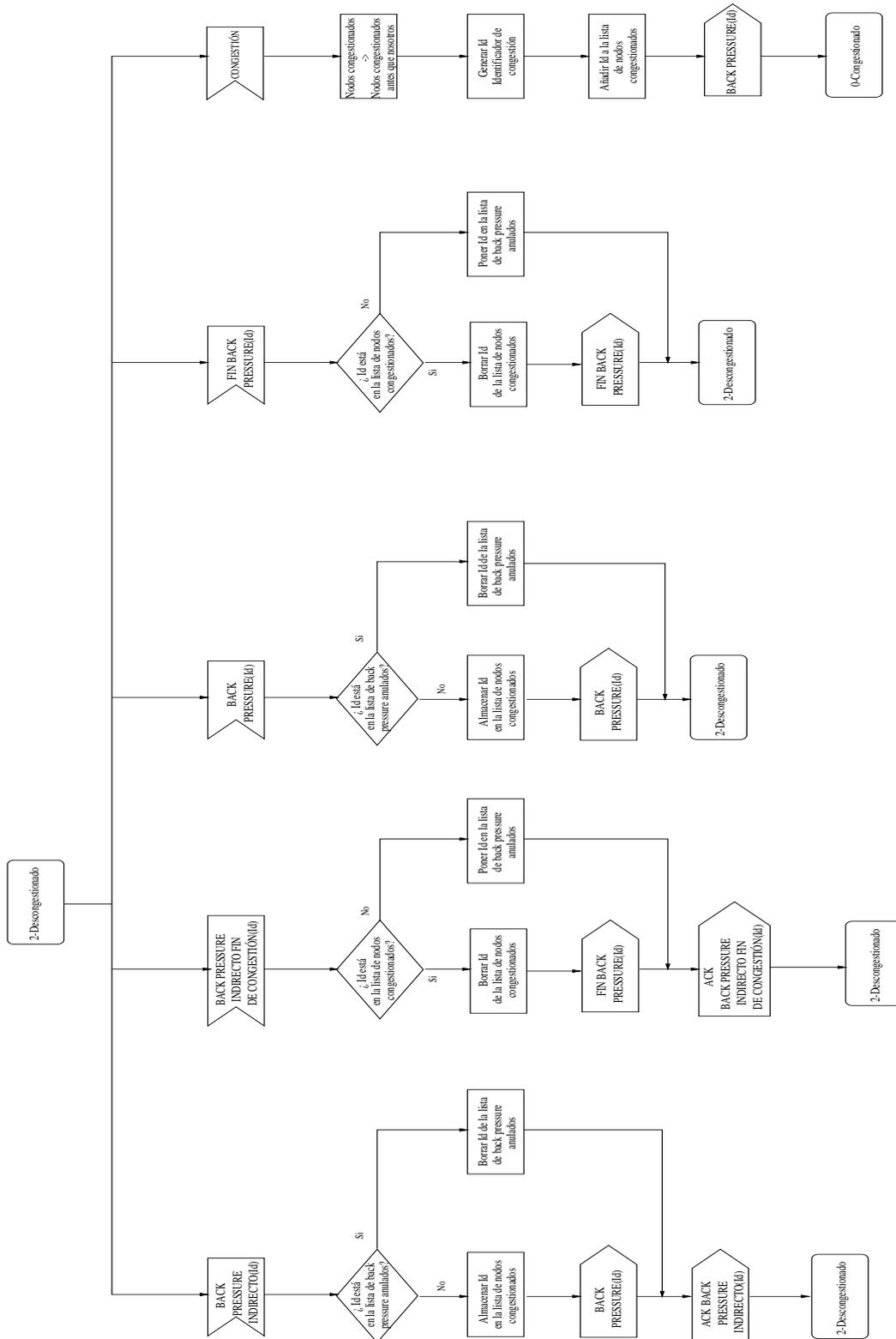


Figura 6.17: Máquina de gestión de estados, nodos tipo X, estado descongestionado.

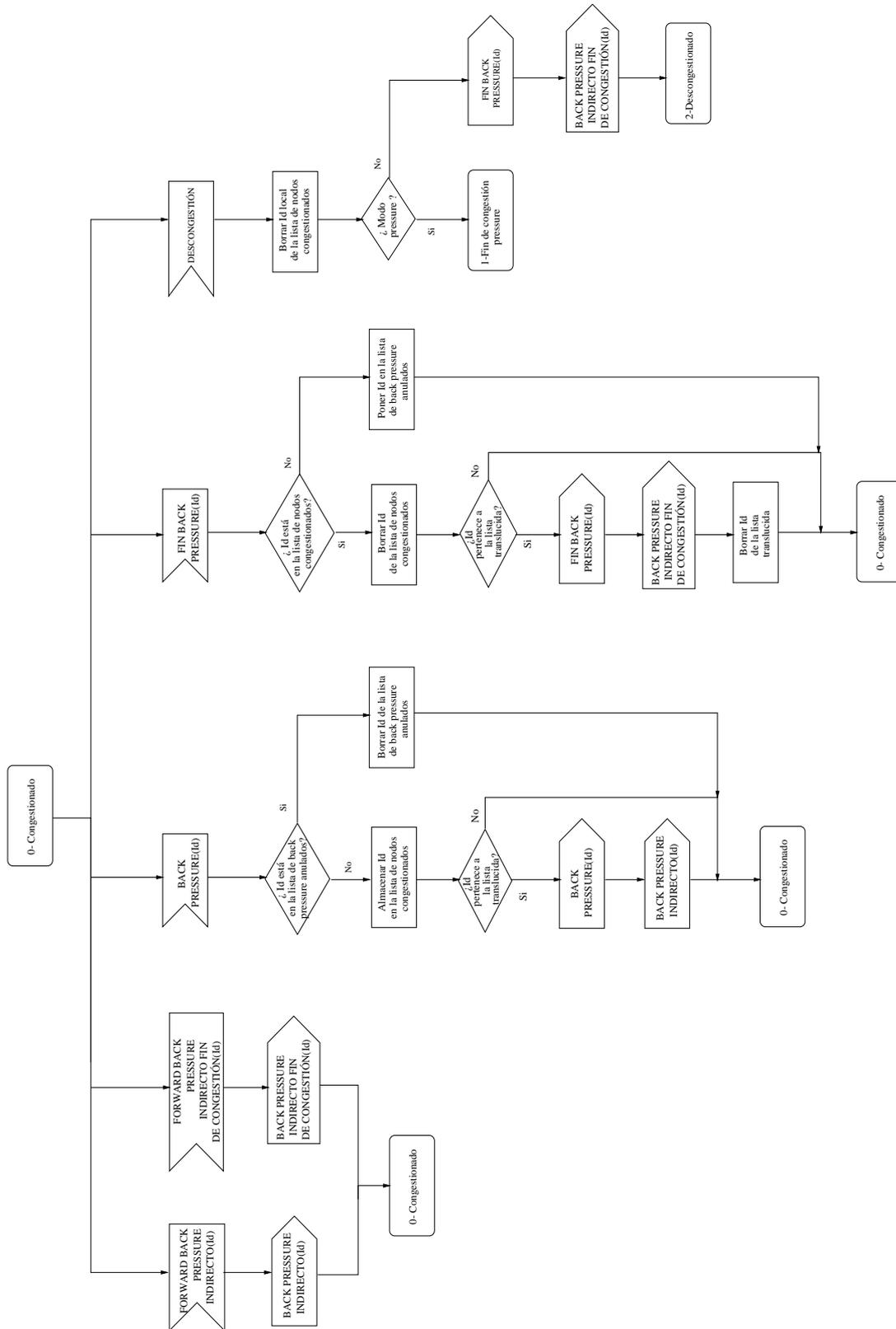


Figura 6.18: Máquina de gestión de estados, nodos tipo Y, estado congestionado.

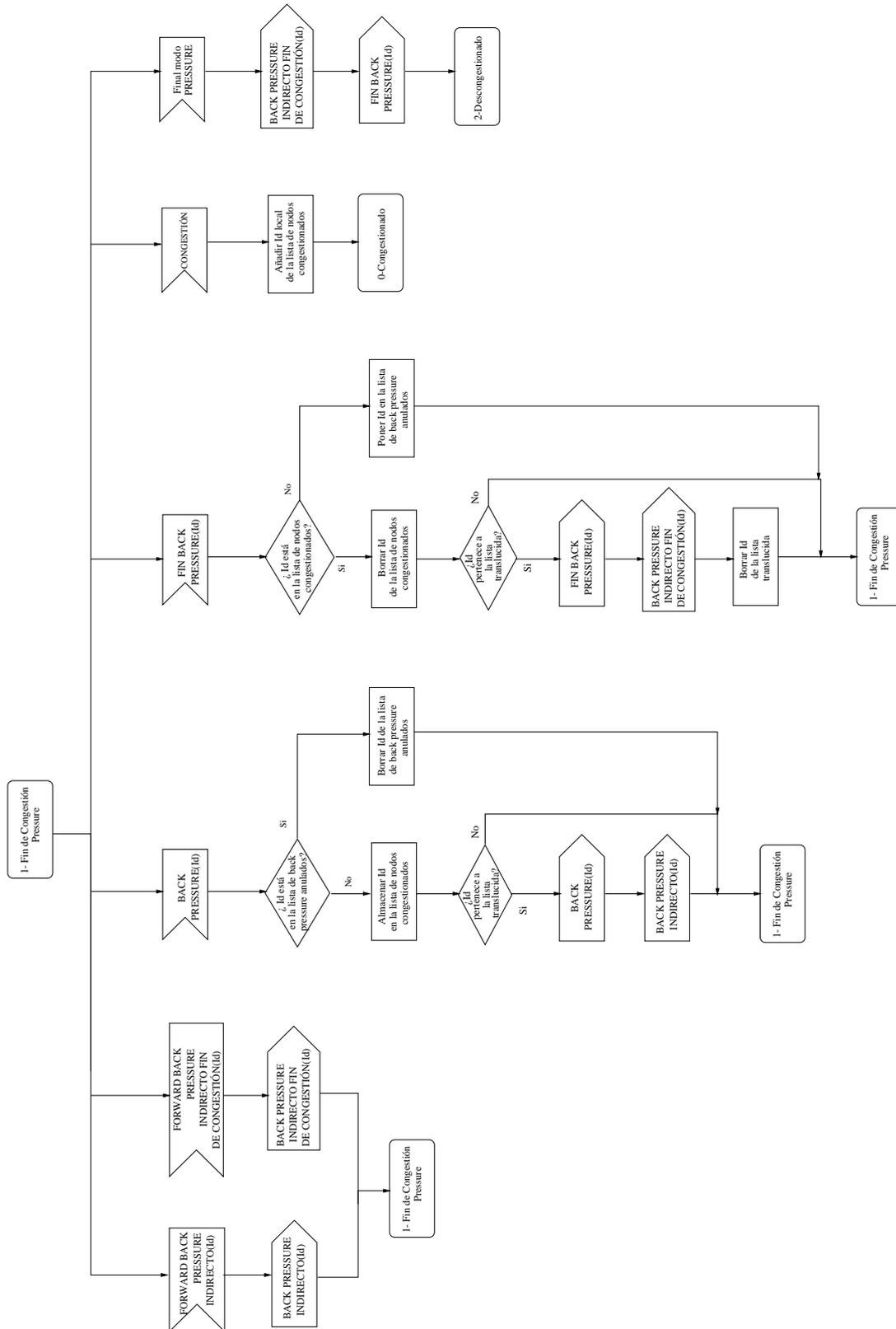


Figura 6.19: Máquina de gestión de estados, nodos tipo Y, estado fin de congestión pressure.

