

92



BIBLIOTECA
FAC. DE INFORMÁTICA
UNLP.

Trabajo de Grado

“Encapsulamiento de IPv6 en Redes ATM”



Autores:

Sergio H. Bonfiglio
Hector A. Bortolotto
Damián S. Tolosa

Directores:

Ing. Luis Marrone
Lic. Javier Diaz

Facultad de Ciencias Exactas
Universidad Nacional de La Plata
1999

<p>TES 99/4 DIF-02069 SALA</p>	<p> UNIVERSIDAD NACIONAL DE LA PLATA FACULTAD DE INFORMÁTICA Biblioteca 50 y 120 La Plata catalogo.info.unlp.edu.ar biblioteca@info.unlp.edu.ar</p> <p> DIF-02069</p>
--	---

DONACION.....
\$.....
Fecha..... 29-9-05
Inv. E..... Inv. B. 2069

TES
9914



...A nuestros Padres



Agradecemos a

Ing. Luis Marrone y Lic. Javier Diaz por el apoyo recibido en el desarrollo de ésta Tesis.

Facultad de Ciencias Exactas de la U.N.L.P. por la formación desinteresada.

...y a nuestros seres queridos por la paciencia en todos estos años



Indice

Prefacio	VIII
CAPITULO 1	1
Redes ATM.....	1
1. Introducción	1
2. Generalidades	1
3. Tipos de Circuitos	2
4. Paths, Circuitos e Identificadores	3
5. Celdas	4
6. Modelo en Capas.....	5
7. Direccionamiento	10
8. ATM Switching.....	11
9. Tipos de conexión.....	12
10. Señalización	14
11. Calidad de Servicio y Manejo de Tráfico	15
11.1. Funciones Genéricas.....	15
11.2. Arquitectura de Servicio ATM.....	17
11.3. Definiciones para las Categorías de Servicios:	17
11.4. Parámetros y Atributos de las Categorías de Servicios.....	19
12. ATM Layer QOS	20
12.1 Parámetros de la Calidad del Servicio.....	20
12.2. Naturaleza de los QOS acordados.....	20
13. Contrato de Tráfico	21
13.1. Parámetros y descriptores del Tráfico	21
13.2 Parámetros de Tráfico	21
13.3. Descriptores del origen del Tráfico	21
13.4. Descriptores de la conexión del Tráfico.....	21
13.5. Especificación del Contrato de Tráfico	21
14. Interface UNI 4.0.....	22
14.1. Configuración de Referencia	22
14.2. Capacidades de la interface UNI 4.0	24
14.3. Llamadas Punto a Punto	25
14.3.1. Direccionamiento	25
14.4. Leaf Initiated Join Capability (LIJ).....	27
14.5. Requerimientos de códigos	27
14.5.1. Setup.....	28
14.5.2. Modificación de los mensajes punto a multipunto	28
14.5.3. Mensajes para llamadas de LIJ y Control de Conexión	28
14.5.4. Elementos de información.....	29
14.6. Procedimientos de Signalling para Soportar la Capacidad LIJ	30
14.6.1. Procedimiento de incorporación en la interface de la hoja	30
14.6.2. Procedimiento de incorporación en la Interface de la Raíz	32
CAPITULO 2	35
IP Versión 4.....	35
1. Introducción	35
2. Definición del Nivel IP.....	35
3. Estructura del Datagrama	36



4. Ruteo	37
5. Detalles de Direccionamiento: Subredes y Broadcasting	38
6. Fragmentación y Reensamblado del Datagrama.....	39
7. ARP: Address Resolution Protocol	39
8. Sistema de Dominios.....	40

CAPITULO 3..... 42

IP Versión 6..... 42

1. Introducción	42
2. Formato del Header.....	43
3. Headers de las Extensiones de IPv6.....	44
3.1 Orden de los Headers Extention	44
3.2. Options.....	45
3.3. Hop-by-Hop Options Header	45
3.4. Routing Header	46
3.5. Fragment Header	48
3.6. Destination Options Header	51
3.7. No Next Header.....	52
4. Consideraciones del tamaño del paquete.....	52
5. Etiquetado de Flujo.....	52
6. Clases de Tráfico.....	54
7. Condiciones de Protocolos de Nivel Superior.....	55
7.1. Checksums de Nivel Superior	55
7.2. Tiempo de vida máximo del paquete.....	56
7.3. Máximo tamaño de Payload de nivel superior.....	56
7.4. Respondiendo a paquetes que acarrean Routing Header.....	56
8. Arquitectura del Direccionamiento en IPv6.....	56
8.1. Direcciones IPv6.....	56
8.2. Modelo de Direccionamiento	57
8.3. Direcciones Conocidas.....	57
8.4. Direcciones requeridas para un nodo.....	60
8.5. Direcciones requeridas por un Router.....	61
9. Neighbor discovery	61
9.1. Modelo conceptual de un Host.....	63
9.2. Algoritmo Conceptual de Envíos	64

CAPITULO 4..... 65

IP Versión 6 sobre una LAN ATM..... 65

1. Introducción	65
2. Términos Relacionados	66
2.1 Terminología de IPv6	66
2.2. Terminología ATM.....	66
2.3. Nuevos Términos	67
3. Canales	67
4. Encapsulamiento	68
5. Consideraciones Generales del Modelo.....	69
6. Arquitectura	69
7. Grupo de Vecinos	71
8. Modelo del Servicio de Multicast	71
8.1. Servidor D'Artagnan.....	72
8.1.1 Características del Servidor D'Artagnan	72
8.1.2 Descripción de la funcionalidad de la Capa Servidora	73
8.2 Cliente	73
8.2.1 Características del Cliente	73



8.2.2 Descripción de la funcionalidad de la Capa Cliente	74
9. Implementación	76
9.1. Estructuras de Datos	76
9.1.1 Estructura de Datos para el Cliente:	76
9.1.2 Estructuras de Datos para el Servidor:	77
9.2 Algoritmos.....	79
9.2.1 Algoritmos relativos al Cliente.....	79
9.2.1.1 Introducción al Grupo de Vecinos	79
9.2.1.2 Envío de un Paquete Multicast	80
9.2.1.3 Envío de un Paquete Unicast.....	80
9.2.1.4 Determinación de QoS.....	80
9.2.1.5 Determinación del tipo de dirección IPv6.....	81
9.2.1.6 Mantenimiento de la Tabla <i>ConexActivCli_Tbl</i>	82
9.2.1.7 Mantenimiento de la Tabla <i>ConexActivSvr_Tbl</i>	82
9.2.2 Algoritmos relativos al Servidor.....	82
9.2.2.1 Solicitud de incorporación de un Vecino al Grupo	82
9.2.2.2 Distribución de paquetes Multicast	83
9.2.2.3 Mantenimiento de la tabla <i>GrupoVecinos_Tbl</i>	83
9.3 Pseudocódigo.....	84
9.3.1 Algoritmos de la Capa Cliente.....	84
9.3.1.1 Algoritmo para envío de Paquetes Multicast y Unicast	84
9.3.1.2 Algoritmo para incorporarse al Grupo de Vecinos	85
9.3.1.3 Algoritmo para determinación de QoS	86
9.3.1.4 Algoritmo de Determinación del tipo de dirección IPV6	86
9.3.1.5 Algoritmo para actualizar las entradas a la tabla <i>ConexActivCli_TBL</i>	86
9.3.1.6 Algoritmo para actualizar las entradas a la tabla <i>ConexActivSvr_TBL</i>	86
9.3.1.7 Alg. para eliminar entradas vencidas en las tablas de la Capa Cliente.....	87
9.3.2 Algoritmos de la Capa Servidora.....	87
9.3.2.1 Algoritmo para incorporar un Vecinos al Grupo	87
9.3.2.2 Algoritmo para actualizar las entradas a la tabla <i>GrupoVecinos_Tbl</i>	88
9.3.2.3 Algoritmo para la distribución de paquetes Multicast	88
9.3.2.4 Alg. para eliminar entradas vencidas en la tabla <i>GrupoVecinos_Tbl</i>	88
10. Seguridad	89
11. Replicación	89
11.1 Intercambio de datos entre Servidores D'Artagnan.....	89
11.2 Sincronización entre los Nodos y el Servidor	90
11.3. Algoritmos para la Replicación de Datos entre Servidores D'Artagnan.....	90
12. Mensajes de Control.....	91

CONCLUSIONES..... 93

APENDICE A 95

Encapsulamiento de IPv4 sobre ATM 95

1. Configuración de las Subredes IP	95
2. Concepto de Subredes Lógicas a nivel IP (LIS)	95
3. Arquitectura de LIS	95
4. Consideraciones Generales del Modelo	96
6. Servicio de Broadcast.....	96
6.1. Características del Servidor de Broadcast	97
6.2. Mecanismo para enviar mensajes de broadcast	97
7. Servicio de ARP.....	97
7.1. Características del Servidor de ARP	97
7.2. Mecanismo para resolver direcciones (ARP, InARP)	98
7.3. Formato del paquete ARP	99
8. Implementación de los Servicios de Broadcast y ARP	100
8.1. Cliente	100



8.2. Servidor.....	100
A P E N D I C E B	102
Análisis comparativo de los Modelos para IPv4 e IPv6	102
BIBLIOGRAFIA.....	105



Prefacio

El trabajo de investigación que se presenta a continuación tiene como objetivo definir un modelo teórico que permita encapsular la Versión 6 de IP sobre Redes ATM.

Este trabajo está organizado en capítulos, los iniciales detallan las características de los conceptos y tecnologías aquí empleados, mientras que el último describe el modelo propuesto para lograr el objetivo planteado. De esta manera en el Capítulo 1 se encuentra la información acerca de Redes ATM, y de la interface UNI 4.0.

En el Capítulo siguiente (Capítulo 2) se describen las características básicas de la Versión 4 del Protocolo de Redes IP, este capítulo tiene como objetivo contar con una reseña de la versión actual del protocolo IP.

La nueva Versión de IP es presentada en el Capítulo 3 de este trabajo, aquí se detallan las características del protocolo como así también los protocolos de control que IPv6 define tales como Neighbor Discovery, etc.

En el Capítulo 4 se describe el modelo teórico propuesto en este trabajo de investigación, en él se detallan el objetivo, los términos relacionados con el modelo, la arquitectura necesaria, las capas definidas, los algoritmos de cada una de ellas, y un sistema de seguridad contra pérdida de información de control necesaria para el funcionamiento de esta solución.

Por último este trabajo contiene dos apéndices, en el Apéndice A, se describe una solución al problema de encapsular la versión actual de IP (IPv4) y en el Apéndice B, se detalla un análisis comparativo entre ambas soluciones (encapsulamiento de IPv6 e IPv4 sobre Redes ATM).

CAPITULO 1

BIBLIOTECA
FAC. DE INFORMÁTICA
U.N.L.P.

Redes ATM

1. Introducción

ATM es una tecnología que por sus características puede ubicarse en nivel 2 dentro del modelo OSI. Sus siglas significan Asynchronous Transfer Mode o modo de transferencia asincrónica y esta basada en la transmisión de paquetes que no incluyen información de reloj (por ello se denominan asincrónicas). Opera en modo orientado a conexión, esto significa que cuando una estación desea comunicarse con otra, debe efectuar un pedido de conexión a la red, ésta establece un circuito fijo durante el tiempo que dura la conexión. Los paquetes utilizados en la transferencia de información son de longitud pequeña y constante, denominados celdas. El hecho de utilizar celdas favorece los siguientes comportamientos:

- Como éstas poseen la misma longitud, es muy sencillo implementar soluciones de switching basadas únicamente en hardware, de forma tal que dicha conmutación sea muy veloz, pudiendo de esta manera soportar altas velocidades de transmisión llegando a varios Gbps.
- Como la longitud es fija, es muy fácil determinar tanto el tiempo de acceso al medio de transmisión, como la latencia (retardo) de la red, fundamental para obtener transmisiones multimediales de buena calidad.

Las redes ATM tienen una tecnología de transmisión digital capaz de soportar cualquier tipo de tráfico, ya sea datos, voz, video, telefonía, etc.; sobre una gran variedad de medios y velocidades.

Otras de las ventajas de ATM es su muy buena escalabilidad, es decir, una red ATM puede caber dentro de un equipo, ser del tamaño de una LAN o cubrir el planeta entero. De esta manera, ATM unifica las tecnologías LAN y WAN, siempre distintas y a veces difíciles de conciliar en una sola.

2. Generalidades

La unidad de transmisión en ATM se denomina celda y tiene un tamaño fijo de 53 octetos, 5 de encabezado y 48 de carga (utilizados para acarrear los datos). En el encabezado se encuentra información de ruteo de la celda, su prioridad, indicadores de congestión, etc.

Una red ATM esta compuesta por switches y por estaciones. Un switch es un dispositivo electrónico especialmente diseñado para transmitir datos a muy alta velocidad. Los switches conforman el corazón de la red y las estaciones se conectan a ellos. Un switch típico soporta la conexión de entre 16 y 32 nodos. Para permitir la transmisión de datos a alta velocidad la conexión entre los nodos y el switch se realiza por medio de un par de hilos de fibra óptica (en la actualidad se esta analizando la posibilidad de usar twisted pair).

Aunque un switch ATM tiene una capacidad limitada, múltiples switches pueden intercomunicarse entre sí para formar una gran red. En particular para conectar nodos que se encuentran en dos sitios diferentes es necesario contar con un switch en cada uno de ellos y ambos deben estar conectados entre sí.

La función básica de los switches es la de rutear las celdas provenientes de las estaciones a sus destinos. Además, entre otras cosas son los encargados de efectuar las conexiones entre estaciones, avisar de congestiones, tasar el tráfico (principalmente en redes públicas) a fin de poder facturar el uso de la red, y cumplir funciones de servidores en la emulación de LAN. Por lo tanto los switches además de hardware para rutear celdas, deben poseer microprocesadores veloces para poder satisfacer éstas otras necesidades.

Las conexiones entre nodos ATM se realizan en base a dos interfaces diferentes:

- User to Network Interface o UNI que define la comunicación entre las estaciones finales (workstations y routers) y switches ATM en redes ATM privadas.
- Network to Network Interface o NNI que define la comunicación entre dos switches.

3. Tipos de Circuitos

ATM provee servicios orientados a conexión. Para comunicarse con un nodo remoto, un host debe solicitar a su switch local el establecimiento de una conexión con el destino. Una vez efectuada dicha conexión queda establecido un circuito entre los hosts. Existen dos tipos de circuitos conocidos como "circuitos virtuales": los conmutados, llamados SVC (Switched Virtual Circuit) y los permanentes, llamados PVC (Permanent Vitual Circuit).

- *Switched Virtual Circuit (SVC)*: un SVC opera del mismo modo que una llamada telefónica convencional. Un host se comunica con el switch ATM local y requiere del mismo el establecimiento de un SVC. El host especifica la dirección completa del nodo destino y la calidad del servicio requerido. Luego espera que la red ATM establezca el circuito. El sistema de señalización de ATM se encarga de encontrar el path necesario desde el host origen al host destino a lo largo de varios switches. El host remoto debe aceptar el establecimiento de la conexión. Durante el proceso de señalización de ATM cada uno de los switches examina el tipo de

servicio solicitado por el host origen y responde si puede satisfacer los requerimientos del pedido, si todos los switches lo satisfacen, la comunicación se establece; sino se negocia hasta encontrar un punto de acuerdo, es decir, el host origen acomoda sus pretensiones de calidad de servicio a lo que la red puede ofrecer en ese momento. Se genera entonces un “contrato” de calidad de servicio, el cual debe ser respetado por ambas partes (el host origen y la red); la red, por ejemplo, manteniendo los parámetros acordados; la estación, cumpliendo con su parte, es decir, no transmitiendo a mayor velocidad que la velocidad tope acordada. Cuando el proceso de señalización concluye el switch local reporta la existencia del SVC, al host originador de la llamada y al host remoto. La interface UNI identifica a cada uno de los SVC por medio de un número de 24 bits. Cuando un host acepta un nuevo SVC, el switch ATM local asigna al mismo un nuevo identificador. Los paquetes transmitidos por la red no llevan información sobre los nodos participantes de la comunicación. El host origen marca cada paquete enviado con el identificador del circuito virtual que lo llevará al nodo destino.

- *Permanent Virtual Circuit (PVC)*: La alternativa al mecanismo de SVC es la de los circuitos permanentes. Estos se definen y se mantienen permanentemente; no es necesario establecerlos dinámicamente. El administrador de la red es el encargado de configurarlos en forma manual. El administrador identifica el Host origen, el Host destino, la calidad de servicio y los identificadores de 24 bits para que cada Host pueda acceder al circuito. Podrían verse como las líneas punto a punto de telefonía.

Una ventaja de las Redes ATM es que pueden mantener varios circuitos virtuales, ya sea permanentes, conmutados o ambos, a través de un único enlace físico. En estas Redes se pueden agrupar varios circuitos virtuales en los llamados “caminos virtuales” (virtual paths). La ventaja de ello es que los switches pueden conmutar por camino virtual en lugar de por circuito. A esta facilidad se la denomina “virtual path switching”. Otra ventaja de utilizar caminos virtuales, es que es más sencillo redireccionar un camino virtual que varios circuitos virtuales en caso de fallar algún nodo de la Red.

4. Paths, Circuitos e Identificadores

ATM asigna un número entero único como identificador para cada conexión abierta por un host. El identificador solo es válido mientras que el circuito permanece abierto. Esta asignación es dinámica.

El identificador es válido para un solo sentido del circuito, esto quiere decir que los identificadores de circuito obtenidos por los host en los extremos del mismo usualmente son diferentes.

Los identificadores usados por la interface UNI están compuestos por 24 bits, divididos en dos campos: el primero de 8 bits y el segundo de 16 bits. Los primeros 8 forman el llamado VPI (Virtual Path Identifier), y los

restantes 16 definen el VCI (Virtual Circuit Identifier). Este conjunto de bits suele recibir el nombre "VPI/VCI pair".

Esta división del identificador en dos campos se hace para que el primer campo identifique la red y el segundo campo identifique el host. Si un conjunto de VCs sigue el mismo path puede asignar a todos ellos un mismo VPI. El hardware de ATM usa entonces los VPI para funciones de ruteo de tráfico.

En el caso de la interface NNI el VPI tiene 12 bits en lugar de 8.

5. Celdas

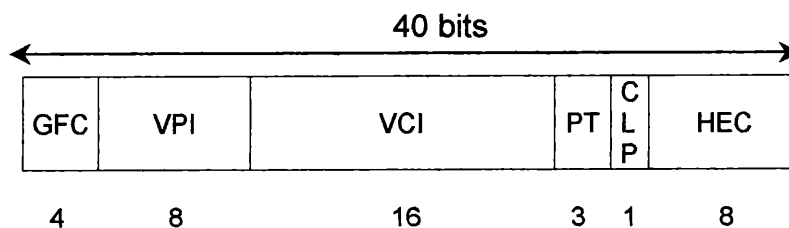
En cuanto al transporte de información, ATM usa tramas de tamaño fijo que reciben el nombre de celdas. El hecho de que todas las celdas sean del mismo tamaño permite construir equipos de "switching" de muy alta velocidad. Cada celda de ATM tiene una longitud fija de 53 octetos, de los cuales 5 pertenecen al encabezado y los restantes 48 a datos.

Dentro del encabezado se coloca el par VPI/VCI que identifica al circuito entre extremos, información de control y un CRC.

La conexión final entre dos nodos recibe el nombre de Virtual Channel Connection o VCC. Una VCC se encuentra formada por un conjunto de pares VPI/VCI.

Existen dos encabezados dependiendo de la interface:

- Formato de encabezado UNI para la comunicación entre los host y los switches ATM:



GFC: *Generic flow control*, se puede usar para proveer funciones locales. Prácticamente no se utiliza.

VPI: Se utiliza para identificar, conjuntamente con el VCI, el próximo destino de una celda.

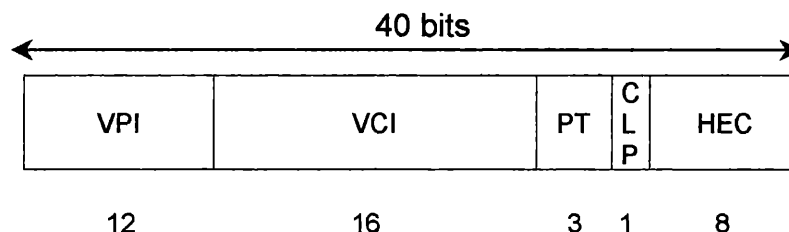
VCI: Es utilizado para identificar, conjuntamente con el VPI, el próximo destino de una celda.

PT: *Payload Type*, el primer bit indica si la celda contiene datos de usuario o de control. Si la celda contiene datos de usuario, entonces el segundo bit indica congestión, y el tercer bit indica cuando una celda es la última de un frame AAL5.

CLP: *Congestion Loss Priority*, indica si la celda puede ser descartada en caso de que la red este muy congestionada.

HEC: *Header Error Control*, checksum del encabezado solamente.

- Formato de encabezado NNI para la comunicación entre switches ATM:



La única diferencia que hay con el encabezado anterior, es que el campo de GFC no existe y el de VPI es 4 bits mayor.

6. Modelo en Capas

El modelo en capas se divide en tres grandes niveles: El Nivel Físico, el Nivel ATM y el Nivel de Adaptación de ATM (AAL).

El nivel ATM y el AAL se pueden ver como si fuera el nivel 2 del capa del modelo OSI (Data Link Layer), el nivel físico de ATM es semejante al nivel físico del modelo OSI.

Nivel Físico: Es independiente de los medios físicos propiamente dichos. Controla la transmisión y recepción de los bits; así como la relación entre las propiedades de las celdas y el tipo de frame utilizado dependiendo del medio físico.

Este nivel esta dividido en dos sub-niveles:

- **Medio Físico (PM):** Es responsable de mandar y recibir un flujo continuo de bits con la correspondiente información de sincronización. Como solo incluye funciones que dependen del medio físico utilizado, entonces su especificación depende del mismo. Los medios que se pueden utilizar son aquellos que sean capaces de transmitir celdas ATM. (Todos los estándares de hoy están relacionados con la fibra óptica; se están estudiando algunos sobre par trenzado).
- **Convergencia de Transmisión (TC):** Este subnivel es responsable de:
 - Mantener los límites de las celdas.
 - Generación y verificación del header error control.
 - Insertar o borrar celdas vacías para adaptar la capacidad de transmisión de datos del sistema.
 - Empaquetar las celdas en frames que se correspondan con el medio utilizado.

- Generar y mantener la estructura del frame de nivel físico.

Nivel ATM: Se encarga de establecer las conexiones y pasar las celdas por la red ATM. Para hacer esto, se usa la información contenida en el encabezado de cada celda ATM. El mecanismo de transporte es uno solo para múltiples opciones de servicio. Es independiente del tipo de información que es transmitida (datos, gráficos, voz, audio, video) con excepción del tipo de servicio (QOS) requerido.

Nivel AAL (ATM Adaptation Layer): Permite a la capa ATM transportar diferentes protocolos y servicios de capas superiores, traduciendo los frames de estas capas en unidades (48 octetos) que conforman la parte de datos de las celdas ATM.

Este nivel esta dividido en dos subcapas:

- CS (Convergence sublayer).
- SAR (Segmentation and Reassembly sublayer).

La capa CS recibe los paquetes y los divide en bloques de igual longitud, denominados CS PDU, donde PDU significa Protocol Data Unit (Unidad de Datos del Protocolo). Al CS PDU se le agrega un encabezamiento y un final con su información (tamaño, tipo, corrección de errores) y se lo pasa a la capa SAR. Esta, a su vez, divide al CS PDU en bloques pequeños de 44 bytes de largo. Se le agregan dos bytes de encabezamiento y dos de final y se forma el SAR PDU. Por último a éste se le agrega un encabezamiento y se forma la celda ATM.

Si bien ATM se maneja con celdas a nivel de capas inferiores, las aplicaciones que generan la información a ser transportada por ATM no trabajan con celdas. Estas aplicaciones interactúan con ATM por medio del Nivel AAL. Existen diferentes especificaciones de AAL dependiendo del tipo de transmisión que se necesite. En el momento de establecer la conexión el Host debe especificar el protocolo de Nivel AAL que se va a usar. Ambos extremos de la conexión deben acordar en el uso del mismo protocolo y éste no puede ser modificado durante la vida de la conexión.

En la siguiente tabla se muestran las principales características de los diferentes AAL.