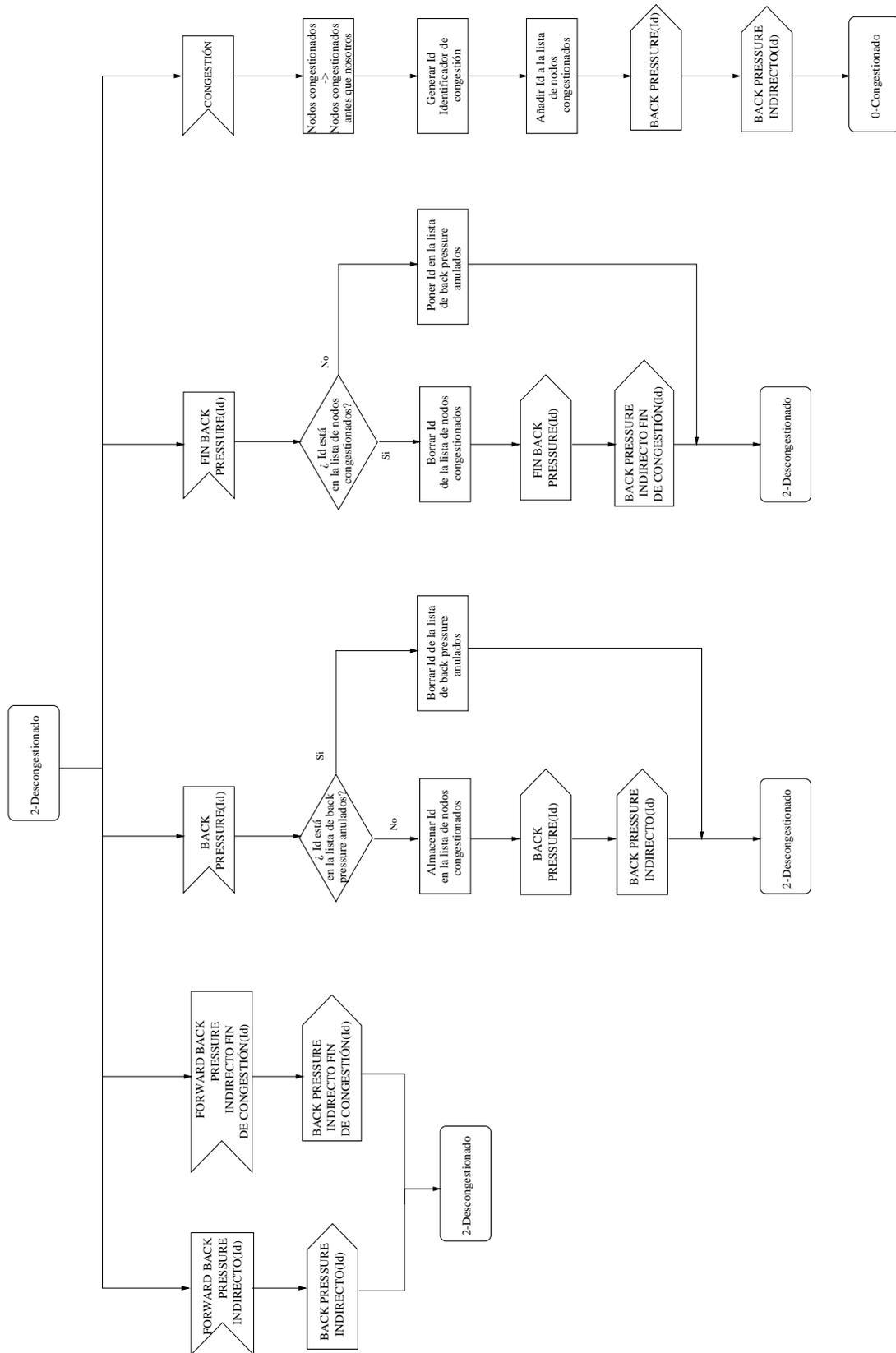
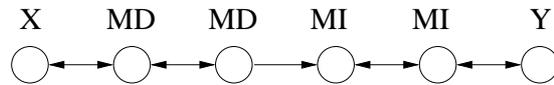


Figura 6.20: Máquina de gestión de estados, nodos tipo Y, estado descongestionado.



Las señales *ack back pressure indirecto* y *ack back pressure indirecto fin de congestión* recibidas por esta máquina son siempre referentes a congestiones de otros nodos.

Los nodos MD ni reciben ni generan señales *forward back pressure indirecto* / *forward back pressure indirecto fin de congestión*.

### 6.2.12 Nodos MI

El funcionamiento de los nodos MI es muy parecido al de los nodos MD, salvo que para anunciar su congestión utilizan el mecanismo de back pressure indirecto y, si tienen enlace hacia atrás, el mecanismo de back pressure directo.

Además, es aplicable el método de ocultamiento utilizado en los nodos MD con el *modo pressure directo*, con la diferencia que en este modo no se garantiza que todos los nodos MI posteriores congestionados estén incluidos en la lista de nodos MI congestionados, sólo lo estarán los que tienen la posibilidad de enviarnos avisos de back pressure directo. Es posible que en el conjunto de nodos MI haya varios grupos aislados con posibilidad de back pressure directo. No obstante, sigue siendo útil el procedimiento, puesto que disminuye el tráfico de señalización de congestión.

También cabe destacar la diferencia entre las señales *ack back pressure indirecto* / *ack back pressure indirecto fin de congestión* y *forward back pressure indirecto* / *forward back pressure indirecto fin de congestión*: la primera indica la congestión / descongestión de un nodo posterior (por lo tanto entra en la lista de nodos congestionados), mientras que la segunda indica la congestión de un nodo anterior (no entra en la lista de nodos congestionados).

Debido a la utilización de los dos métodos de propagación de ondas de back pressure (directo e indirecto), un nodo MI puede recibir por dos caminos diferente la misma señal. Esto se tiene en cuenta en la anulación de frentes de ondas: la señal *ack back pressure indirecto fin de congestión* no anula las señales *back pressure directo* pendientes de llegar.

Los diagramas sdl están en las figuras 6.25, 6.26, 6.27, 6.28.

### 6.2.13 Nodos O

Los nodos de tipo O se limitan a retransmitir las señales *back pressure indirecto* y *back pressure indirecto fin de congestión*.

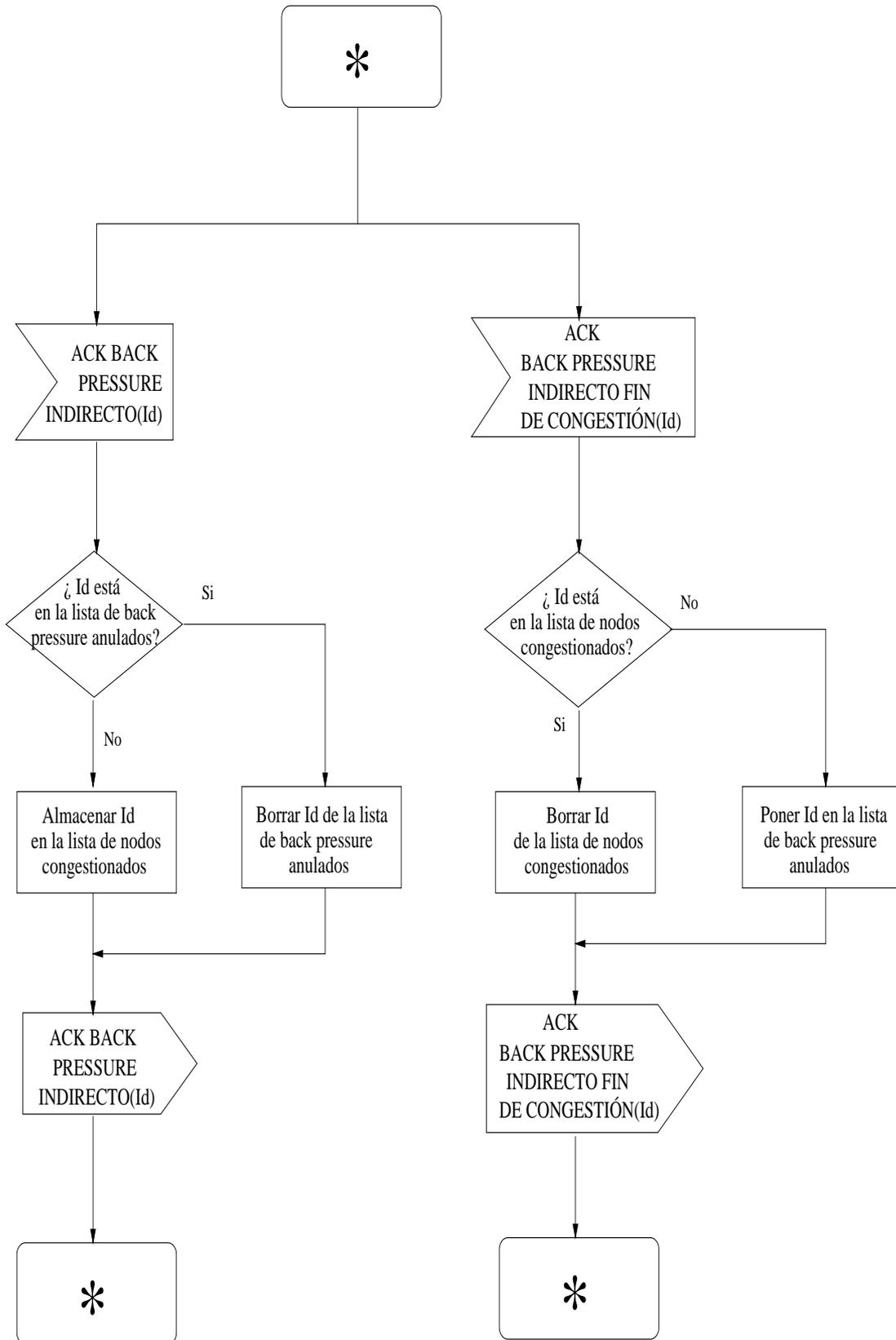


Figura 6.21: Máquina de gestión de estados, nodos tipo MD, transiciones en todos los estados.

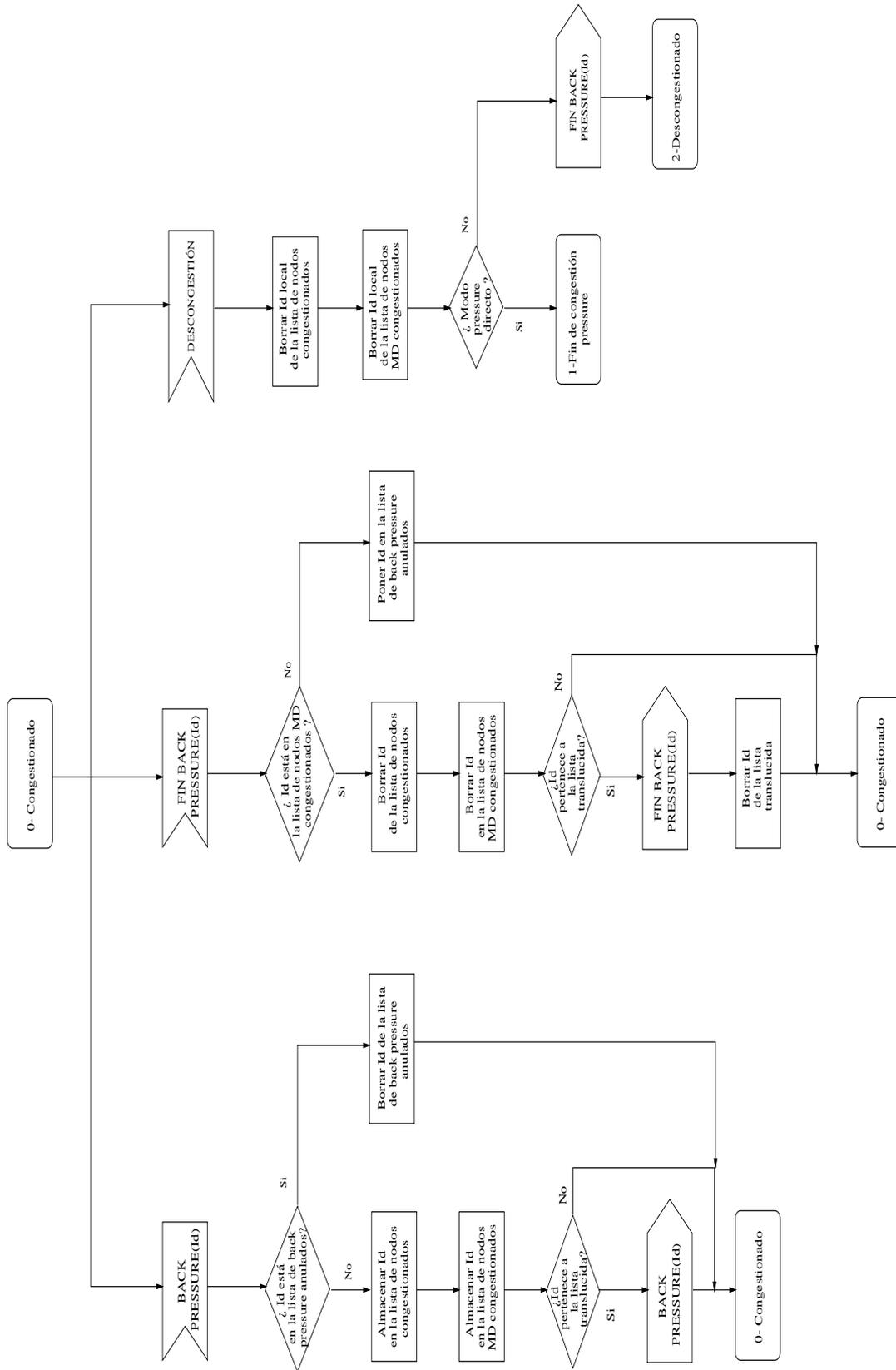


Figura 6.22: Máquina de gestión de estados, nodos tipo MD, estado congestionado.