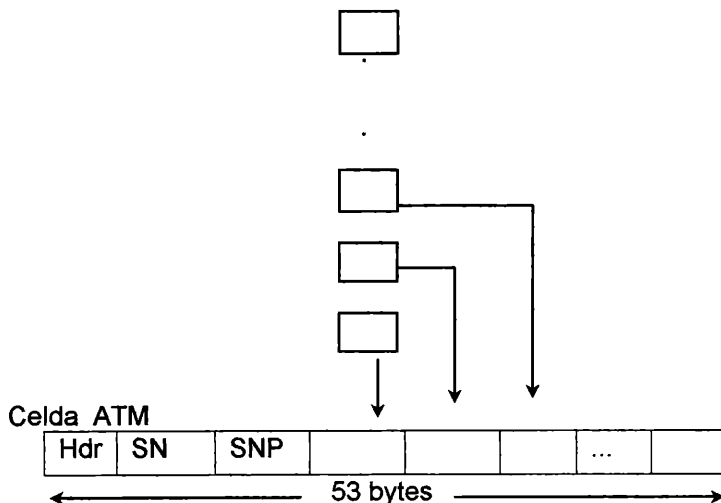
Características	AAL1	AAL3/4	AAL4	AAL5
Requiere sincronismo	Si	No	No	No
Tasa de Transferencia	Constante	Variable	Variable	Variable
Modo de Conexión	Orientado a Conexión	Orientado a Conexión	Sin Conexión	Orientado a Conexión
Tipos de Tráficos	Voz y Emulación de circuitos	Datos	Datos	Datos

AAL1:

AAL1 es apropiado para transmitir voz y video sin comprimir. Como requiere sincronización entre el origen y el destino depende de un medio que soporte clocking, como es SONET.

Este nivel asemeja la Red ATM a un canal de datos constante o una línea dedicada. Se utiliza para transmitir circuitos de velocidades constantes.

Armado de celda



El campo SN (Sequence Number) y el campo SNP (Sequence Number Protection) proveen la información necesaria para que el receptor verifique si esta recibiendo las celdas en correcto orden.

AAL3/4:

Es usada para transmitir paquetes SMDS (Switched Multimegabit Data Service) sobre una Red ATM.

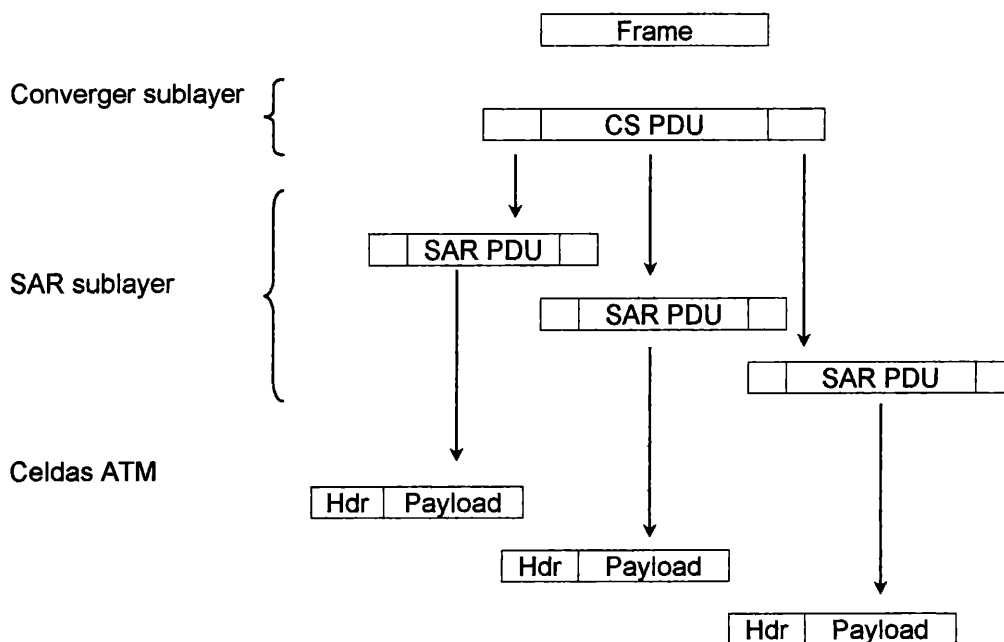
El nivel de Convergencia (CS) crea un PDU (Protocol Data Unit) a partir del Frame, anteponiendo un encabezado y agregando al final un campo de longitud.

El nivel de segmentación y reensamblado (SAR) fragmenta el CS PDU y le antepone a cada fragmento un encabezado que consiste de:

- *Tipo*: identifica cuando una celda es el comienzo, continuación o final de mensaje.
- *Número de secuencia*: indica el orden en el cuál las celdas deben ser reensambladas.
- *Identificador de multiplexado*: identifica celdas de orígenes distintos mezcladas en un mismo circuito virtual de conexión (VCC) para ser reensambladas correctamente en el destino.

Este nivel también agrega un CRC-10 a cada fragmento del CS PDU. El SAR PDU completo pasará a ser el payload de la celda ATM.

A continuación se muestra gráficamente dicho proceso:

Armado de celda

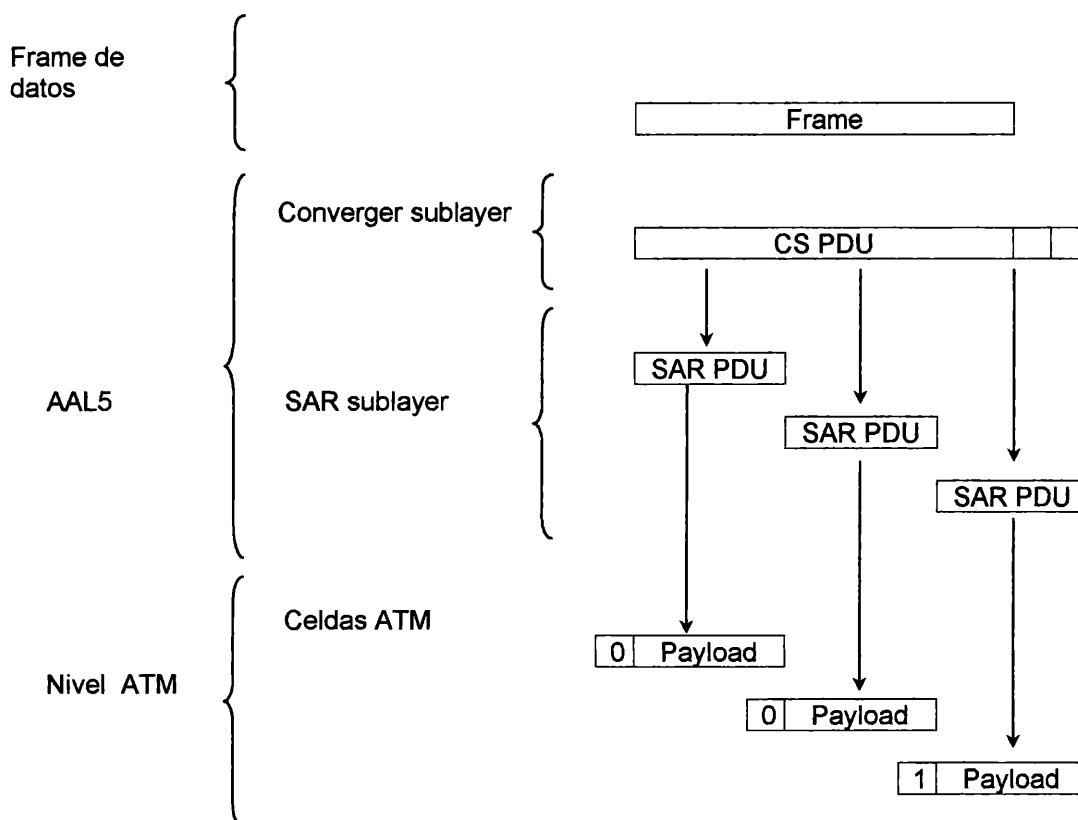
AAL5:

El nivel de Convergencia le agrega al Frame un relleno de longitud variable y 8 octetos de trailers. El relleno es de longitud variable para hacer coincidir la longitud del CS PDU con un número múltiplo de 48. El trailer incluye la longitud del Frame y un CRC de 32 bits calculado sobre todo el CS PDU. A continuación el nivel SAR segmenta el CS PDU en bloques de 48 octetos para que el nivel ATM introduzca cada bloque como porción de datos de una celda. En cada celda, excepto la última, uno de los bits del campo PT es seteado en 0 y en la última celda ese bit es seteado en 1, indicando que la misma es la última de la serie de celdas que componen el Frame. Cuando la celda arriba a su destino, el nivel ATM extrae el campo Payload de la celda; el subnivel SAR reensambla el CS PDU y el CS usa el CRC y la longitud del frame para verificar que el mismo ha sido transmitido y reensamblado correctamente.

Este nivel de adaptación es usado para transmitir la mayoría de los no-SMDS como Classical IP sobre ATM y emulación de LAN.

A continuación se muestra gráficamente como trabaja el AAL5:

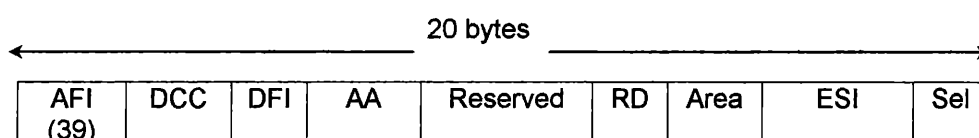
Armado de celda:



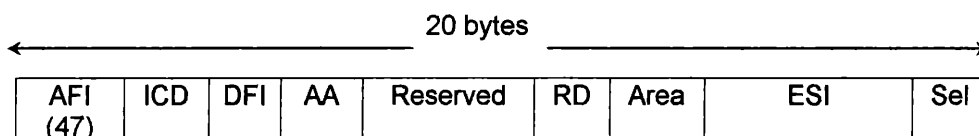
7. Direccionamiento

El ATM Forum adoptó el modelo de direccionamiento de subred en el cual el Nivel ATM es responsable de mapear las direcciones de nivel de red en direcciones ATM. Distintos formatos de dirección han sido desarrollados, uno para redes públicas y tres para redes privadas ATM. Típicamente las redes ATM públicas utilizan números del tipo E.164. Para las redes ATM privadas se desarrollaron tres formatos. Estas tres formas son Data Country Code (DCC), Internacional Code Designator (ICD), y Network Service Access Point (NSAP) encapsulando direcciones E.164.

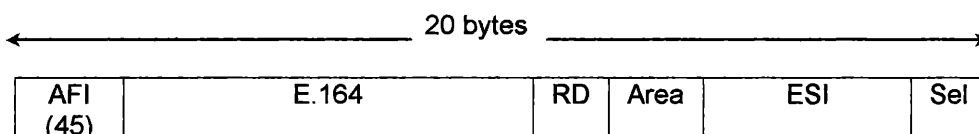
Formato de Dirección DCC:



Formato de Dirección ICD:



Formato de Dirección E.164 :



Los campos se describen a continuación:

- AFI: *Authority and Format Identifier*. 1 octeto. Identifica el tipo de dirección. Se definieron los valores 45, 47 y 39 para las direcciones E.164, ICD y DCC respectivamente.
- DCC: *Data Country Code*. 2 octetos.
- DFI: *Domain Specific Part (DSP) Format Identifier*. 1 octeto.
- AA: *Administrative Authority*. 3 octetos.
- RD: *Routing Domain*. 2 octetos.
- Area: *Area Identifier*. 2 octetos.
- ESI: *End System Identifier*. 6 octetos. Es una MAC Address.
- Sel: *NSAP Selector*. 1 octetos.
- ICD: *Internacional Code Designator*. 2 octetos.
- E.164: Número telefónico de *Integrated Services Digital Network (ISDN)*. 8 octetos.

8. ATM Switching

Los switches ATM usan los campos VPI/VCI del encabezado de la celda para identificar el próximo segmento de red que la celda necesita transitar para llegar al destino.

Un canal virtual es equivalente a un circuito virtual, es decir, ambos términos describen una conexión lógica entre host finales de comunicación. Un virtual path es un agrupamiento lógico de circuitos virtuales que le permiten a un switch ATM ejecutar operaciones sobre grupos de ellos.

La función principal del switch ATM es recibir celdas en un port y cambiarlos al port indicado según los valores VPI y VCI de la celda. Este cambio es hecho basándose en la tabla de switcheo que mapea un valor de entrada en la tabla en uno de salida, correspondiente al port de salida, según lo indican los valores VPI/VCI de la celda.

El siguiente es un ejemplo de una tabla de switcheo:

Entrada			Salida		
Port	VPI	VCI	Port	VPI	VCI
1	1	8	2	4	5
2	4	5	1	1	8
1	6	4	3	2	9
3	2	9	1	6	4
...

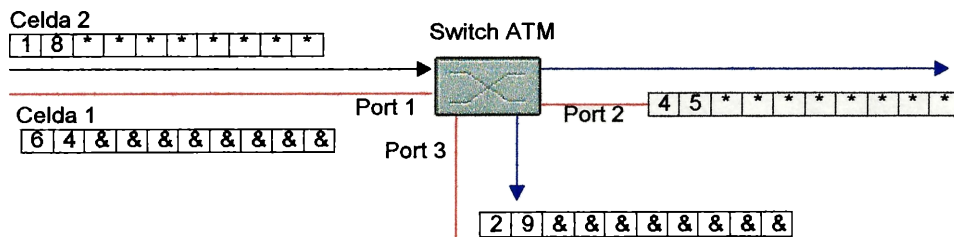


Figura 1.1

En el ejemplo mostrado en la Figura 1.1 llegan dos celdas al switch ATM por el Port 1. Primero, el switch examina los campos VPI y VCI de la celda 1 y encuentra que los campos tienen los valores 6 y 4 respectivamente. El switch examina en su tabla de switcheo para determinar por cual port deberá enviar la celda. Encuentra que cuando recibe una celda por el Port 1 con un VPI de 6 y un VCI de 4 deberá enviar dicha celda por el Port 3 con un VPI de 2 y un VCI de 9. Entonces, para la celda 1, el switch cambia el VPI por 2 y el VCI por 9 y envía la celda por el Port 3.

Luego, el switch examina la celda 2, ésta tiene un VPI de 1 y un VCI de 8. Encuentra en la tabla que lo llegado por el Port 1 con un VPI de 1 y un VCI de 8 debe enviarse por el Port 2 con un VPI de 4 y un VCI de 5. Entonces produce los cambios y envía la celda.

A la inversa, cuando arriban celdas por el Port 3 con un VPI de 2 y un VCI de 9, al consultar la tabla se obtendrá que a esa celda se le cambiarán estos valores por 6 y 4 respectivamente y se la enviará por el Port 1. También cuando llegue una celda por el Port 2 con los valores de VPI y

VCI de 4 y 5 respectivamente, a través de la tabla de switcheo se obtendrá que se deberá enviar esa celda por el Port 1 cambiando los valores de VPI y de VCI por 1 y 8 respectivamente. Hay que notar que los valores de VPI y de VCI son significativos solo en la interface local.

El campo VPI es usado para agrupar circuitos virtuales en circuitos lógicos. De esta manera se reduce el número de campos ha ser cambiados por cada celda que pasa por el switch, así la performance del switch aumenta.

A continuación se muestra un ejemplo gráfico de cómo agrupar circuitos virtuales en circuitos lógicos:

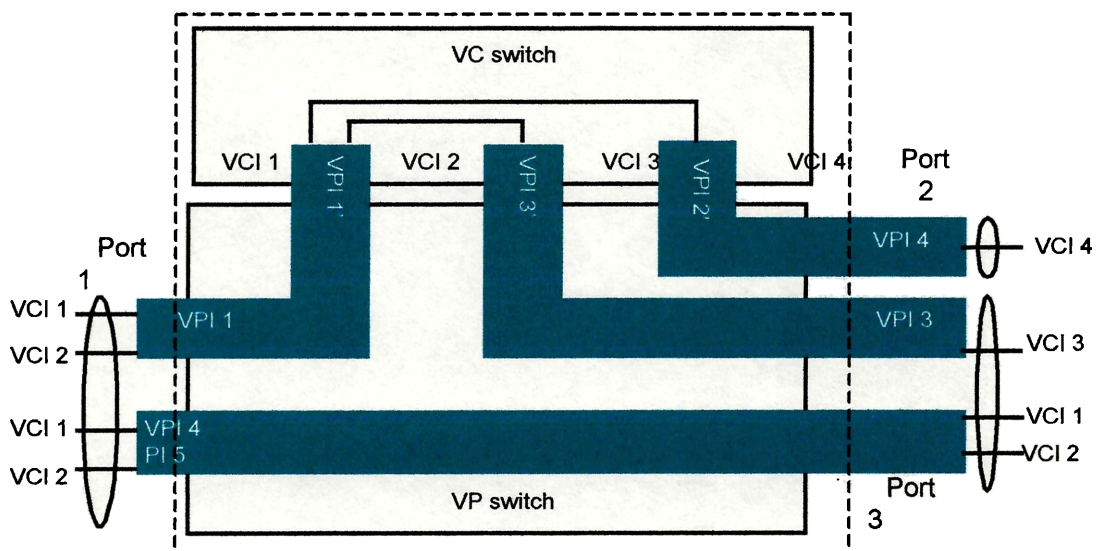


Figura 1.2

En la Figura 1.2 se ve que para las celdas que entran al switch ATM por el Port 1 y tienen un VPI con valor 4 son procesadas a través del "VP switch", el cuál cambia el valor del VPI por 5 en cada celda, mantiene intacto el valor del VCI, y envía la celda por el Port 3. Las celdas que tienen el VPI con valor 1 son procesadas por el "VC switch". Por cada celda que tiene un VCI con valor 1, el "VC switch" cambia el VPI a 4 y el VCI a 4 y la envía por el Port 2. Por cada celda con el VCI con valor 2, el "VC switch" cambia el VPI por 3 y el VPI por 3 y la manda por el Port 3.

9. Tipos de conexión

ATM soporta dos tipos de conexión:

- Punto a Punto. Estas conexiones pueden ser unidireccionales o bidireccionales.
- Punto a Multipunto. Este tipo de conexión es unidireccional únicamente.

Las siguientes figuras muestran los dos tipos de conexiones existentes en las tecnologías ATM:

Conexión Punto a Punto

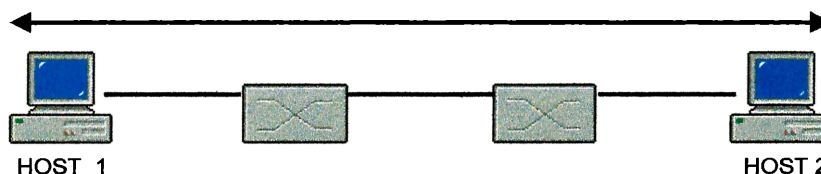


Figura 1.3

Conexión Punto a Multipunto

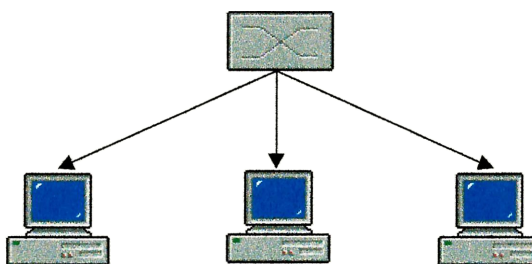


Figura 1.4

Sería deseable que ATM soportara comunicación Multipunto a Multipunto, el equivalente a Broadcast. Desafortunadamente, el estándar de AAL5 no provee una forma para que el receptor identifique celdas individuales de diferentes orígenes. Esto no permite un apropiado reensamble de las celdas llegadas en los frames. Una solución a este problema es utilizar un servidor de Multicast.

Un servidor de Multicast puede existir en una red ATM y todos los miembros de un grupo multicast pueden establecer una comunicación Punto a Punto con él. El servidor de multicast crea, entonces, una comunicación Punto a Multipunto con todos los miembros del grupo.

La Figura 1.5 muestra la comunicación entre un servidor Multicast y un grupo de hosts.

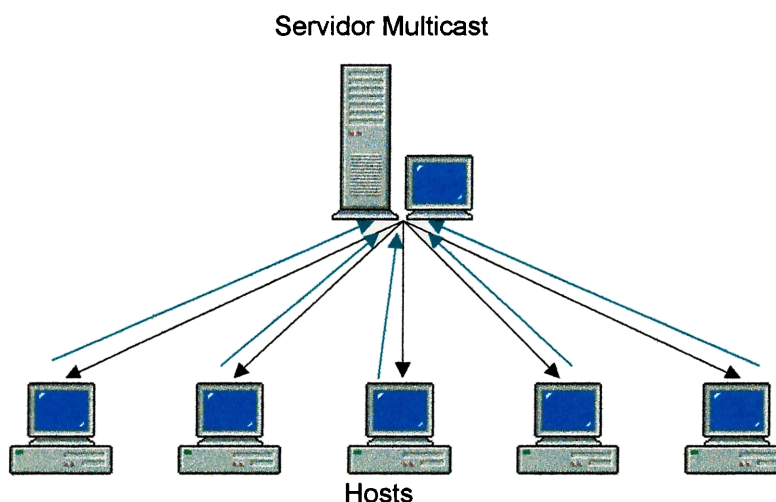


Figura 1.5

Los datos enviados por el server de multicast son serializados. Los servidores de multicast pueden soportar grupos dinámicos debido a que los miembros pueden ser borrados y agregados al árbol.

10. Señalización

Cuando un dispositivo ATM quiere establecer una comunicación con otro dispositivo ATM envía un paquete de pedido de conexión al switch ATM al cuál esta conectado. El pedido contiene la dirección ATM del destino y los parámetros de Calidad del Servicio (QOS) que requiere la conexión.

Este paquete es reensamblado y examinado por cada uno de los switch intermedios. Si alguno de estos dispositivos no puede cumplir con los parámetros QOS, el pedido es descartado, y se envía un mensaje al origen avisándole de dicho descarte. Por el contrario, si el paquete llego al destino final y éste acepta los parámetros QOS, se devuelve un mensaje de aceptación que se propaga hasta el origen, estableciendo de esta manera el circuito. También se recibe en la aceptación los valores del par VPI/VCI con los cuales el origen identificará al destino pedido.

La Figura 1.6 siguiente muestra gráficamente el pedido de conexión de un host:

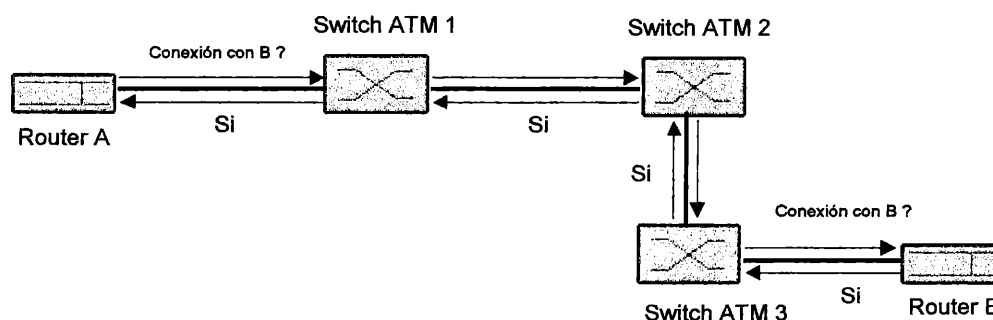


Figura 1.6

En el ejemplo, se ve como el Router A hace el pedido de conexión al switch al cuál está directamente conectado (Switch ATM 1), este evalúa los parámetros de la calidad del servicio del pedido y como puede cumplir con dichos parámetros pasa el pedido al Switch ATM 2, y así sucesivamente hasta llegar al Router B que es el destino final. Este también evalúa los parámetros y acepta el pedido. Entonces éste manda un mensaje de aceptación que llega hasta el Router A, que fue el originador del pedido.

11. Calidad de Servicio y Manejo de Tráfico

ATM es una tecnología desarrollada para soportar una gran variedad de servicios y aplicaciones. El Control del Tráfico está fundamentalmente relacionado con la habilidad de la red de proveer apropiadamente los QOS (Quality of Service o Calidad del Servicio).

Un rol primario en el manejo del tráfico es proteger a la red y a los usuarios finales de la congestión y así asegurar los objetivos de la performance de la red. Un rol adicional es el de promover el uso eficiente de los recursos de la red.