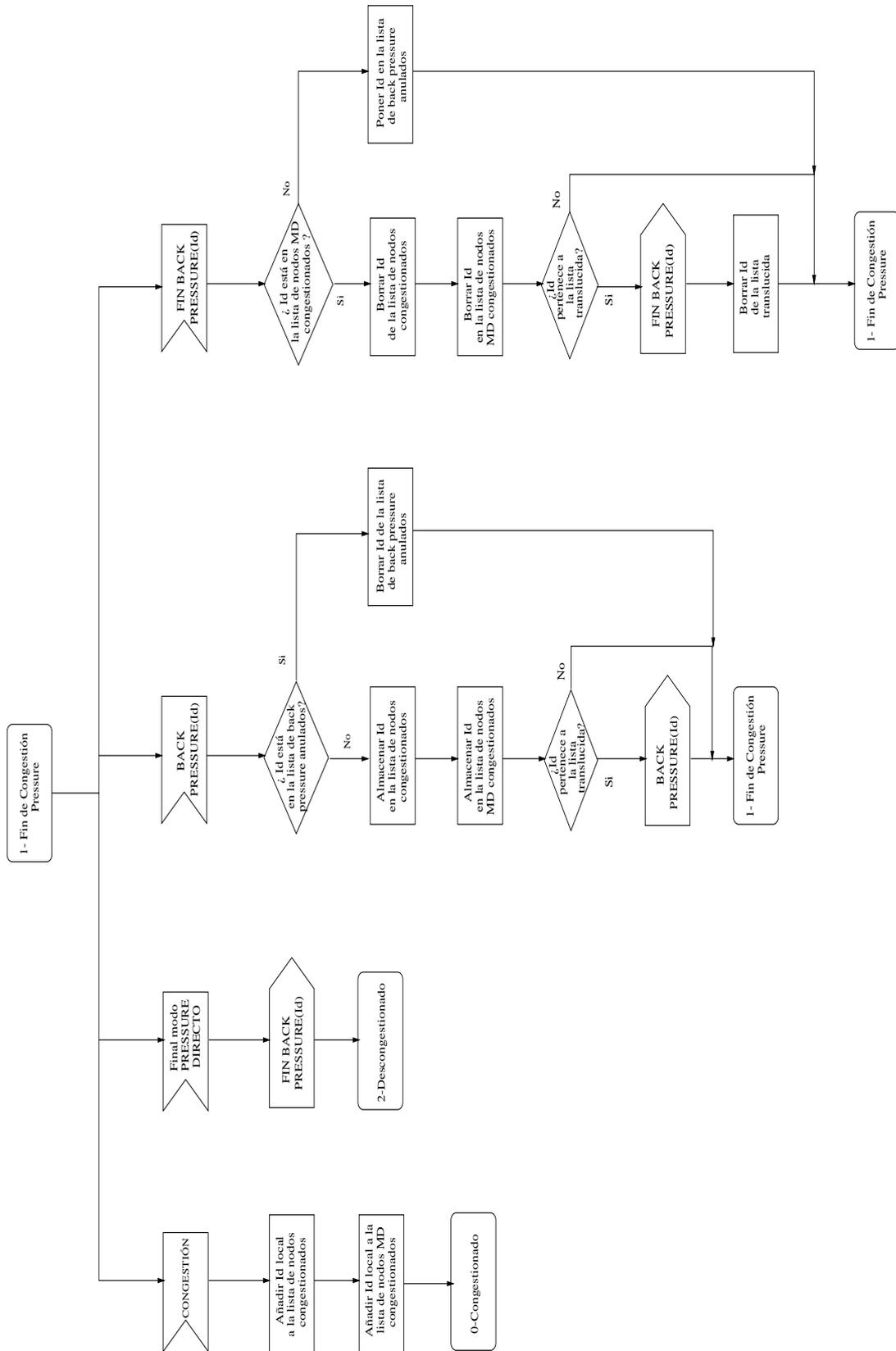


Figura 6.23: Máquina de gestión de estados, nodos tipo MD, estado fin de congestión pressure.

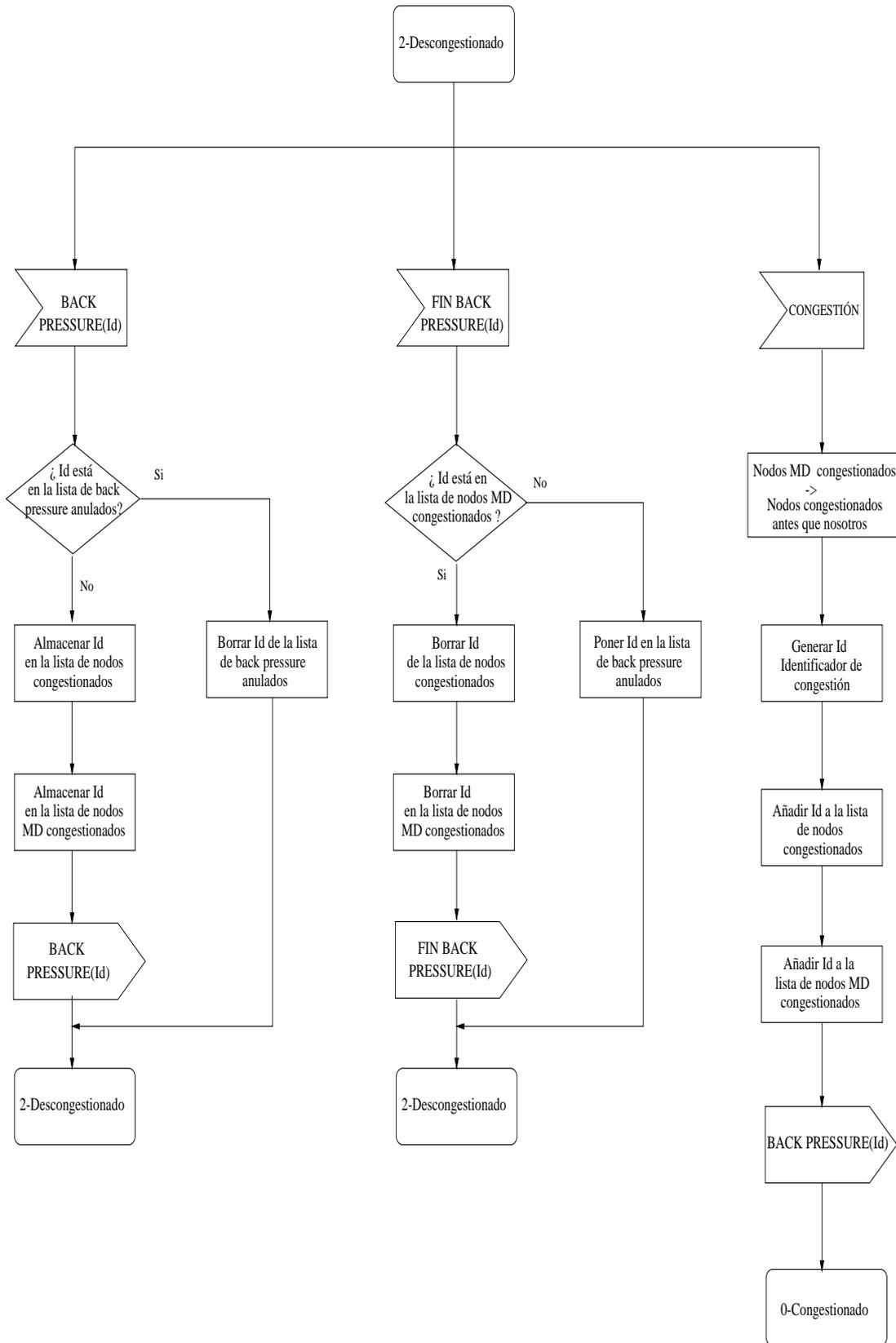


Figura 6.24: Máquina de gestión de estados, nodos tipo MD, estado descongestionado.

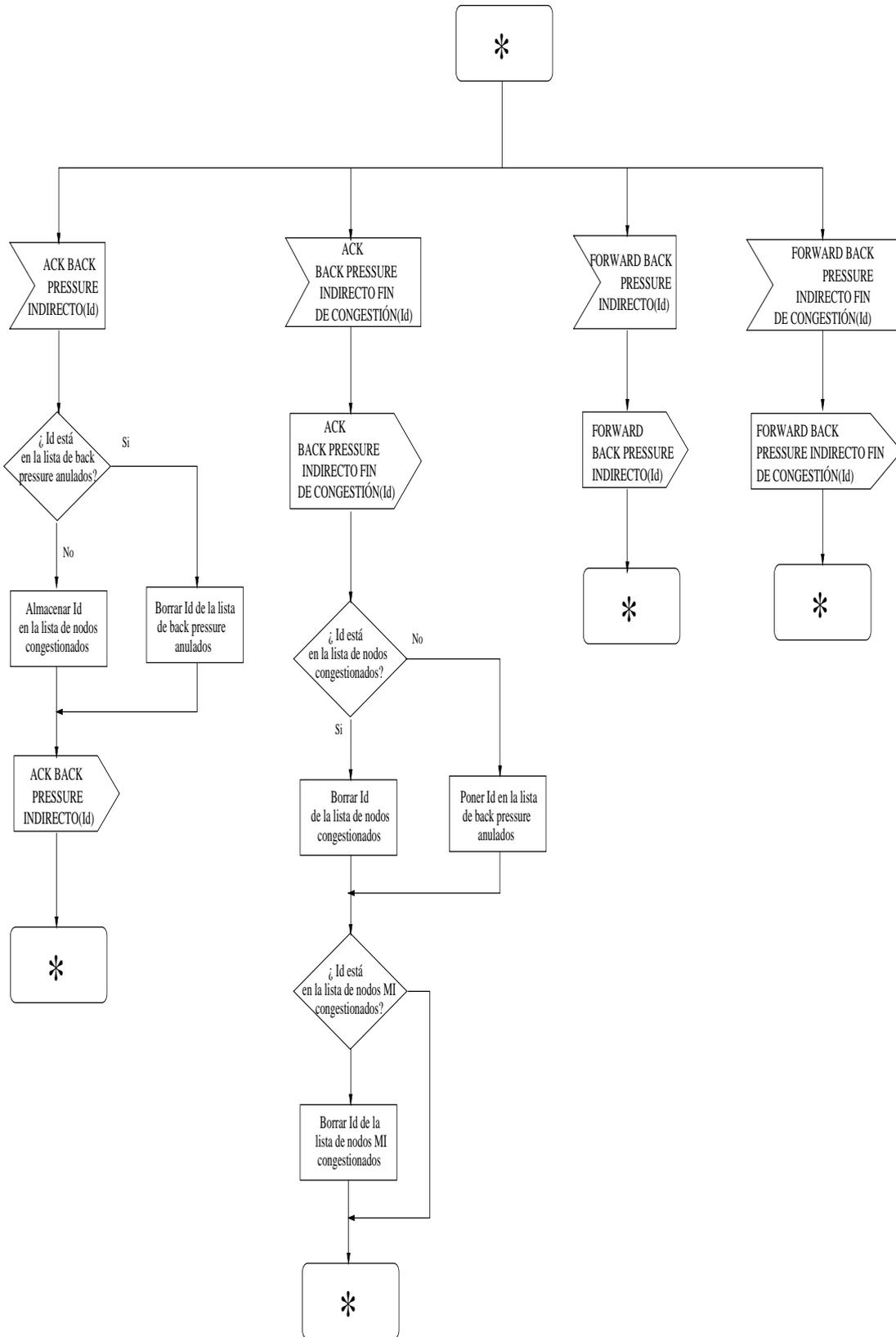


Figura 6.25: Máquina de gestión de estados, nodos tipo MI, transiciones en todos los estados.