Se podrían definir cinco categorías para los servicios. En cada una de ellas se da un conjunto de parámetros que describen el tráfico presentado a la red y el QOS que se requiere de la misma. Un número de mecanismos de control de tráfico son definidos para que la red alcance los QOS requeridos.

11.1. Funciones Genéricas

Las siguientes funciones son utilizadas para alcanzar los objetivos propuestos anteriormente:

- **Connection Admission Control (CAC):**

Es definida como un conjunto de acciones tomadas por la red durante el Setup de la conexión para determinar cuando el

request de conexión puede ser aceptado y cuando puede ser rechazado (o cuando un request para realocación puede ser acomodado).

- **Feedback Controls:**

Son un conjunto de acciones tomadas por la red y los host para regular el tráfico presentado a las conexiones ATM de acuerdo a los estados de los elementos de la red.

- **Usage Parameter Control (UPC):**

La red tiene definido un conjunto de acciones para monitorear y controlar el tráfico, en términos de tráfico ofrecido y validez de la conexión ATM, en el acceso de los Host.

El propósito principal es proteger los recursos de la red de las malas conductas ya sean intencionadas o no, que pueden afectar los QOS de las conexiones ya establecidas, detectando violaciones de los parámetros negociados y tomando las acciones correspondiente. Tales acciones incluyen descartar y etiquetado de celdas.

- **Cell Lost Priority Control:**

Para varias categorías de servicio los Host deben generar flujo de tráfico con el bit CLP marcado. La red puede seguir modelos que traten a esta marca en forma transparente o dándole significado, en cuyo caso la red descarta celdas marcadas para proteger tanto como sea posible los objetivos de los QOS de las celdas con mayor prioridad.

- **Traffic Shapping:**

Los mecanismos de Traffic Shapping deben ser usados para llevar a cabo una modificación deseada en las características del tráfico.

- **Network Resource Management (NRM):**

La arquitectura de servicio permite separación lógica de las conexiones de acuerdo a las características del servicio. Si bien el scheduling de celdas y el provisionamiento de recursos son implementaciones específicas de la red, estas pueden ser utilizadas para proveer una apropiada separación y accesos a los recursos. Virtual Path es una herramienta útil para el manejo de recursos.

- **Frame Discard:**

Una red congestionada que necesita descartar celdas puede descartar al nivel de frame en lugar de a nivel celdas.

- **ABR Flow Control:**

El protocolo ABR para Control de Flujo puede usarse para compartir el ancho de banda disponible entre los usuarios participantes.

11.2. Arquitectura de Servicio ATM

Las categorías de servicio provistas a nivel ATM son:

- CBR : Constant Bit Rate
- rt-VBR: Real Time - Variable Bit Rate
- nrt-VBR: Non Real Time – Variable Bit Rate
- UBR: Unspecified Bit Rate
- ABR: Available Bit Rate

Estas categorías de servicio relacionan las características del tráfico y los requerimientos de los QOS con el comportamiento de la red.

Funciones con Ruteo, CAC y asignación de recursos, en general, son estructurados diferentemente para cada categoría de servicio. Las categorías de servicio son distinguidas en principio como Real Time o Non Real Time. Para tráfico de tiempo real existen dos categorías CBR y rt-VBR distinguidas cuando el descriptor del tráfico contiene el parámetro Peak Cell Rate (PCR) o ambos parámetros, el PCR y el Sustainable Cell Rate (SCR).

Todas estas categorías se aplican tanto a los VCC como a los VPC.

11.3. Definiciones para las Categorías de Servicios:

Se definen las categorías de servicios ATM usando los siguientes parámetros QOS:

- Peak-to-Peak Cell Delay Variation (Peak-to-Peak CDV)
- Maximun Cell Transfer Delay (max CTD)
- Cell Loss Ratio (CLR)

Definición de la Categoría CBR:

Esta categoría es usada por conexiones que requieren un ancho de banda fijo continuamente disponible durante el tiempo de la conexión. Esta cantidad de ancho de banda está caracterizada por el valor Peak-to-Peak (PCR).

En esta categoría el origen puede emitir celdas de acuerdo al PCR en cualquier unidad de tiempo y por cualquier duración de tiempo.

Este tipo de servicio está pensado para soportar aplicaciones de tiempo real, como por ejemplo voz, video, pero no es restrictiva a las mismas.

El origen no esta obligado a utilizar constantemente el PCR para la transmisión.

Las celdas que son retrasadas más allá del valor especificado en el max CTD pierden sentido para la aplicación. Esta categoría de servicio puede ser usada tanto para los VCC como para los VPC.

Definición de las Categoría rt-VBR:

Al igual que la categoría anterior, esta categoría de servicio también esta orientada a aplicaciones de tiempo real, las conexiones de este tipo se caracterizan en términos de un PCR, un Sustainable Cell Rate y un Maximum Burst Size (MBS). Los orígenes estarán esperando para transmitir a una tasa que varía con el tiempo.

De la misma manera que en la categoría anterior, las celdas que son retrasadas más allá del max CTD pierden sentido para la aplicación.

Los servicios de esta categoría pueden soportar multiplexado estático de orígenes de tiempo real.

Definición de la Categoría ntr-VBR:

Esta categoría de servicio esta orientada para aplicaciones que no trabajan en tiempo real las cuales tienen características de tráfico repentino y están caracterizadas en términos de un PCR, SCR y MBS. Para aquellas celdas que son transferidas dentro del contrato del tráfico, las aplicaciones esperan una tasa baja de pérdida de celdas.

Los servicios Non Real Time VBR pueden soportar multiplexado estático de conexiones. Esta categoría no tiene límites asociados para los retrasos.

Definición de la Categoría UBR:

Esta orientada a aplicaciones que no trabajan en tiempo real, por ejemplo aquellas que no requieren un control sobre los retrasos ni cuentan con retrasos variables (e-mail).

UBR no especifica garantías. No se ha enumerado que pasa con respecto al CLR y al CTD en este tipo de conexiones.

Una red puede o no aplicar PCR a las funciones CAC y UPC. En el caso donde la red no fuerza el uso del PCR, el mismo es solo de información. Cuando no se fuerza el uso de PCR, éste puede ser útil para su negociación. A partir de esta negociación el origen puede descubrir cuál es el ancho de banda del camino de la conexión.

El control de congestión para UBR puede ser utilizado a un nivel más alto.

Definición de la Categoría ABR:

ABR es una categoría de servicio del nivel ATM para la cuál los límites de transferencia provistos por la red pueden cambiar luego de establecida la conexión.

Se especifica un mecanismo de control de flujo que soporta varios tipos de feedback para controlar el rate del origen en respuesta para cambiar las características de la transmisión.

Este feedback es llevado al origen a través de celdas específicas llamadas Celdas Manejadoras de Recursos (Resource Management Cells) o celdas RM. Se espera que un Host que adapta

su tráfico de acuerdo con el feedback tenga una baja tasa de pérdidas de celdas y comparta su problema con el ancho de banda disponible de acuerdo a una disposición de asignación específica de la red.

Este servicio no requiere limitar el retraso (delay) y no está orientado para trabajar con aplicaciones de tiempo real.

Cuando se establece una conexión ABR el Host debe especificar a la red el máximo y el mínimo ancho de banda que quiere utilizar. Estos valores deben ser asignados como PCR y MCR. El MCR puede ser 0 (cero); el ancho de banda disponible puede variar pero nunca puede ser menor al MCR.

11.4. Parámetros y Atributos de las Categorías de Servicios

Atributo	Categoría de Servicio del Nivel ATM				
	CBR	nt-VBR	nrt-VBR	UBR	ABR
Parámetros de Tráfico					
PCR y CDVT(4,5)	especificado			especific2	especific3
SCR, MBS, CDVT(4,5)	n/a	especificado		n/a	
MCR	n/a			n/a	especific.
Parámetros QoS					
Peak-to-Peak CDV	especificado		sin especificar		
Max CTD	especificado		sin especificar		
CLR4	especificado			sin especific.	Ver Nota 1
Otros Atributos					
Feedback	sin especificar				especific.

Notas:

1. El CLR es inferior para orígenes que adjuntan el flujo de celdas para el control de información.
2. Puede no estar propuesto para los procedimientos CAC y PCR.
3. Representa la máxima velocidad con la cuál el origen ABR puede enviar siempre. La velocidad actual es propuesta por el control de información.
4. Estos parámetros están especificados en forma explícita o implícita por los PVCs o los SVCs.
5. CVDT se refiere a Cell Delay Variation Tolerance. En general no tiene un único valor para una conexión. Diferentes valores pueden aplicarse a cada interface a lo largo del camino de una conexión.

12. ATM Layer QOS

Este nivel está dado por un conjunto de parámetros que caracterizan una conexión a nivel ATM. Estos parámetros cuantifican la performance de la red Punto a Punto a nivel ATM.

12.1 Parámetros de la Calidad del Servicio

Se identifican seis parámetros los cuales corresponden a un objetivo de performance en la red, tres de estos pueden ser negociados entre los host y la red.

Los siguientes parámetros de QOS son negociados:

- Peak-to-Peak Cell Delay Variation (Peak-to-Peak CDV).
- Maximun Cell Transfer Delay (max CTD).
- Cell Loss Ratio (CLR).

Los siguientes parámetros no son negociables:

- Cell Error Ratio (CER).
- Severely Errored Cell Block Ratio (SECBR).
- Cell Misinsertion Rate (CMR).

12.2. Naturaleza de los QOS acordados

Una red puede soportar uno o más objetivos de performance para cada parámetro de QOS. Para cada conexión de una dirección, un QOS específico es negociado entre la red/es y los Host. La red acuerda alcanzar o exceder los QOS negociados tanto como los Host puedan cumplir con el contrato de tráfico.

Los QOS tienen la intención de ser la primera aproximación de la performance que la red puede ofrecer.

La precisión con la cuál los varios valores de QOS pueden ser codificados pueden ser significativamente grandes con respecto a la exactitud con la cuál la red los puede predecir, medir o mantener en un nivel de performance dado.

13. Contrato de Tráfico

13.1. Parámetros y descriptores del Tráfico

Los Parámetros de Tráfico describen las características de tráfico de un origen. Para una conexión dada estos parámetros son agrupados dentro de un descriptor de tráfico del origen.

13.2 Parámetros de Tráfico

Un parámetro de Tráfico describe una característica inherente al origen del tráfico, ésta puede ser cuantitativa o cualitativa, por ejemplo PCR, SCR, MBS y MCR.

13.3. Descriptores del origen del Tráfico

Un descriptor es un conjunto de parámetros de tráfico de un origen ATM. Este es usado durante el establecimiento de la conexión para capturar las características intrínsecas del tráfico de una conexión de un origen particular.

13.4. Descriptores de la conexión del Tráfico

Este descriptor especifica las características del tráfico de una conexión ATM. Incluye el descriptor del origen del tráfico, el CDVT y la definición de conformidad que es usada para especificar en forma no ambigua la celda que conforman la conexión.

Los procedimientos de CAC usan este descriptor para alocar recursos y para derivar valores de los parámetros para la operación de UPC. Este descriptor contiene información necesaria para conformar un testeo de celda de la conexión a nivel UNI. Estos parámetros de tráfico deberían cumplir los siguientes requerimientos:

- Que sean comprendidos por los Host.
- Ser útiles en los esquemas de asignación de recursos, alcanzando los requerimientos de performance de la red.
- Tener la capacidad de ser forzados por el UPC.

13.5. Especificación del Contrato de Tráfico

Un Contrato de Tráfico especifica las características negociadas de una conexión.

El Contrato del Tráfico en una UNI pública consistirá de un descriptor de conexión de tráfico y un conjunto de parámetros QOS para cada dirección de la conexión, e incluirá la definición de una conexión simple.

La UNI privada puede opcionalmente soportar el mismo Contrato de Tráfico que la UNI pública o uno diferente.

El descriptor de la conexión de tráfico consiste de todos los parámetros usados para especificar en forma no ambigua la conformación de las celdas de conexión, por ejemplo:

- El descriptor del origen del tráfico (PCR, SCR, MBS y MCR).
- El CDVT.
- La definición de conformidad.

Para el CBR, rt-VBR, nrt-VBR y UBR se usa una especificación libre de ambigüedades para formar las celdas de una conexión a nivel UNI.

Para ABR la definición se refiere a la conducta especificada para los orígenes ABR, destinos y switches, pero permite delays entre el origen y la UNI que pueden perturbar el flujo del tráfico. Esta definición no se puede interpretar como un algoritmo de UPC.