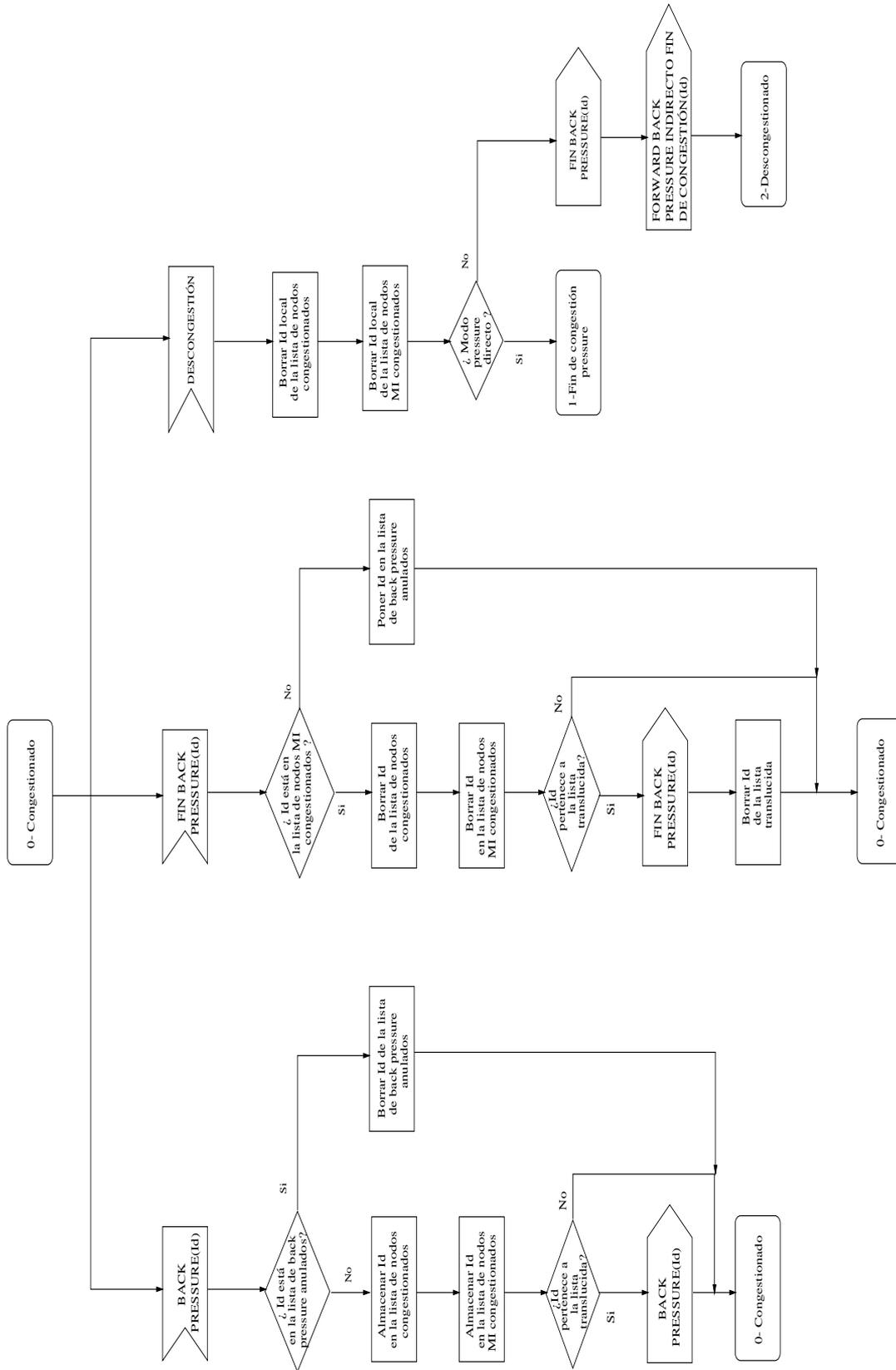


Figura 6.26: Máquina de gestión de estados, nodos tipo MI, estado congestionado.

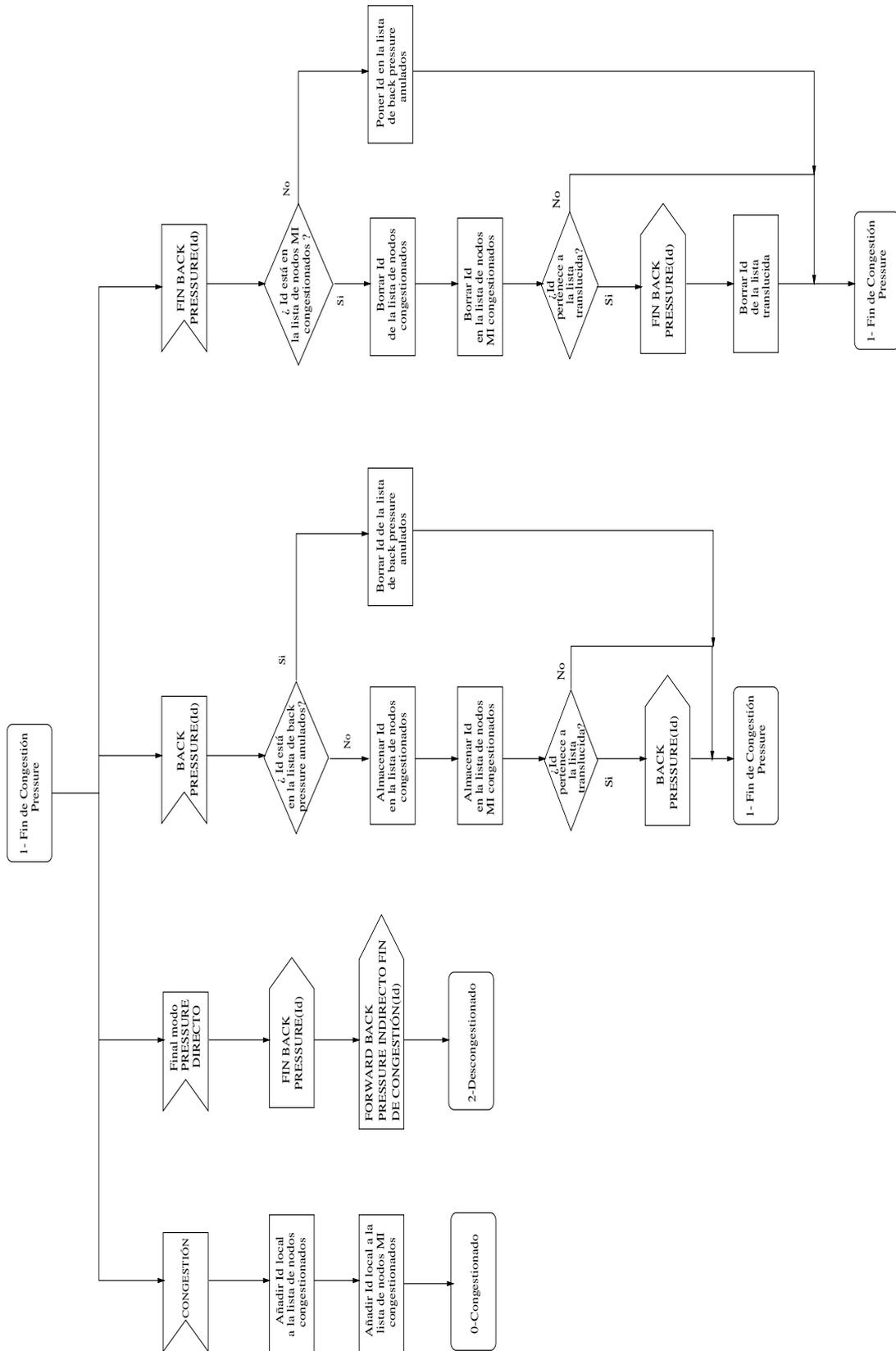


Figura 6.27: Máquina de gestión de estados, nodos tipo MI, estado fin de congestión pressure.

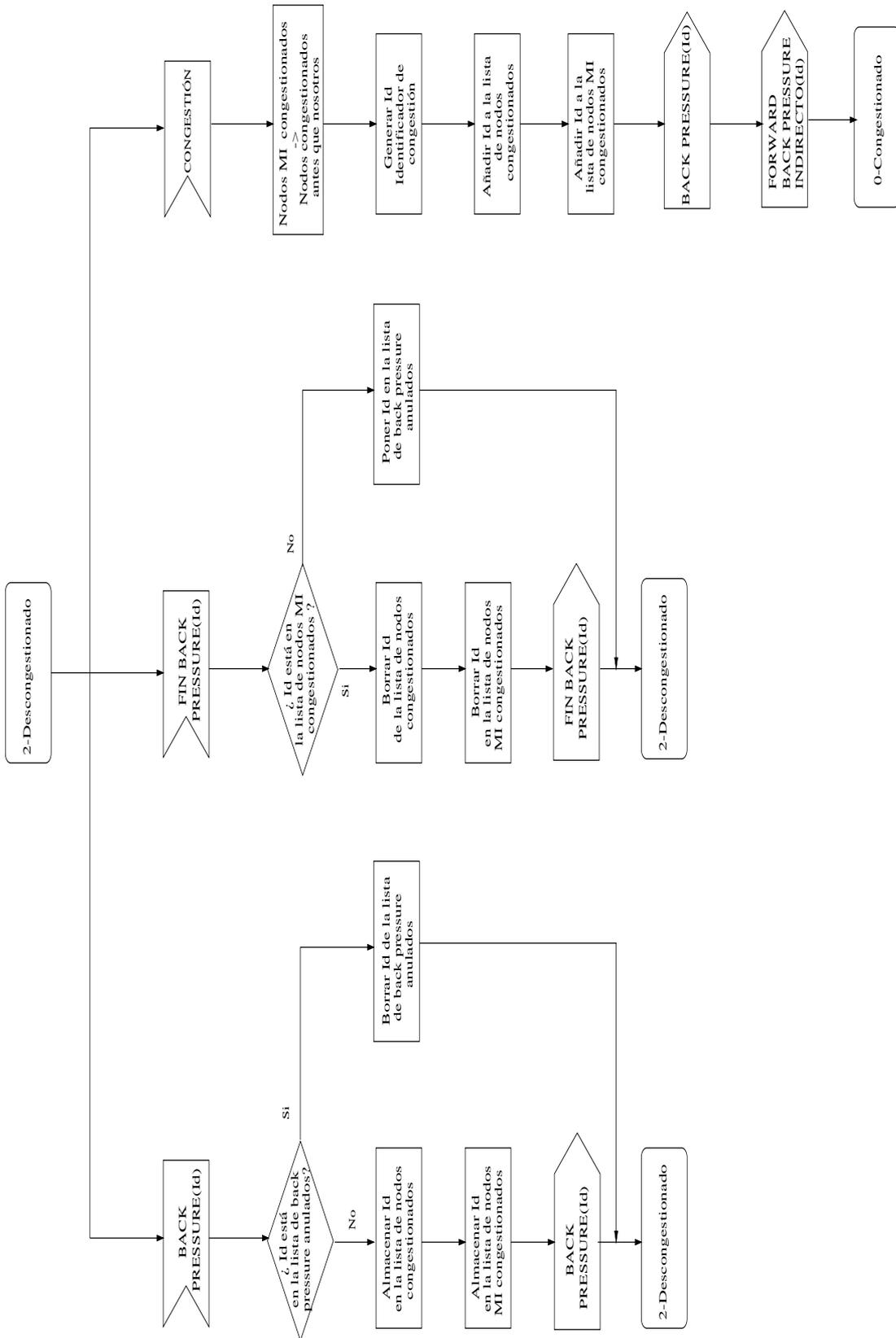


Figura 6.28: Máquina de gestión de estados, nodos tipo MI, estado descongestionado.