Los valores de los parámetros del contrato de tráfico pueden ser especificados explícita o implícitamente. El valor del parámetro se especifica explícitamente cuando su valor es asignado por los Host usando señalamiento para SVCs o cuando éste es especificado por el NMS (Network Management Systems) para PVCs.

El valor de un parámetro especificado en tiempo de subscripción se considera especificado implícitamente cuando es asignado por la red usando las reglas default, las cuales dependen de la información especificadas explícitamente por los Host.

Los procedimientos CAC y UPC son específicos de una red y tienen que tener en cuenta el contrato específico para operar eficientemente.

Para poner información adicional, parámetros experimentales de tráfico a nivel UNI, los mensajes de señalización tienen la capacidad de codificarlos.

14. Interface UNI 4.0

Esta especificación provee los procedimientos de Signaling para establecer, mantener y limpiar dinámicamente las conexiones ATM en una interface UNI ATM. Los procedimientos están definidos en términos de los mensajes y los elementos de información usados para caracterizar la información.

14.1. Configuración de Referencia

El protocolo es válido para interfaces UNI privadas y públicas. El propósito de una configuración de referencia para la especificación de

signaling UNI es listar todos los elementos y links de una red ATM entre las cuales la especificación se aplica.

Los elementos de red en este contexto son:

- Equipos "endpoints"
- Redes ATM privadas
- Redes ATM públicas

Para los propósitos de esta especificación una red pública o privada consiste de uno o más sistemas de switching ATM bajo la misma administración.

Reference configurations

Between equipment	End point Network	Private ATM Network	Public ATM
End Point Equipment Private ATM Network Public ATM Network	Nota 1 Private UNI Public UNI	Private UNI Nota1 Public UNI	Public UNI Public UNI Nota1

Nota1: La entrada para esta conexión esta fuera del alcance de esta especificación.

14.2. Capacidades de la interface UNI 4.0

Nro.	Capacidad	Terminal equipment	Switching System
1	Point-to-Point calls	M	M
2	Point-to-Multipoint Calls	O	M
3	Signalling of Individual QoS Parameters	M	M
4	Leaf Initiated Join	O	O
5	ATM Anycast	O	Nota1
6	ABR Signalling for Point-to-Point Calls	O	O
7	Generic Identifier Transport	O	O
8	Virtual UNI	O	O
9	Switched Virtual Path (VP) service	O	O
10	Proxy Signalling	O	O
11	Frame Discard	O	O Nota2
12	Traffic Parameters Negotiation	O	O
13	Supplementary Service	O	O
13.1	Direct Dialing In (DDI)	-	-
13.2	Multiple Subscriber Number (MSN)	O	O
13.3	Calling Line Id. Presentation (CLIP)	O	O
13.4	Calling Line Id. Restriction (CLIR)	O	O
13.5	Connected Line Id. Presentation (COLP)	O	O
13.6	Connected Line Id. Restriction (COLP)	O	O
13.7	Subaddressing (SUB)	O	Nota3
13.8	User-User Signalling (UUS)	O	O

Nota1: Esta capacidad es opcional para sistemas de switching/redes públicas y es mandatorio para sistemas switching/redes privadas.

Nota2: El transporte de la indicación de Frame Discard es Mandatoria.

Nota3: Esta capacidad es Mandatoria para redes y sistemas de switching (públicas y privadas) que soportan sólo el formato de direcciones E.164.

14.3. Llamadas Punto a Punto



BIBLIOTECA
FAC. DE INFORMÁTICA
U.N.L.P.

14.3.1. Direccionamiento

Una dirección ATM puede ser una E.164 de 15 dígitos de longitud o una de 20 octetos basados en el formato de codificación ISO NSAP.

Redes Privadas

Una dirección de un sistema final ATM identifica uno o más endpoints ATM. El formato de una dirección ATM para "endpoints" en redes ATM privadas es modelado después del formato de la OSI Network Service Access Point como fue especificada en ISO 8348.

En redes ATM existen 2 tipos de direcciones: individuales y grupales; una dirección ATM individual identifica un solo ATM "endsystem" mientras que una dirección ATM de grupo identifica una o más ATM endsystems.

DCC ATM Format
ICD ATM Format
E.164 ATM Format

La habilidad de un "endpoint" para originar una llamada a otro "endpoint" debe ser independiente de la estructura de la dirección ATM del sistema llamado.

Todas las redes privadas deben ser capaces de aceptar mensajes de inicialización 'call setup' conteniendo direcciones ATM con cualquiera de los formatos IDI que se describen en este documento y seguir con la correspondiente llamada en sentido contrario hasta el "endpoint" destino, si éste es alcanzable.

Initial Domain Part (IDP): Initial Domain Part (IDP) especifica únicamente la autoridad administrativa que tiene la responsabilidad para el alocaimiento y asignación de DSP (Domain Specified Part). IDP consta de dos partes, la autoridad e identificador del formato (AFI) y el Initial Domain Identifier (IDI).

- *AFI*: identifica la autoridad alocadora del Data Country Code, International Code Designator, o el nuevo E.164; el formato del IDI y la sintaxis del resto de la dirección. La longitud de este campo es de 1 octeto.
- *Data Country Code (DCC)*: especifica el país en el cual esta registrada la dirección. Los códigos son dados en ISO 3166 (tiene una longitud de 2 octetos). Los dígitos del DCC están codificados en Binary Code Decimal.
- *International Code Designator (ICD)*: identifica una autoridad que administra un esquema de codificación. El cuerpo responsable para el esquema de la codificación identificada por el ICD, provee una autoridad administrativa

que es responsable de la asignación de identificadores dentro de este esquema de códigos para organizaciones. La designación de una autoridad para ICD es mantenida por el British Standard Institute (2 octetos de longitud). Los códigos son justificados a izquierda y rellenos a derecha con el valor Hexa F.

- *E.164*: especifica nuevos Servicios Integrados en Redes Digitales (Integrated Services Digital Network). Estos números incluyen números telefónicos. Puede ser usado el formato internacional de estos números. Estos pueden ser de hasta 15 dígitos de longitud. La longitud de este campo es de 8 octetos. Los dígitos de un número E.164 son codificados en BCD.

Domain Specific Part (DSP): Esta sección se subdivide en DSP de alto orden (HO-DSP) y DSP de bajo orden, la cual consiste de End System Identifier (ESI) y del Selector (SEL).

- *HO-DSP*: el código de este campo es especificado por la autoridad o por el esquema de codificación identificado por IDP. La autoridad determina cuantos identificadores deberán ser asignados e interpretados dentro del dominio. La autoridad puede crear subdominios. Esto es, la autoridad puede definir algún número de subcampos del HO-DSP y usarlos para identificar un nivel más bajo de autoridad, el cual a su turno, define el balance del HO-DSP. Los subcampos de HO-DSP de la izquierda son siempre más significativos que los de la derecha. Los contenidos de estos campos no solo describen la jerarquía de direccionamiento, sino que también tienen significado de topología. Esto es, HO-DSP podría ser construido de tal manera que el ruteo a través de redes ATM interconectadas sea facilitado.
- *End System Identifier (ESI)*: identifica un sistema final. Este identificador debe ser único dentro de un valor particular de IDP + HO-DSP. Además, para asegurar la habilidad de autoconfiguración de direcciones de un endsystem, este ESI puede ser un identificador unívoco con alcance global especificado por una dirección MAC IEEE (6 octetos).
- *Selector*: no es utilizado para ruteo ATM, pero podrá ser usado por los "endsystem".

Redes Públicas

La interface UNI pública soporta uno de los siguientes:

1. E.164
 - Type of Number Field = numero internacional

- Numbering Plan Identification field: recomendación E.164
2. Private ATM Address Structure (todos los formatos (3))
 - Type de Number: desconocido
 - Numbering Plan Indicator: ATM Endsystem Address

14.4. Leaf Initiated Join Capability (LIJ)

Esta sección especifica la capacidad de las hojas para conectarse a conexiones punto a multipunto con o sin intervención de la raíz .

Existen dos modos de operación asociados con la capacidad LIJ.

Join propuesto por la hoja sin notificación de la raíz:

Una hoja puede generar y enviar un Request sobre la UNI para “engancharse” a una conexión punto a multipunto. En este modo de operación, si el Request de la hoja es para una conexión existente, el Request es manejado por la red. La raíz no es notificada cuando cada hoja es agregada o borrada de una conexión (excepto cuando la raíz se agrega o se borra como hoja). Note que cuando una hoja realiza el Request para agregarse a una conexión que todavía no fue establecida, la raíz ejecutará el setup de la conexión. Este tipo de conexiones, las cuales son manejadas por la red, son denominadas “Network LIJ Connection”.

Join Propuesto por la Raíz

Una hoja puede generar y enviar un Request sobre la UNI para “engancharse” a una conexión punto a multipunto. En este modo de operación el Request es manejado por la raíz de la conexión. La raíz agrega y remueve hojas de la nueva conexión o de la ya establecida. Este tipo de conexión es conocida como “Root LIJ Connection”.

La capacidad LIJ requiere un nuevo estado de máquina. Este nuevo estado existe solo sobre el lado usuario de la UNI donde una hoja intenta agregarse o borrarse a una conexión; este lado usuario puede estar en uno de los siguientes dos estados en un momento dado:

Leaf Setup Initiated: se entra en este estado cuando el mensaje Leaf Setup Request ha sido enviado al lado de Red de la interface UNI.

Null: se entra en este estado cuando un mensaje Setup, Add Party o Leaf Setup Failure han sido recibidos por el lado usuario de la interface o cuando el timer T331 expira.

14.5. Requerimientos de códigos

Esta sección especifica los mensajes y los elementos de información requeridos para soportar la capacidad de LIJ.

14.5.1. Setup

Este mensaje es enviado por el llamador a la red y por la red al llamador para iniciar un establecimiento de la conexión.

Elementos de información para tener en cuenta:

- LIJ Call Identifier (Nota1)
- LIJ Parameters (Nota2)
- Leaf Sequence Number (Nota3)

Nota1: no incluido para llamada punto a punto. Incluido en el sentido usuario-red para requerir la capacidad LIJ. Esta información será usada por la red para soportar la capacidad LIJ. Incluido en el sentido red-usuario para interactuar con redes privadas o para proveer la opción de re-inclusión de la hoja.

Nota2: cuando esta presente, los elementos de información, identificador de llamada de LIJ y el "Calling Party Number" deben también estar presentes.

Nota3: debe ser incluida cuando se inicia una llamada a una hoja en respuesta al mensaje Leaf Setup Request. El valor debe ser igual al valor especificado por la hoja en el mensaje Leaf Setup Request de la hoja.

14.5.2. Modificación de los mensajes punto a multipunto

Add Party

Este mensaje es enviado por un usuario a la red para pedir la inclusión a una conexión existente.

Elementos de información a tener en cuenta:

- Leaf Sequence Number: debe ser incluido cuando se agrega una nueva parte a una llamada punto a multipunto en respuesta a un mensaje Leaf Setup Request. El valor debe ser igual al valor especificado por la hoja en el mensaje Leaf Setup Request.

14.5.3. Mensajes para llamadas de LIJ y Control de Conexión

Leaf Setup Failure

Este mensaje es enviado por la raíz o por la red a las hojas para indicar la falla del pedido de incorporación de una hoja.

Elementos de información a tener en cuenta:

Protocolo Discriminator
Call Reference (Nota1)
Message type
Message length
Cause
Called Party Number (Nota2, Nota3)

Called Party Subaddress
Leaf Sequence Number
Transit Network Selection

Nota1: debe ser enviado para el valor del "Dummy Call Reference"

Nota2: mandatario en el sentido usuario-red. Mandatario en el sentido red-usuario, si el Calling Party Number fue incluido en el mensaje Leaf Setup Request.

Nota3: la longitud depende del "numbering plan". Esta longitud máxima es de 25 octetos.

Nota4: mandatario si el mensaje Leaf Setup Request contiene un "Calling Party Subaddress", si no esta permitido.

Leaf Setup Request

Este mensaje es enviado para iniciar un procedimiento de "enganche" de hoja.

Elementos de información a tener en cuenta:

Protocol Discriminator
Call Reference (Nota1)
Message Type
Message Length
Transit Network Selection (Nota2)
Calling Party Number (Nota3)
Calling Party Subaddress
Called Party Number (Nota4)
LIJ Call Identifier
Leaf Sequence Number

Nota1: debe ser seteado para el valor del "Dummy Call Reference"

Nota2: usado para especificar el tránsito de red sobre el cual la raíz (Called Party) del LIJ Call debería ser alcanzado.

Nota3: la longitud mínima depende del "numbering plan" (máximo 26 octetos).

Nota4: idem Nota3 (longitud máxima 25 octetos).

14.5.4. Elementos de información

Leaf Initiated Join Call Identifier

Es usado para identificar unívocamente una llamada punto a multipunto en la interface raíz. El identificador de la llamada LIJ es especificado en el mensaje Setup cuando la raíz crea una llamada punto a multipunto con capacidades de LIJ. Este identificador es especificado en el mensaje Leaf Setup Request cuando una hoja quiere agregarse a la llamada identificada.

Valor Identificador: el valor identificador es representado por cuatro octetos, siendo usados para diferenciar las llamadas creadas por la misma raíz.

Leaf Initiated Join Parameters

Usados por la raíz para asociar opciones con las llamadas cuando éstas son creadas.

Leaf Sequence Number

Este elemento de información es usado por una hoja entrante para asociar los mensajes de respuesta con los correspondientes mensajes de Request.

14.6. Procedimientos de Signalling para Soportar la Capacidad LIJ**14.6.1. Procedimiento de incorporación en la interface de la hoja****Leaf Setup Request**

Cuando una hoja quiere incorporarse a una llamada debe transmitir un mensaje de Leaf Setup Request, disparar el timer T331 y entrar en un estado de Leaf Setup Initiated. Dentro de este mensaje de Setup la hoja debe incluir un elemento de información que identifica a ese pedido y tiene asociado el estado del mismo. Este número debe ser elegido de tal forma que no este asociado a algún otro estado.

Para identificar la llamada a la cual la hoja se quiere incorporar, el mensaje de Setup Request debe contener el elemento de información que identifica la llamada LIJ, y la dirección de la raíz (elemento de información Called Party Number y opcionalmente el elemento de información Called Party Subaddress). La hoja deberá incluir su dirección en el Calling Party Number y opcionalmente el elemento de información (IE) Calling Party Subaddress. El Calling Party Number no deberá incluir una dirección de Grupo. Si en el IE del Calling Party Subaddress se omite la red, asume que nada es necesario para identificar unívocamente a la hoja.

Nota: si la hoja intenta ingresar a una llamada inexistente o la Root LIJ no esta habilitada, el indicador de presentación del Calling Party Number en el mensaje de Setup Request no debe ser seteado a "Presentación estricta" entonces la raíz usa el Calling Party Number del mensaje de Request para generar el Called Party Number en el mensaje de Setup o Add Party que va a ser retornado a la hoja.

Como una opción, la hoja puede incluir el IE correspondiente a una red de tránsito en el mensaje de Leaf Setup Request, la red utiliza esta red de tránsito para setear el mensaje de Setup.

Si el timer T331 expira antes de recibir una respuesta, la hoja puede opcionalmente retransmitir el Setup Request. La hoja puede usar tanto el número de secuencia del Request original o seleccionar un nuevo valor. Si el mismo valor es usado, la hoja debería usar el estado existente de LIJ de la máquina, pero no va ser capaz de diferenciar una respuesta al primer Request (timer out) de la respuesta del segundo Request. Un nuevo número de

secuencia debe ser elegido si se requiere esta diferencia. Si la hoja elige no retransmitir el mensaje de Setup Request o si el timer T331 expira por segunda vez la hoja debe pasar al estado LIJ a Null.

Control de Información de Invalid Call/Connection o Service Request en el mensaje Leaf Setup Request

Cuando la red recibe el mensaje Setup Request no cambia el estado en la UNI de la hoja.

El Called Party Number y el Called Party Subaddress, si fue especificada, son usados para "forwardear" internamente el Request a la raíz. Si la red determina que la información de la llamada recibida desde el usuario es inválida, entonces la red envía un mensaje de Leaf Setup Failure usando el "Dummy Call Reference" y el mismo número de secuencia de hoja que el mensaje de Setup Request. El número y su dirección de Called Party en el mensaje de Setup Failure son tomados del número y su dirección de Calling Party del mensaje de Setup Request.

La causa usada en los mensajes de falla indica la razón de la misma, tales como:

#1 "Unassigned (unallocated) number"

#3 "No route to destination"

#22 "Number changer"

#28 "Invaled number format (incomplete number)"

Similarmente, si la red determina que el servicio de Request no esta autorizado, no implementado o no esta disponible, la red enviará un mensaje Leaf Setup Failure con la siguiente causa:

#63 "Service or option not available, unespecified" después de enviar un mensaje de Leaf Setup Failure, la red no cambia el estado UNI de la hoja.

Leaf Setup Request Received

Si la red puede determinar que el acceso para el Servicio LIJ requerido es autorizado y disponible, la red le da curso al mensaje Leaf Setup Request usando el Leaf Sequence Number recibido.

Leaf Setup Request Completed

Si el Leaf Setup Request es satisfactorio, un mensaje de Setup o Add Party es enviado a la hoja para agregarla a la llamada LIJ pedida. El mensaje Setup o Add Party enviado a la hoja contiene el número de secuencia de la hoja del Request correspondiente y puede también contener el identificador y los parámetros de la llamada LIJ. Si están presente, tanto en el Setup o en el Add Party, el Calling Party Number y el Calling Party Subaddress son los correspondientes a la raíz, como se especifico en el mensaje de Leaf Setup Request.

Nota: Normalmente el Calling Party Number del mensaje Setup o Add Party es igual al Called Party Number del Setup Request, sin

embargo esto no ocurre en todos los casos (pueden no ser iguales cuando se interconectan redes públicas y privadas).

Cuando se recibe un mensaje Setup o Add Party la hoja extrae el número de secuencia y lo usa para encontrar un estado de LIJ de la máquina. Si la hoja no tiene un estado de LIJ asociado con ese número de secuencia de hoja (si el timer de la hoja expiró previamente), entonces el mensaje de Setup o Add Party es tratado como un ofrecimiento de llamado independiente (esto no es en respuesta a un mensaje de Request) y aceptado o rechazado como se desee. Si la hoja tiene un estado de LIJ asociado con ese número de secuencia, entonces, detiene el timer T331 asociado, entra en el estado de "Null LIJ" y entonces sigue los procedimientos apropiados para aceptar o rechazar los Call Request entrantes.

Leaf Setup Request Failure

Si la red o raíz están deshabilitadas para completar el Join Request por alguna razón, un Leaf Setup Failure es retornado a la hoja usando el "Dummy Call Reference". El mensaje Leaf Setup Failure contiene la siguiente información tomada del Leaf Setup Request: el Leaf Sequence Number, el Called Party Number (tomado del Calling Party Number) y, opcionalmente el Called Party Subaddress (tomado desde el Calling Party Subaddress). La hoja usa el Leaf Sequence Number del mensaje Leaf Setup Failure para localizar el estado de LIJ y asociar la respuesta con el correspondiente Leaf Setup Request que disparó la respuesta. Una vez recibido el Leaf Setup Failure, el timer T331 es detenido y se entra en el estado de Null LIJ.

La causa usada en el mensaje de falla indica la razón de la misma:

- #21 "Call Rejected"
- #47 "Resources unavailable, unspecified"
- #49 "Quality of Service unavailable"

14.6.2. Procedimiento de incorporación en la Interface de la Raíz

Creación de la llamada LIJ

Para crear una llamada de incorporación propuesta por la hoja (Network LIJ), una para la cual la red automáticamente intenta incorporar las hojas solicitantes, la raíz usa un procedimiento de punto a multipunto e incluye el Calling Party Number, los parámetros de LIJ y los elementos de información (IE) de la llamada LIJ en el mensaje inicial de Setup. Todos los IE son obligatorios para la generación de una llamada de Network LIJ. Si los IE de los parámetros LIJ están especificados, excepto el Calling Party Number y el identificador de la llamada LIJ, la red deberá descartar el pedido de Setup (#56 "Mandatory element is missing"). Si la información de los parámetros de la llamada LIJ no se especifican, la red asume la creación de una llamada Root LIJ (y

nunca más va a intentar agregar automáticamente hojas solicitantes). Cuando se crea una llamada de Network LIJ, el identificador de la llamada LIJ especificado por la raíz debe ser único, por ejemplo este identificador no puede estar en uso para otra conexión de Network LIJ creada por esta raíz.

La raíz puede opcionalmente especificar los IE, Leaf Sequence Number y el Calling Party Subaddress cuando crea una llamada de Network LIJ. El Leaf Sequence Number es requerido si el mensaje de Setup es usado en respuesta a un mensaje de Leaf Setup Request. La red combina el Calling Party Number, el Calling Party Subaddress (si se especifica) y el identificador de la llamada LIJ para formar un label único global para la llamada de Network LIJ dentro de la red.

Cuando un mensaje de Setup es recibido, la red verifica que los IE de los parámetros LIJ sean válidos. Adicionalmente, los IE de los identificadores de las llamadas LIJ son chequeados y deben ser de un tipo y formato válidos, si alguno de estos chequeos falla, la red rechaza el pedido de Setup.

Pedidos Entrantes de Leaf Setup

Si una hoja intenta incorporarse a una llamada de Root LIJ o una llamada de Network LIJ que no están en el estado de link activo en la UNI de la raíz, pero para la cuál una raíz válida fue especificada, el mensaje de Leaf Setup Request debe ser entregado por la red a la raíz usando la "Dummy Call Reference". Una vez enviado el Leaf Setup Request a la raíz la red no debe cambiar el estado del link, no debe iniciar un nuevo "Party State", no debe disparar ningún timer y no debe guardar ninguna información interna relativa al Leaf Setup Request.

El mensaje Leaf Setup Request entregado por la raíz contiene todos los IE especificados por la hoja.

Aceptación de Incorporación Vía Setup

Asumiendo que la llamada todavía no existe, una vez que se recibe el Leaf Setup Request, la raíz tiene la opción de rechazar el Request o de aceptarlo creando una llamada a la hoja. En el caso en que la raíz elige crear la llamada, ésta asume que el identificador de llamada LIJ especificado por la hoja tiene algún significado para la raíz con respecto al tipo de llamada que va a ser creada (esto es, el identificador de la llamada LIJ implica un cierto descriptor de tráfico, calidad de servicio, etc.). Si no la raíz puede rechazar el Request.

La raíz crea una llamada punto a multipunto usando cualquier referencia de llamada disponible. La raíz tiene que incluir el número de secuencia de hoja del Leaf Setup Request, usado en el mensaje de Setup, para crear la llamada. El valor del Called Party Number debe ser tomado del Calling Party Number del Leaf Setup Request. Lo mismo para el Called Party Subaddress. La raíz puede crear la llamada como una llamada de Network LIJ, donde es soportado que las hojas se incorporen por medio de la red, o como una

llamada de Root LIJ, donde todos los mensajes de Leaf Setup Request son "forwardados" hacia la raíz.

Desde que la red entrega todos los paquetes de Leaf Setup Request a la raíz cuando la llamada indicada no esta en el estado de "Call Delivered", los mensajes de Leaf Setup Request pueden arribar al lado raíz de la interface cuando la llamada ya existe. Cuando la raíz desea agregar hojas pueden suceder 2 cosas:

- Los mensaje de Leaf Setup Request arriban cuando la llamada esta en un estado "bloqueado" (clearing), en este caso la raíz deberá guardar el mensaje de Leaf Setup Request hasta que la llamada se torne "nula". Entonces se procede como ya se describió.
- Los mensajes de Leaf Setup Request arriban cuando la llamada esta en un estado de establecimiento. En este caso la raíz deberá esperar hasta que la llamada entre en un estado activo o de "Call Delivered", entonces la raíz agrega la hoja usando los procedimientos de Add Party descrito más abajo. Si la llamada pasa de un estado de establecimiento a un estado en blanco o "Null", entonces la raíz debe proceder como en la cláusula 1.
Para las hojas que la raíz explícitamente agrega usando mensajes de Setup o Add Party, la raíz recibe confirmaciones explícitas si la inserción fue satisfactoria o falló.