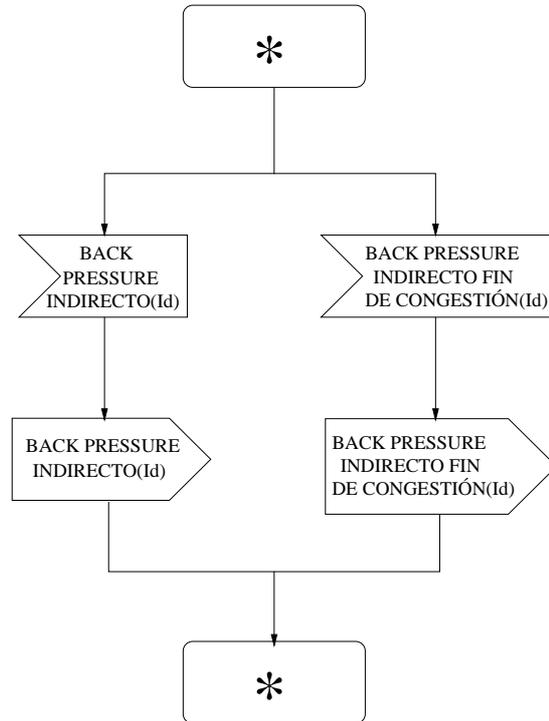


Figura 6.29: Máquina de gestión de estados, nodo tipo O.

## 6.2.14 Criterios de congestión de enlaces

### 6.2.14.1 Enlaces con conexión hdlc

La máquina de medición enlace a nivel de red conoce la longitud de la cola de pds pendientes de transmitir, para ello le basta contar el número de bits  $N_{colatx}$  total de las pds pendientes de asentimiento de la *tabla almacén de primitivas DL-DATA Request transmitidas* (ver apartado 5.6.1).

Por cada interfaz, definimos dos valores umbral  $N_{colatx_1}$ <sup>4</sup> y  $N_{colatx_2}$ <sup>5</sup>, con  $N_{colatx_1} < N_{colatx_2}$ . Si el enlace no está congestionado, se considera que se congestiona cuando  $N_{colatx} > N_{colatx_2}$ . Si el enlace está congestionado, se considera que se descongestiona cuando  $N_{colatx} < N_{colatx_1}$ .

### 6.2.14.2 Enlaces sin conexión hdlc

Un indicador muy realista del estado de congestión de un enlace es el tamaño de su cola de transmisión. No obstante, como ya hemos destacado a lo largo de este documento, no es posible su medición en enlaces sin conexión hdlc.

<sup>4</sup>“Umbral de descongestión de enlaces con conexión hdlc”. Se trata de un parámetro local del nodo asociado a cada canal. Ver apéndice B.

<sup>5</sup>“Umbral de congestión de enlaces con conexión hdlc”. Se trata de un parámetro local del nodo asociado a cada canal. Ver apéndice B.

La ocupación del canal  $OC_i$  es un indicador que da indicios válidos de congestión. Para enlaces con conexión con hdlc este indicador no es muy válido, puesto que el propio sistema de ventanas con parada y espera ejerce una función de control de flujo que limita la ocupación del canal, pero para enlaces sin conexión hdlc un canal ocupado por portadoras de datos en alto porcentaje del tiempo, es síntoma de congestión o de posible congestión (si se desea que el nivel de red sea prudente y previsor, en ambos casos es conveniente activar medidas de control de flujo).

Por cada interfaz, definimos dos valores umbral  $OC_1^6$  y  $OC_2^7$ , con  $OC_1 < OC_2$ . Si el enlace no está congestionado, se considera que se congestiona cuando  $OC_i > OC_2$ . Si el enlace está congestionado, se considera que se descongestiona cuando  $OC_i < OC_2$ .

Nota: la medición de  $OC_i$  debe hacerse en el nodo origen, salvo en enlaces unidireccionales con equipos que sólo pueden transmitir y no recibir.

### 6.2.15 Procedimiento de información de congestión

Para cada enlace, la máquina de medición de enlaces a nivel de red realiza un seguimiento de  $Ncolatx$  (localmente) y  $OC_i$  (realizando peticiones a la máquina de mediciones a nivel de enlace con la primitiva MEA-OCCUPATION Request). Cuando detecta la congestión o descongestión en un canal determinado, informa a *todas* las máquinas de circuito virtual cuyo circuito virtual utiliza este enlace, mediante las primitivas VC-CONGEST (principio de congestión) y VC-ENDCONGEST (descongestión, fin de congestión).

Para ello, cada máquina de medición de enlaces a nivel de red debe conocer la lista de circuitos virtuales que están utilizando su enlace, es decir que es necesaria una variable (tabla o lista) con esta información, tal y como también se dijo en el apartado 4.6.

---

<sup>6</sup>“Umbral de descongestión de enlaces sin conexión hdlc”. Se trata de un parámetro local del nodo asociado a cada canal. Ver apéndice B.

<sup>7</sup>“Umbral de congestión de enlaces sin conexión hdlc”. Se trata de un parámetro local del nodo asociado a cada canal. Ver apéndice B.

# Capítulo 7

## Encaminamiento

### 7.1 Procedimiento general

El encaminamiento consiste en la determinación del camino que va a seguir el circuito virtual. El nodo origen del circuito virtual a partir del grafo de la red decide el camino y lo establece. Una vez establecido, el camino del circuito virtual permanece invariante, hasta el momento en el que se cierra (se siguen los procedimientos clásicos seguidos en las redes de circuitos virtuales, ver [11]).

En el caso de circuitos virtuales bidireccionales, el encaminamiento se hace en dos fases:

1. El nodo origen decide el camino del circuito virtual de ida, y lo establece.
2. El nodo destino al detectar el establecimiento del circuito virtual de ida, decide el camino del circuito virtual de vuelta y lo establece<sup>1</sup>.

### 7.2 Métricas

Los métodos de obtención del camino óptimo se basan en el grafo de la red, y en métricas que califican el coste de los enlaces. Obtener el mejor camino consiste en hallar el camino menos costoso, siendo el coste del camino la suma de los costes de los enlaces utilizados por el camino.

El nodo que decide el encaminamiento del circuito virtual deberá asignar las métricas a los enlaces a partir de sus parámetros característicos, e implementar un algoritmo de encaminamiento que le permita obtener el mejor camino. En la literatura de redes de ordenadores existe una variedad de algoritmos de cálculo, que no vamos a describir en este proyecto.

---

<sup>1</sup>Además, para el control de flujo interno hay una tercera fase, pero no está relacionada con el encaminamiento, ver apartado 6.2.3

Sólo queremos destacar que el algoritmo no tiene porque ser el más rápido ni el más complejo, puesto que el proceso de encaminamiento se realiza poco frecuentemente (en el establecimiento solamente). Como veremos en el apartado siguiente, Jnet no siempre elige el camino de menor coste, y es por lo tanto probable que se deba modificar o aplicar de forma diferente los algoritmos de encaminamiento, por este motivo, se recomienda utilizar un algoritmo sencillo y claro, aunque consuma más recursos de memoria y de tiempo de procesamiento.

Puesto que los algoritmos de encaminamiento dan el mismo resultado (el camino de menor coste), es el criterio de asignación de los costes de los enlaces que determina las propiedades de la ruta elegida. Una ventaja interesante del encaminamiento en el origen es que se puede implementar cualquier criterio de asignación de métricas localmente, sin necesidad de realizar ningún otro cambio; la asignación de métricas que presentamos a continuación puede ser mejorada o adaptada a características específicas que quisiéramos dar al encaminamiento.

Jnet elige preferentemente los enlaces con mayor capacidad disponible, es decir los de mayor  $C_j - F_j$ . La asignación del coste  $K_j$  de cada enlace seguirá la regla:

$$K_j = \frac{1}{C_j - F_j}$$

El coste total que se quiere minimizar es:

$$K = \sum \frac{1}{C_j - F_j}$$

Es una asignación con propiedades interesantes, porque minimizar  $K$  significa maximizar el paralelo de las capacidad disponible, puesto que:

$$\frac{1}{K} = (C_1 - F_1) // (C_2 - F_2) // \dots // (C_n - F_n)$$

Esta magnitud representa con cierta veracidad la capacidad ofrecida al nodo origen, que tratamos de maximizar hallando el mejor camino. [Nota: Dada la gran amplitud del rango de valores de las capacidades (del orden de  $10^9$ ),  $K_j$  seguramente deba ser codificado en coma flotante. Ello en principio no es un problema, puesto que los algoritmos de encaminamiento no requieren operaciones complejas, generalmente sólo sumas y restas; en caso de que lo fuera, siempre se puede multiplicar  $K_j$  por una constante, y tomar su parte entera).

### 7.3 Probabilidad de error

Puesto que se conoce la probabilidad  $P_j$  de que un paquete no se pierda en un enlace, se conoce la probabilidad  $P$  de que un paquete no se pierda en el circuito virtual:  $P = \prod P_j$ . Para enlaces con conexión hdlc, se considera que  $P_j = 1$ .

$P$  expresa la calidad del circuito virtual. Jnet permite que el usuario demande una probabilidad  $P_{min}$  mínima, tanto para el circuito virtual de ida como el de vuelta.

Jnet debe elegir la mejor ruta (menor coste) que se conforme a la probabilidad exigida  $P > P_{min}$ . Esto significa que debe ser capaz de descubrir las mejores rutas, clasificarlas por orden creciente de coste, y recorrerlas hasta que alguna cumpla con la calidad requerida.

El caso particular  $P_j = 1$  corresponde a circuitos virtuales formados por enlaces de alta calidad. Si además se desea que todos estos enlaces sean con conexión hdlc, se le deberá indicar al nivel de red mediante un indicador en la primitiva de conexión.

## 7.4 Primitiva de establecimiento de conexión

Finalmente, para clarificar y plasmar los conceptos anteriores, veamos como son las primitivas de conexión en notación OSI (ver “Redes de Ordenadores” [14]).

Las dos primitivas en sentido de ida son:

N-CONNECT Request (calle, caller, acks\_wanted, exp\_wanted, qos, user\_data)

N-CONNECT Indication (calle, caller, acks\_wanted, exp\_wanted, qos, user\_data)

calle, caller Direcciones de red del nodo llamante y llamado.

acks\_wanted, exp\_wanted Indicadores de confirmación y de envío de datos acelerados, no soportados por Jnet.

qos Calidad de servicio. En ella se incluyen 4 parámetros:

$P_{ida}$	Probabilidad mínima de no pérdida del circuito virtual de ida.
$Con_{ida}$	Indicador booleano, verdadero si se desea que el circuito virtual de ida sólo contenga enlaces con conexión hdlc, falso en caso contrario.
$P_{vuelta}$	Probabilidad mínima de no pérdida del circuito virtual de vuelta.
$Con_{vuelta}$	Indicador booleano, verdadero si se desea que el circuito virtual de vuelta sólo contenga enlaces con conexión, falso en caso contrario.

user\_data Campo para datos de usuario, que el nivel de red transporta transparentemente.

Las dos primitivas en sentido de vuelta son:

N-CONNECT Response (responder, acks\_wanted, exp\_wanted, qos, user\_data)

N-CONNECT Confirmation (responder, acks\_wanted, exp\_wanted, qos, user\_data)

responder Direcciones de red del nodo que responde.

acks\_wanted, exp\_wanted Indicadores de confirmación y de envío de datos acelerados, no soportados por Jnet.

qos Calidad de servicio. En ella se incluyen 2 parámetros:

$P_{vuelta}$  Probabilidad mínima de no pérdida del circuito virtual de vuelta.

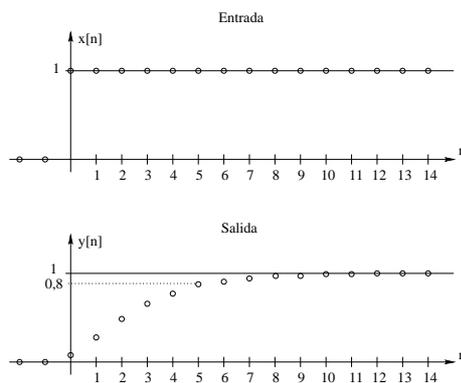
$Con_{vuelta}$  Indicador booleano, verdadero si se desea que el circuito virtual de vuelta sólo contenga enlaces con conexión, falso en caso contrario.

user\_data Campo para datos de usuario, que el nivel de red transporta transparentemente.

# Apéndice A

## Filtro digital para mediciones síncronas

Se trata de diseñar un filtro digital paso bajo, cuya respuesta al escalón unidad sea aproximadamente 0,8 para la salida en  $n=5$ .



El filtro será un filtro definido por ecuaciones en diferencias de coeficientes constantes, es decir definido por la ecuación:

$$\sum_{k=0}^N a_k y[n-k] = \sum_{k=0}^M b_k x[n-k]$$

La transformada Z de la respuesta al impulso de este filtro es:

$$H(z) = \frac{\sum_{k=0}^M b_k z^{-k}}{\sum_{k=0}^N a_k z^{-k}} = \frac{Y(z)}{X(z)} \quad (\text{A.1})$$

Queremos que  $H(z)$  sea un filtro paso bajo con  $N$  polos en  $z = \beta$ .

Para ello,  $X(z)$  debe tener  $N$  ceros en  $z = \beta$ , y puede expresarse de la forma:

$$\begin{aligned} X(z) &= (z^{-1} - \alpha)^N \\ \alpha &= \beta^{-1} \end{aligned}$$

$H(z)$  va a tener la forma:

$$H(z) = \frac{cte}{(z^{-1} - \alpha)^N}$$

Aplicando el binomio de Newton:

$$X(z) = (z^{-1} - \alpha)^N = \sum_{k=0}^N C_k^N (z^{-1})^k (-\alpha)^{N-k} = \sum_{k=0}^N C_k^N (-\alpha)^{N-k} z^{-k}$$

Es decir:

$$H(z) = \frac{cte}{\sum_{k=0}^N C_k^N (-\alpha)^{N-k} z^{-k}}$$

Identificando con (A.1), vemos que:

$$a_k = C_k^N (-\alpha)^{N-k}$$

Se desea que el filtro paso bajo tenga ganancia unidad en la frecuencia  $\omega = 0$ , por lo tanto:

$$H(1) = 1 = \frac{cte}{(1 - \alpha)^N} \Rightarrow cte = (1 - \alpha)^N$$

Para que el filtro sea estable, sus polos deben de estar en el interior de la circunferencia unidad. Vamos a escoger como polo múltiple  $\beta = \frac{1}{2}$ , es decir  $\alpha = 2$ .

Por lo tanto:

$$cte = (1 - \alpha)^N = (1 - 2)^N = (-1)^N$$

Con  $\alpha = 2$ , la ecuación en diferencias de coeficientes constantes del filtro es:

$$\sum_{k=0}^N C_k^N (-2)^{N-k} y[n-k] = (-1)^N x[n] \quad (\text{A.2})$$

$$\Leftrightarrow C_0^N (-2)^N y[n] = (-1)^N x[n] - \sum_{k=1}^N C_k^N (-2)^{N-k} y[n-k] \quad (\text{A.3})$$

$$\Leftrightarrow y[n] = \frac{1}{(-2)^N} [(-1)^N x[n] - \sum_{k=1}^N C_k^N (-2)^{N-k} y[n-k]] \quad (\text{A.4})$$

El orden N del filtro se determina mediante la condición  $y[5] \simeq 0,8$  en la respuesta al escalón unidad.

Para tal fin, se recurre a la simulación, mediante un programa sencillo escrito en C, que se expone a continuación<sup>1</sup>:

```
/*
*****
filtro_digital.c Por Julián Muñoz - 01/06/99
Calcula la respuesta al escalón de un filtro
paso bajo de orden N con polos en z=0.5
Npolos es el número de Polos, se corresponde
con el N del apéndice "filtro digital para
mediciones síncronas".
*****/
#include <stdio.h>
#include <string.h> /* memset */
#include <math.h>
unsigned int factorial
    (unsigned int n) {
    unsigned int i=1;
    unsigned int temp=1;
    if (n==0) return(1);
    for (i=1, temp=1; i<=n ; i++)
        temp=temp*i;
    return (temp);
}
unsigned int Cnm
    (unsigned int n, unsigned int m) {
    return(factorial(n)/factorial(m)/
        factorial(n-m));
}
main ( int argc, char *argv[]) {
    unsigned int Npolos;
    unsigned int prueba;
    /* Numero de pasos */
    unsigned int Npasos;
    /* Memoria del filtro, donde se almacena las salidas
y[k], se corresponde con y[n-k] del proyecto fin de
carrera */
    float y[100];
    /* pasos */
    int i=0;
```

---

<sup>1</sup>Es necesario compilar este programa con la librería matemática estándar. Compilable por ejemplo con el comando “gcc -lm filtro\_digital.c -o filtro\_digital”

```
/* salida temporal del filtro */
float f=0.0;
float alfa=2.0;
int k;
printf ("\n Cuantos polos ? ");
scanf ( "%u", &Npolos);
printf ("\n Cuantos pasos ? ");
scanf ( "%u", &Npasos);

/* Se pone a cero la memoria del filtro */
memset ( &y ,0, 100*sizeof(y[0]));
/* Bucle principal */
for ( i=0 ; i<=Npasos ; i++) {
  /* Cálculo del sumatorio */
  for (f=0.0, k=1; k<=Npolos ; k++)
    f=f+ Cnm(Npolos,k) *
      ( pow (-alfa, Npolos-k) )
      * y[k] ;
  f= (1 * pow(-1,Npolos) -f)/
    pow (-alfa,Npolos);
  printf ("\n y[%i] = %f ", i, f);
  y[0]=f;
  /* desplazamiento */
  for (k=Npolos ; k>0 ; k--) y[k]=y[k-1];
}
}
```

Se prueba aumentar el orden del filtro, hasta encontrar uno que satisfaga la condición  $y[5] \simeq 0.8$ .

**1 Polo:**

```
Cuantos polos ? 1
Cuantos pasos ? 5
y[0] = 0.500000
y[1] = 0.750000
y[2] = 0.875000
y[3] = 0.937500
y[4] = 0.968750
y[5] = 0.984375
```

**2 polos:**

Cuantos polos ? 2  
Cuantos pasos ? 5  
y[0] = 0.250000  
y[1] = 0.500000  
y[2] = 0.687500  
y[3] = 0.812500  
y[4] = 0.890625  
y[5] = 0.937500

**3 polos:**

Cuantos polos ? 3  
Cuantos pasos ? 5  
y[0] = 0.125000  
y[1] = 0.312500  
y[2] = 0.500000  
y[3] = 0.656250  
y[4] = 0.773438  
y[5] = 0.855469

**4 polos:**

Cuantos polos ? 4  
Cuantos pasos ? 5  
y[0] = 0.062500  
y[1] = 0.187500  
y[2] = 0.343750  
y[3] = 0.500000  
y[4] = 0.636719  
y[5] = 0.746094

**5 polos:**

Cuantos polos ? 5  
Cuantos pasos ? 5  
y[0] = 0.031250  
y[1] = 0.109375  
y[2] = 0.226562  
y[3] = 0.363281  
y[4] = 0.500000  
y[5] = 0.623047

Observamos que un filtro de orden  $N=4$  es suficiente.

Desarrollando la expresión (A.4) para  $N=4$  obtenemos la ecuación del filtro:

$$\begin{aligned}y[n] &= \frac{1}{16} [x[n] - C_1^4(-2)^3y[n-1] - C_2^4(-2)^2y[n-2] \\ &\quad - C_3^4(-2)^1y[n-3] - C_4^4(-2)^0y[n-4]] \\y[n] &= \frac{1}{16} [x[n] + 32y[n-1] - 24y[n-2] + 8y[n-3] - y[n-4]]\end{aligned}$$

# Apéndice B

## Parámetros del nodo

---

Para su funcionamiento, Jnet requiere constantes locales, definidas en cada nodo. Este apéndice las recopila a continuación, exponiendo:

- Su descripción resumida.
  - Sus unidades.
  - Su ámbito (genérica de ámbito global al nodo, o asociada a cada canal).
  - El valor que toma por defecto (son valores adecuados para instalaciones normales con enlaces entre 1200 y 9600 baudios).
  - La referencia del apartado y la página de este documento en la que se define esta constante.
- 

**Bi** Velocidad bruta en bps ofrecida al nivel de enlace por el interfaz radio. Por ejemplo 1200 bps, 9600 bps, 19200 bps.  
Valor por defecto: 1200 bps.  
Constante asociada a canal.

**Treq** Período (en segundos) con el que la máquina de gestión de red muestrea (o consulta) las mediciones síncronas (tasa de bits y capacidad de enlaces) de las máquinas de medición de enlaces a nivel de red de un cierto canal.  
Valor por defecto: 10 s.  
Constante asociada a canal.  
Definido en el apartado 5.4.2.1, página 52.

**Tasin** Tiempo (en segundo) máximo de consideración de la actual salida del filtro asíncrono para el cálculo de su siguiente salida.  
Valor por defecto: 1800 s.  
Constante asociada a canal.  
Definido en el apartado 5.4.2.2, página 54.

- TPj** Período (en segundo) de transmisión del número de paquetes transmitidos en un determinado canal.  
Valor por defecto: 600 s.  
Constante asociada a canal.  
Definido en el apartado 5.5.1, página 57.
- Npaqtxmin** Número mínimo de paquetes transmitidos para lanzar la señalización de medición de probabilidad de error de paquete en el nodo origen de un enlace.  
Valor por defecto: 50 paquetes.  
Constante asociada a canal.  
Definido en apartado 5.5.1, página 57.
- VmaxPj, VmaxCj, VmaxFj, VmaxHj** Valor máximo de la salida de los cuantificadores de los parámetros primarios Pj, Cj, Fj o Hj.  
Valor por defecto: VmaxPj=1, VmaxCj=Bi, VmaxFj=Bi, VmaxHj=Bi.  
Constante asociada a canal.  
Definido en apartado 5.4.3, página 55.
- NintPj, NintCj, NintFj, NintHj** Número de intervalos de cuantificación de los cuantificadores de Pj, Cj, Fj o Hj.  
Valor por defecto: NintPj=16, NintCj=20, NintFj=21, NintHj=19.  
Constante asociada a canal.  
Definido en apartado 5.4.3, página 55.
- TFj** Período (en segundos) de medición de Fj .  
Valor por defecto: 10s.  
Constante asociada a canal.  
Definido en apartado 5.6.1, página 60, y apartado 5.6.2 página 61.
- TFjwin** Ventana de medición de Fj  
Valor por defecto: 120s.  
Constante asociada a canal.  
Definido en apartado 5.6.1, página 60, y apartado 5.6.2 página 61.
- Nrafmin** Longitud mínima de ráfaga.  
Valor por defecto: 0  
Constante asociada a canal.  
Definido en apartado 5.7.2.3, página 66.
- K** Tamaño (en bits) de bloque de leaky bucket para acceso a la red.  
Valor por defecto: 500 bits.  
Constante de ámbito global al nodo.  
Definido en apartado 6.1.7, página 83.

**A** Probabilidad mínima de no pérdida de señalización en un bucle.

Valor por defecto: 0.9

Constante de ámbito global al nodo.

Definido en apartado 6.2.4.3, página 91.

**T1INIT** Valor inicial (en segundos) del temporizador de retransmisión por back pressure directo.

Valor por defecto: 50s.

Constante de ámbito global al nodo.

Definido en apartado 6.2.4.4, página 91.

**T2INIT** Valor inicial (en segundos) del temporizador de retransmisión por back pressure indirecto

Valor por defecto: 120s.

Constante de ámbito global al nodo.

Definido en apartado 6.2.4.4, página 91.

**Ncolatx1** Umbral (en bits) de descongestión de enlaces con conexión hdlc, asociado a cada canal.

Valor por defecto: 300 bits.

Constante asociada a canal.

Definido en el apartado 6.2.14.1, página 116.

**Ncolatx2** Umbral (en bits) de congestión de enlaces con conexión hdlc.

Valor por defecto: 5000 bits.

Constante asociada a canal.

Definido en el apartado 6.2.14.1, página 116.

**OC1** Umbral de descongestión de enlaces sin conexión hdlc

Valor por defecto: 0.7

Constante asociada a canal.

Definido en el apartado 6.2.14.2, página 116.

**OC2** Umbral de congestión de enlaces sin conexión hdlc.

Valor por defecto: 0.9

Constante asociada a canal.

Definido en el apartado 6.2.14.2, página 116.

# **Apéndice C**

## **Primitivas**

A continuación se presenta una recopilación de las primitivas utilizadas en este documento.

Origen	Destino	Primitiva	Observación	Apartado
Máquina de gestión de red	Máquina de medición de enlace a nivel de enlace	MEA-SENT Request (dirección AX.25)	Petición del número de paquetes transmitidos por nuestro nodo	5.5.1 Medición de Pj en nodo origen
Máquina de medición de enlace a nivel de enlace	Máquina de gestión de red	MEA-SENT Confirm (numero de paquetes transmitidos)	Respuesta a la petición anterior	5.2.2 Especificación
Máquina de gestión de red	Máquina de medición de enlace a nivel de enlace	MEA-HEARD Request (dirección AX.25)	Petición del número de paquetes recibidos por nuestro nodo provenientes de un determinado indicativo	5.5.2 Medición de Pj en nodo destino
Máquina de medición enlace a nivel de enlace	Máquina de gestión de red	MEA-HEARD Confirm (numero de paquetes oídos, tiempo transcurrido).	Respuesta a la petición anterior	5.2.2 Especificación
Máquina de medición de enlace a nivel de red, máquina de gestión de red	Máquina de medición de enlace a nivel de enlace	MEA-OCCUPATION Request	Petición del factor de ocupación del canal	6.2.15 Detección de congestión, 5.7.3 medición del ancho de banda disponible
Máquina de medición de enlace a nivel de enlace	Máquina de medición de enlace a nivel de red, máquina de gestión de red	MEA-OCCUPATION Confirm (Ocupación)	Respuesta a la petición anterior	5.2.2 Especificación
Decodificador de señalización	Máquina de gestión de red	NM-Error Indication (Canal, Dirección AX.25, Npaqtx)	Indicación de que el nodo del canal y dirección AX.25 indicados anuncia haber transmitido Npaqtx paquetes. Utilizado para la medición de la probabilidad de no error de paquete.	5.5.2 Medición de Pj en nodo destino
Encaminador, Máquina de gestión de red	Máquina de medición de enlace a nivel de red	NIMEA-DATA Request	Petición de envío de datos a través de un enlace con conexión hdlc	5.6.1, 5.7.2; Medición de Fj y Cj en enlaces con conexión hdlc
Encaminador, Máquina de gestión de red	Máquina de medición de enlace a nivel de red	NIMEA-UNIT-DATA Request	Petición de envío de datos a través de un enlace sin conexión hdlc	

Origen	Destino	Primitiva	Observación	Apartado
Máquina de gestión de red	Máquina de medición de enlace a nivel de red	NMEA-CJ Request	Petición del flujo de bits transportado por un enlace con conexión <i>hdlc</i>	5.7 Medición de Cj
Máquina de medición de enlace a nivel de red	Máquina de gestión de red	NMEA-CJ Confirm(Cj)	Respuesta a la petición anterior	
Máquina de gestión de red	Máquina de medición de enlace a nivel de red	NMEA-FJ Request	Petición del flujo de bits transportado por un enlace	5.6 Medición de Fj
Máquina de medición de enlace a nivel de red	Máquina de gestión de red	NMEA-FJ Confirm(Fj)	Respuesta a la petición anterior	
Máquina de medición de enlace a nivel de red	Máquina de circuito virtual	VC-CONGEST Indication	Aviso de congestión	6.2.15
Máquina de medición de enlace a nivel de red	Máquina de circuito virtual	VC-ENDCONGEST Indication	Aviso de fin de congestión	6.2.15
Máquina de medición de enlace a nivel de red	Segmentador	SEG-DATA Request, SEG-UNIT-DATA Request	Ver apartado 5.3	5.3 Ampliación del segmentador AX.25
Segmentador	Máquina de medición de enlace a nivel de red	SEG-DATA Indication, SEG-UNIT-DATA Indication, SEG-DATA Confirm		

## Apéndice D

# Medición de la capacidad de un canal a partir de la longitud media de la cola

Mediante la teoría de colas y modelando el sistema como M/M/1 o M/D/1, es posible obtener la capacidad teórica de un canal, conocidos dos valores:

El número de bits  $F_j$  cursados por segundo.

La longitud media de la cola ( $\bar{Q}$ ), o en número medio de usuario en el sistema ( $\bar{N}$ ).

En el caso de  $\bar{N}$ , el cálculo es un poco más complejo, pero generalmente es más fácil conocer  $\bar{N}$  que  $\bar{Q}$ , porque  $\bar{N}$  solo requiere un contador que se incremente al entrar un usuario en el sistema y se decremente al salir del sistema; el cálculo de  $\bar{Q}$  requiere este mismo decremento pero a la entrada del servidor, punto que, en ciertos casos, es más difícil de definir con exactitud.

El análisis matemático resumido es el siguiente:

$$(1) \left. \begin{array}{l} \rho = \frac{F_j}{C_j} \\ \bar{N}, F_j \Rightarrow \rho \end{array} \right\} \Rightarrow C_j$$

El paso (1) se realiza gracias a las fórmulas de Pollaceek-Khintchine:

$$\begin{aligned} \bar{Q} &= \frac{\rho^2(1 + C_s^2)}{2(1 - \rho)} \\ C_s^2 &= \frac{\sigma_s^2}{E^2[s]} \\ \bar{N} &= \bar{Q} + \rho \end{aligned}$$

En el caso de una distribución de servicio determinista (M/D/1):

$$E[s] = \frac{1}{\mu}; E[s^2] = \frac{1}{\mu^2} \Rightarrow C_s^2 = 0 \Rightarrow \bar{Q} = \frac{\rho^2}{2(1 - \rho)}$$

APÉNDICE D. MEDICIÓN DE LA CAPACIDAD DE UN CANAL A PARTIR DE LA LONGITUD MEDIA DE LA COLA

Por lo tanto:

$$\begin{aligned}\bar{N} &= \rho + \frac{\rho^2}{2(1-\rho)} \Leftrightarrow 2(\bar{N} - \rho)(1 - \rho) = \rho^2 \\ 2(\bar{N} - \rho\bar{N} - \rho + \rho^2) &= \rho^2 \Leftrightarrow -\rho^2 + 2\rho(\bar{N} + 1) - 2\bar{N} = 0 \\ &\Leftrightarrow \rho^2 - 2(\bar{N} + 1)\rho + 2\bar{N} = 0\end{aligned}$$

Se trata de una ecuación de segundo grado:

$$\begin{aligned}\rho_{1,2} &= \frac{2(\bar{N} + 1) \pm \sqrt{4(\bar{N} + 1)^2 - 4 \times 2\bar{N}}}{2} = (\bar{N} + 1) \pm \sqrt{(\bar{N} + 1)^2 - 2\bar{N}} \\ \rho_{1,2} &= \bar{N} + 1 \pm \sqrt{\bar{N}^2 + 1}\end{aligned}$$

Pero  $\rho_2$  no es válido ya que:

$$\rho_2 = \bar{N} + 1 + \sqrt{\bar{N}^2 + 1} \Rightarrow \rho_2 > 1$$

Por lo tanto:

$$\rho = \bar{N} + 1 - \sqrt{\bar{N}^2 + 1}$$

Y obtenemos la capacidad en función de  $F_j$  y  $\bar{N}$ :

$$C_j = \frac{F_j}{\bar{N} + 1 - \sqrt{\bar{N}^2 + 1}}$$

## Apéndice E

### Función umbral del retardo incremental total $L(r)$

Partimos de estas 3 ecuaciones:

$$L(0) = \frac{25}{C_{min}} = \frac{A}{B} + D = L_o \quad (1)$$

$$L(C_{min}) = \varepsilon = \frac{A}{B + C_{min}} + D \quad (2)$$

$$L\left(\frac{C_{min}}{10}\right) = L_o - \frac{L_o - \varepsilon}{2} = \frac{A}{B + \frac{C_{min}}{10}} + D = K \quad (3)$$

Despejemos  $A, B, D$ :

$$(1) \Rightarrow D = L_o - \frac{A}{B}$$

$$(2) \Rightarrow \varepsilon = \frac{A}{B + C_{min}} + L_o - \frac{A}{B} \Leftrightarrow \varepsilon - L_o = \frac{A \times B - A(B + C_{min})}{B(B + C_{min})} \Leftrightarrow \varepsilon - L_o = \frac{-A \times C_{min}}{B(B + C_{min})}$$

$$\Leftrightarrow \frac{-B(B + C_{min})}{C_{min}}(\varepsilon - L_o) = A \quad (4)$$

$$(3) \Rightarrow K = \frac{\frac{-B(B + C_{min})}{C_{min}}(\varepsilon - L_o)}{B + \frac{C_{min}}{10}} + L_o + \frac{B + C_{min}}{C_{min}}(\varepsilon - L_o)$$

Sea  $H = \frac{B}{C_{min}}$ :

$$K = \frac{-(H+1)(\varepsilon - L_o)}{1 + \frac{1}{10H}} + L_o + (H+1)(\varepsilon - L_o) = (H+1)(\varepsilon - L_o)\left(-\frac{1}{1 + \frac{1}{10H}} + 1\right) + L_o$$

Como  $-\frac{1}{1 + \frac{1}{10H}} + 1 = -\frac{10H}{10H+1} + \frac{10H+1}{10H+1} = \frac{1}{10H+1}$ , tenemos:

$$K - L_o = \frac{H+1}{10H+1}(\varepsilon - L_o) \Rightarrow (10H+1)(K - L_o) = (H+1)(\varepsilon - L_o)$$

$$\Leftrightarrow H \times 10(K - L_o) - H(\varepsilon - L_o) = \varepsilon - L_o - (K - L_o)$$

$$\Leftrightarrow H = \frac{(\varepsilon - L_o) - (K - L_o)}{10(K - L_o) - (\varepsilon - L_o)}$$

Por lo tanto:

$$B = H \times C_{min} = -C_{min} \left( \frac{V - W}{V - 10W} \right)$$

Siendo:

$$V = \varepsilon - L_o$$

$$(3) \Rightarrow \left. \begin{array}{l} W = K - L_o \\ K = L_o - \frac{L_o - \varepsilon}{2} = \frac{L_o + \varepsilon}{2} \end{array} \right\} \Rightarrow W = \frac{L_o + \varepsilon}{2} - L_o = \frac{\varepsilon - L_o}{2} = \frac{V}{2} = W$$

Obtenemos:

$$B = -C_{min} \left( \frac{V - \frac{V}{2}}{V - 5V} \right) \Leftrightarrow$$

$$B = \frac{C_{min}}{8} \quad (5)$$

$$\left. \begin{array}{l} (4) \\ (5) \end{array} \right\} \Rightarrow A = \frac{-B(B + C_{min})}{C_{min}} (\varepsilon - L_o) = -\frac{C_{min}}{8 \times C_{min}} \left( \frac{C_{min}}{8} + C_{min} \right) (\varepsilon - L_o)$$

$$\Leftrightarrow A = -\frac{1}{8} \left( \frac{9C_{min}}{8} \right) (\varepsilon - L_o) = \frac{9C_{min}}{64} (L_o - \varepsilon) \Leftrightarrow$$

$$A = \frac{9C_{min}}{64} (L_o - \varepsilon) \quad (6)$$

$$\left. \begin{array}{l} (1) \\ (5) \\ (6) \end{array} \right\} \Leftrightarrow D = L_o - \frac{A}{B} = L_o - \frac{9C_{min}}{64} (L_o - \varepsilon) \times \frac{8}{C_{min}} = L_o - \frac{9}{8} (L_o - \varepsilon) \Leftrightarrow$$

$$\Leftrightarrow D = -\frac{L_o}{8} + \frac{9}{8} \varepsilon \Leftrightarrow$$

$$D = \frac{9\varepsilon - L_o}{8} \quad (7)$$

## Apéndice F

# Control de flujo de acceso a la red: implementación del algoritmo de resolución numérica de $r_{max}$ .

---

El programa expuesto a continuación es el código fuente en lenguaje C de la implementación del algoritmo de resolución numérica de  $r_{max}$ . Puede ser compilado en compiladores compatibles ANSI C.

---

```
#include <stdio.h>

/* Simulación de redes con el control de flujo de acceso a la red */
/* Jnet */

double Dt (int enlaces, double C[], double F[], double deltaf)
{
double valor=0;
int contador=0;
for (;contador<enlaces;contador++)
valor=valor+(C[contador])/((C[contador] - F[contador] - deltaf)
*(C[contador] - F[contador] - deltaf ));
return valor;
};

double L (double a, double b, double d ,double r) {
return (a/(b+r)+d);
};
```

*APÉNDICE F. CONTROL DE FLUJO DE ACCESO A LA RED:  
IMPLEMENTACIÓN DEL ALGORITMO DE RESOLUCIÓN NUMÉRICA DE*  
**R<sub>MAX</sub>**.

---

```
double min (double min1, double min2)
{
```

```
    if (min1<min2) return (min1);
    else return (min2);
};
```

```
double Cmin (int enlaces, double C[])
{
    double Minimo;
    int contador=0;
    Minimo=C[0];
    for (;contador<enlaces;contador++)
        Minimo=min (Minimo,C[contador]);
    return Minimo;
}
```

```
double epsilon (int enlaces, double C[])
{
    double e=0;
    int contador=0;
    for (;contador<enlaces;contador++)
        e=e+1/C[contador];
    return e;
}
```

```
main (int arg, char *(argv[] )
{
```

```
    int contador;
    int enlaces;
    double C[100];
    double F[100];
    double CmenosF[100];
    double DT;
    double E;
    double CMIN;
    double A;
    double B;
    double D;
    double CLO;
```

*APÉNDICE F. CONTROL DE FLUJO DE ACCESO A LA RED:  
IMPLEMENTACIÓN DEL ALGORITMO DE RESOLUCIÓN NUMÉRICA DE*  
**R<sub>MAX</sub>**.

---

```
double LO;
double ractual;
double Deltax;

printf ("\n ¿ Cuántos enlaces ? ");
scanf ( "%i", &enlaces);
printf (" Numero = %i", enlaces);

for ( contador = 0 ; contador < enlaces ; contador++) {
    printf ("\nC%i :",contador);
    scanf ("%lf", C+contador);
    printf ("F%i :", contador);
    scanf ("%lf", F+contador);
    printf ("Capacidad: %f ; Flujo: %f \n\n", C[contador], F[contador]);
};

printf (" d(pm) : ");
scanf ("%lf", &CLO);

DT=Dt(enlaces,C,F,0);
printf ("\nDt=%f",DT);
E=epsilon(enlaces, C);
CMIN=Cmin(enlaces, C);
printf ("\nE=%f",E);
printf ("\nCMIN=%f",CMIN);

LO=CLO/CMIN;
printf ("\nLO=%f",LO);

A= 9.0/ 64.0 *CMIN*(LO-E);
B=CMIN/8.0;
D= (9.0*E -LO)/8.0;
printf ("\nA=%f",A);
printf ("\nB=%f",B);
printf ("\nD=%f",D);

printf ("\nractual :");
scanf ("%lf", &ractual);
```

*APÉNDICE F. CONTROL DE FLUJO DE ACCESO A LA RED:  
IMPLEMENTACIÓN DEL ALGORITMO DE RESOLUCIÓN NUMÉRICA DE*  
**R<sub>MAX</sub>**.

---

```
/* Calculo C-F */
for ( contador = 0 ; contador < enlaces ; contador++)
    CmenosF[contador]=C[contador]-F[contador];

printf("Minimo de C-F :%f", Cmin(enlaces, CmenosF));
Deltax= (min(CMIN, Cmin(enlaces, CmenosF) + ractual))/100.0;
printf ("\nDeltaX = %f", Deltax);

/* Búsqueda de rmax */
for (contador=0;
    (contador < 100) && ( Dt(enlaces, C, F, contador*Deltax-ractual )
        < L(A,B,D,contador*Deltax) )
    ; contador++);

printf("\nrmax = %f", contador*Deltax);
printf("\ni = %i",contador);
printf("\nDt=%f", Dt(enlaces, C, F, contador*Deltax-ractual) );
printf("\nL=%f", L(A,B,D,contador*Deltax) );

}
```

# Apéndice G

## Información acerca de este documento

**JNET: NIVEL DE RED PARA RADIO-PAQUETE AX.25** es copyright (C) 1999 Universidad Politécnica de Madrid.

Este documento es de contenido abierto (OpenContent). Puede ser redistribuido y/o modificado bajo los términos de la Licencia OpenContent (OPL) versión 1.0, que exponemos en idioma inglés en el apéndice H.

Este documento se distribuye con la esperanza de que sea útil, pero SIN NINGUNA GARANTÍA; sin incluso la garantía implícita de COMERCIALIZABILIDAD o CONVENIENCIA PARA UN PROPÓSITO PARTICULAR. Vea la Licencia OpenContent para más detalles.

Este documento o cualquier nueva versión del mismo podrán encontrarse en Internet en la dirección web:

<http://www.etsit.upm.es/~rclub/socios/julian/jnet/>

A continuación se encuentra una tabla a rellenar con las modificaciones realizadas posteriormente a la publicación del documento original:

<b>Versión</b>	<b>Fecha</b>	<b>Autor</b>	<b>Cambios</b>
1.0	6/9/1999	Julián Muñoz Domínguez	Documento original

# Apéndice H

## Licencia de uso

### OpenContent License (OPL)

**Version 1.0, July 14, 1998.**

This document outlines the principles underlying the OpenContent (OC) movement and may be redistributed provided it remains unaltered. For legal purposes, this document is the license under which OpenContent is made available for use.

The original version of this document may be found at <http://www.opencontent.org/opl.shtml>

### LICENSE

Terms and Conditions for Copying, Distributing, and Modifying Items other than copying, distributing, and modifying the Content with which this license was distributed (such as using, etc.) are outside the scope of this license.

1. You may copy and distribute exact replicas of the OpenContent (OC) as you receive it, in any medium, provided that you conspicuously and appropriately publish on each copy an appropriate copyright notice and disclaimer of warranty; keep intact all the notices that refer to this License and to the absence of any warranty; and give any other recipients of the OC a copy of this License along with the OC. You may at your option charge a fee for the media and/or handling involved in creating a unique copy of the OC for use offline, you may at your option offer instructional support for the OC in exchange for a fee, or you may at your option offer warranty in exchange for a fee. You may not charge a fee for the OC itself. You may not charge a fee for the sole service of providing access to and/or use of the OC via a network (e.g. the Internet), whether it be via the world wide web, FTP, or any other method.
2. You may modify your copy or copies of the OpenContent or any portion of it, thus forming works based on the Content, and distribute such modifications or

work under the terms of Section 1 above, provided that you also meet all of these conditions:

- (a) You must cause the modified content to carry prominent notices stating that you changed it, the exact nature and content of the changes, and the date of any change.
  - (b) You must cause any work that you distribute or publish, that in whole or in part contains or is derived from the OC or any part thereof, to be licensed as a whole at no charge to all third parties under the terms of this License, unless otherwise permitted under applicable Fair Use law.  
These requirements apply to the modified work as a whole. If identifiable sections of that work are not derived from the OC, and can be reasonably considered independent and separate works in themselves, then this License, and its terms, do not apply to those sections when you distribute them as separate works. But when you distribute the same sections as part of a whole which is a work based on the OC, the distribution of the whole must be on the terms of this License, whose permissions for other licensees extend to the entire whole, and thus to each and every part regardless of who wrote it. Exceptions are made to this requirement to release modified works free of charge under this license only in compliance with Fair Use law where applicable.
3. You are not required to accept this License, since you have not signed it. However, nothing else grants you permission to copy, distribute or modify the OC. These actions are prohibited by law if you do not accept this License. Therefore, by distributing or translating the OC, or by deriving works herefrom, you indicate your acceptance of this License to do so, and all its terms and conditions for copying, distributing or translating the OC.

#### **NO WARRANTY**

4. BECAUSE THE OPENCONTENT (OC) IS LICENSED FREE OF CHARGE, THERE IS NO WARRANTY FOR THE OC, TO THE EXTENT PERMITTED BY APPLICABLE LAW. EXCEPT WHEN OTHERWISE STATED IN WRITING THE COPYRIGHT HOLDERS AND/OR OTHER PARTIES PROVIDE THE OC "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK OF USE OF THE OC IS WITH YOU. SHOULD THE OC PROVE FAULTY, INACCURATE, OR OTHERWISE UNACCEPTABLE YOU ASSUME THE COST OF ALL NECESSARY REPAIR OR CORRECTION.
5. IN NO EVENT UNLESS REQUIRED BY APPLICABLE LAW OR AGREED TO IN WRITING WILL ANY COPYRIGHT HOLDER, OR ANY OTHER

PARTY WHO MAY MIRROR AND/OR REDISTRIBUTE THE OC AS PERMITTED ABOVE, BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE OC, EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

# Bibliografía

- [1] MUÑOZ, J.  
Comunicaciones digitales en el entorno de la radioafición.  
<http://www.etsit.upm.es/~rclub/socios/julian/julian.html>
  
- [2] FOX, J.  
AX.25 Amateur Packet Radio Link-Layer Protocol Version 2.0.  
Newington, USA.  
American Radio Relay League.  
1984.  
<ftp://ftp.funet.fi/pub/ham/packet/misc/ax25.doc>
  
- [3] BEECH, W. , NIELSEN, D. , TAYLOR, J.  
AX.25 Link Access Protocol for Amateur Radio, version 2.2.  
<http://www.tapr.org/tapr/pdf/AX25.2.2.pdf>
  
- [4] WELWARSKY, M. y col.  
FlexNet Homepage  
<http://dl0td.afthd.tu-darmstadt.de/~flexnet/>
  
- [5] BERTSEKAS, D. , GALLAGER, R.  
Data Networks (Second Edition).  
Prentice Hall.  
1987.  
pp 433-437, pp 500-507, pp 519-524
  
- [6] MOULTON, T.  
ROSE X.25 Packet Switch  
<http://www.rats.org/rose/>
  
- [7] MOULTON, T.  
ROSE X.25 Packet Switch  
rzsw33.zip <http://www.rats.org/rose/roseover.html>
  
- [8] MOULTON, T.  
Correo electrónico: [w2vy@xanthus.net](mailto:w2vy@xanthus.net)

Comunicación realizada por correo electrónico con el autor de rose el día 6/11/1998.

- [9] SOFTWARE 2000  
netrom.ps ftp://hes.iki.fi/pub/ham/linux/ax25/ax25-doc-1.0.tar.gz
- [10] GAL, A.  
A New Routing Specification for Packet Radio Datagram Networks.  
<http://www.nordlink.org/eng/inp3.htm>
- [11] KUO, F. y col.  
Protocols & Techniques for Data Communications Networks.  
Prentice-Hall.  
1981.  
pp 149-169
- [12] THAKER, G., CAIN, J.  
Interactions Between Routing and Flow Control Algorithms.  
IEEE Transactions on Communications.  
Vol. com-43, n°3, marzo 1986, pp 269-274
- [13] KARN, P., PARTRIDGE, C.  
Improving Round-Trip Time Estimates in Reliable Transport Protocols.  
Proceedings of ACM SIGCOMM '87.  
11-13 Agosto 1987, pp 2-7  
(Disponible en <http://people.qualcomm.com/karn>)
- [14] TANENBAUM, A.  
Redes de ordenadores.  
Prentice-Hall.  
1991.  
pp 324-325