Aceptación de Incorporación Vía Add Party

Asumiendo que la llamada ya existe cuando el Leaf Setup Request arriba, la raíz tiene la opción de aceptar o rechazar el Request agregando la hoja a la llamada especificada. Para agregar la hoja, la raíz emite un mensaje de Add Party conteniendo el número de secuencia de la hoja del Leaf Setup Request. El Called Party Number y el Called Party Subaddress deberán ser tomados del Calling Party Number y del Calling Party Subaddress del Leaf Setup Request. Si la llamada esta en un estado "bloqueado" cuando el Leaf Setup Request arriba la raíz deberá setear este mensaje hasta que la llamada tome el estado Null y después debe proceder como se detalló en el punto anterior. Si la llamada está en un estado de establecimiento, la raíz deberá esperar hasta que la llamada entre en un estado activo, "Call Delivered", o un estado "Null", agregando después la hoja usando un mensaje de Add Party o Setup.

Rechazo de Incorporación Vía un Leaf Setup Request

Cuando la raíz recibe un Leaf Setup Request y quiere rechazar la incorporación de la hoja, ésta deberá responder con un Leaf Setup Failure enviándolo sobre el "Dummy Call Reference", conteniendo el número de secuencia de la hoja del Leaf Setup Request con una causa apropiada. Si no se especifica una causa, la red insertará la causa #31 "Normal no especificada". El mensaje

Leaf Setup Failure contiene la siguiente información del Request, número de secuencia de la hoja, Called Party Number (tomado del Calling Party Number) y Called Party Subaddress (tomado del Calling Party Subaddress).

Call/Connection and Party Clearing

El limpiado de la Call/Connection y Party en las llamadas LIJ siguen los procedimientos de limpiado punto a multipunto con dos excepciones:

- Si la raíz borra el último Party que agregó en la Network LIJ Connection, deberá hacerlo enviando un mensaje de Drop Party en lugar de un mensaje de Release y continua manteniendo la llamada en un estado de link activo. Si no hay hojas activas en la llamada, la red puede limpiar la llamada. De otra manera la red debe mantener la llamada en un estado activo hasta que la última hoja se borre ella misma. Mientras tanto la raíz puede agregar "Parties" adicionales. Si la raíz quiere limpiar la llamada ésta envía un Release en cualquier momento.
- Cuando la red borra el último Party agregado por la raíz en una Network LIJ Connection, pero otras hojas (que se agregaron ellas mismas) están participando en la conexión, la red deberá enviar un mensaje de Drop Party en vez de un Release y continua manteniendo la llamada en un estado de link activo. Una vez recibido el mensaje de Drop Party, la raíz debería también mantener la llamada en un estado de link activo. La raíz enviará un mensaje de Release cuando la última hoja agregada se borra a sí misma de la llamada. Como establecimos anteriormente, la raíz puede enviar un mensaje de Release en cualquier momento para blanquear la llamada.

IP Versión 4

1. Introducción

Podemos ver a TCP/IP como un conjunto de protocolos dispuestos en un modelo de capas. Como la mayoría de los protocolos de la industria de la computación fue desarrollado para compartir recursos entre computadoras distantes que están interconectadas.

Generalmente las aplicaciones TCP/IP usan cuatro niveles:

- Un protocolo de aplicación, como mail.
- Un protocolo como TCP, que provee servicios necesarios para las aplicaciones.
- IP, que provee los servicios básicos para transportar datagramas al destino.
- Los protocolos necesarios para manejar los medios físicos, por ejemplo Ethernet.

Esta parte del informe va a estar referida al nivel IP, y en particular a la versión 4 del mismo.

2. Definición del Nivel IP

El protocolo IP surgió para interconectar redes. El principal trabajo de IP es buscar una ruta para que los datos lleguen al destino. Con respecto al modelo OSI (7 capas) podemos ubicar a este protocolo en el tercer nivel (Capa de Red).

Una de las principales características de IP es que no está orientado a conexión, esto quiere decir que para la transmisión de datos entre dos hosts no se construye ningún vínculo que los conecte antes del envío de los mismos. IP utiliza como unidad de transmisión el datagrama.

Cada host se individualiza mediante una dirección de 32 bits, dividida en 4 octetos. En ella se especifica un identificador de red y un identificador de host.

Para administrar estas direcciones se definieron diferentes clases:

- Clase A: 8 bits para red, 24 bits para hosts.
- Clase B: 16 bits para red, 16 bits para hosts.
- Clase C: 24 bits para red, 8 bits para hosts.
- Clase D y E: reservadas (D para multicasting)

Las direcciones origen y destino a nivel IP nunca cambian en la vida de una trama.

Para permitir a los dispositivos intermedios transmitir datagramas, éste cuenta con un encabezado en el que se especifican todos los parámetros de control necesarios para que el datagrama llegue a destino (dirección origen, dirección destino, tipo de protocolo, etc.). Además del encabezado, el datagrama contiene la porción de datos que se está queriendo transmitir. El encabezado tiene una parte fija de 20 octetos y una parte opcional de longitud variable.

3. Estructura del Datagrama

0	8	16	24	32
Versión	H Len	Tipo Servicio	Longitud Total del datagrama	
Identificador del datagrama		D	M	Offset
TTL	Protocolo		Checksum	
Dirección Origen				
Dirección Destino				
Opciones...				
Datos...				

Versión: indica a qué versión del protocolo pertenece cada uno de los datagramas. Mediante la inclusión de la versión en cada datagrama, no se excluye la posibilidad de modificar los protocolos mientras la red se encuentra en operación.

H Len: especifica la longitud que tiene el encabezado en palabras de 32 bits, es necesario puesto que la longitud del encabezado es variable.

Tipo Servicio: indica el tipo de servicio, es posible tener varias combinaciones con respecto a la seguridad y a la velocidad.

Longitud Total del Datagrama: incluye todo el datagrama, tanto el encabezado como los datos, está expresado en bytes.

Identificador del Datagrama: se necesita para permitir al destino determinar a qué datagrama pertenece el fragmento recién llegado. Todos los fragmentos de un datagrama contienen el mismo identificador (el identificador se asigna aleatoriamente).

Bit D: si este campo está seteado con 1 indica que el datagrama no se puede fragmentar.

Bit M: si este bit se encuentra en 1 significa que existen mas fragmentos, todos los fragmentos excepto el último deberán tener este bit seteado en 1.

Offset: es el desplazamiento del fragmento dentro del datagrama original. Se utiliza para regenerar el datagrama original.

TTL (Time To Live): es un contador que se utiliza para limitar el tiempo de vida de los paquetes. Cada vez que el datagrama pasa por un router el campo TTL se decrementa en 1, cuando llega a cero el datagrama se descarta.

Protocolo: indica el protocolo de nivel superior que el datagrama está transportando.

Checksum: es el campo que se utiliza para el reconocimiento de errores en IP, el alcance es sobre el encabezado. Divide al encabezado en palabras de 16 bits, las suma en complemento a 1 y al resultado los complementa a 1.

Direcciones Origen y Destino: especifican las direcciones IP del host origen y del host destino respectivamente.

Opciones: este campo se utiliza con fines de seguridad, informe de errores, depuración, así como para otro tipo de información. Permite también, incluir a versiones de protocolos subsiguientes información que no esta presente en el diseño original.

4. Ruteo

Como se menciona anteriormente IP es responsable de llevar un datagrama al destino indicado en el encabezado, pero poco se dijo cómo se hace. La tarea de encontrar como llevar un datagrama al destino es conocida como Routing.

Es necesario conocer el modelo en que IP esta basado. IP asume que cada host esta conectado a una red local, también se asume que el host puede enviar el datagrama dentro de la misma red. Pero el problema surge cuando se quiere enviar un datagrama a un host que se encuentra en una red diferente. Este problema es manejado por los Gateways (dispositivos intermedios). Un gateway es un sistema que conecta a una red con una o mas redes, generalmente son computadoras normales con mas de una interface de red. En muchos casos, gateways de propósitos especiales proveen mejor performance y confiabilidad que los gateways de propósitos generales.

El ruteo en IP se basa en el número de red de la dirección destino. Cada computadora tiene una tabla de números de red. Para cada número de red se tiene un gateway que es el que se intentará alcanzar si se desea enviar un datagrama a la red asociada.

Hay que notar que el gateway no tiene que estar conectado directamente a la red, pero éste debe ser teóricamente el mejor ubicado para acceder a la red.

Cuando una computadora desea enviar un datagrama, primero chequea si la dirección destino está en su red local, si esto ocurre el datagrama puede enviarse directamente, de otra manera el sistema espera encontrar una entrada en la tabla para la dirección destino y utiliza ese gateway para enviar el datagrama.

Las tablas pueden volverse muy grandes por lo cual existen técnicas para reducir el tamaño de las mismas. Una de estas técnicas consiste en definir un "default gateway" que es la única salida hacia fuera de la red. Este gateway debe conectar a la red local con las demás redes. En este caso no necesitaremos tener una entrada por cada red en el mundo, sino que simplemente tenemos un gateway como default, y si no encontramos una ruta específica para un datagrama, éste es enviado al gateway default.

Un gateway por default se puede definir aunque existan varios gateways en la red.

Existe la posibilidad de que un gateway mande un mensaje especificando que él no es la mejor opción e informando cual sí lo es.

La mayor parte de las interfaces de red son diseñadas para usar este tipo mensajes para agregar o modificar entradas en la tabla.

Se recomienda que los host en forma individual no traten de buscar el camino hacia el destino final por ellos mismos, sino que dejen esta tarea a los gateways.

Para que los gateways puedan armar sus tablas de ruteo se necesitan protocolos de ruteo.

5. Detalles de Direccionamiento: Subredes y Broadcasting

Algunas organizaciones creen conveniente dividir su número de red en subredes, esto se realiza utilizando algunos de los bits de la dirección IP reservados para host. Para determinar a que subred pertenece una dirección se utiliza una máscara (se efectúa un AND lógico entre la dirección y la máscara).

Supongamos que contamos con una red R1 a la que le fue asignada una dirección de Clase B 128.6; y deseamos usar el tercer octeto de la dirección IP para indicar cuales host son Ethernet dentro de la red. Esta división no tiene sentido fuera de R1, una computadora de otra red enviará los datagramas direccionados a 128.6 de la misma manera. De esta manera las computadoras fuera de R1 no tendrán diferentes rutas para 128.6.4 o 128.6.5. Pero dentro de la red R1, a las direcciones 128.6.4 y 128.6.5 las vemos como redes separadas. En efecto los gateways dentro de la Red R1 tienen entradas separadas para cada subred, mientras que los gateways que se encuentran afuera de R1 cuentan con una entrada para 128.6

Dentro de las direcciones IP los números 0 y 255 tienen un significado especial, o son reservado para máquinas que no conocen su dirección. En ciertas circunstancias es usado por máquinas que no conocen el número de red en la que se encuentra, aún conociendo su propio número de host,

por ejemplo 0.0.0.23 es un máquina que conoce su número de host pero desconoce el número de red a la cual pertenece.

El número 255 es usado para "broadcast". Un mensaje de Broadcast es aquel que todos los host pueden leer. Este es usado en algunas situaciones donde se desconoce la dirección con el host que queremos comunicarnos. Algunas veces no se conoce la dirección del "name server" más cercano, en este caso se debe enviar un Request como broadcast.

Existen casos en donde un host esta interesado en enviar la misma información a varios host. Es más simple entonces, enviar un simple broadcast a las máquinas en cuestión que enviar un datagrama individualmente a cada host. Para enviar este tipo de broadcast se debe usar una dirección que esta construida usando el número de red seguido de unos en la parte de la dirección que corresponda al número de host (por ejemplo si la máquina se encuentra sobre la red 128.6.4 deberá usar 128.6.5.255 como broadcast).

La implementación de broadcast depende del medio físico, en muchos casos no es posible usarlo, sin embargo sí es posible sobre Ethernet.

Debido a que 0 y 255 son usados para direcciones desconocidas y de broadcast respectivamente, un host nunca debe tener asignado como dirección ni la 0 y ni la 255.

Las direcciones nunca deben comenzar con 0 o 127.

6. Fragmentación y Reensamblado del Datagrama

TCP/IP está diseñado para usarse para diferentes clases de redes. Desafortunadamente los diseñadores de redes no se ponen de acuerdo acerca del tamaño optimo del paquete a ser enviado. Ethernet puede usar paquetes de 1500 octetos de longitud, mientras que los paquetes de Arpanet tienen un máximo de alrededor de 1000 octetos. Hay redes de gran velocidad que pueden usar paquetes de mayor longitud. En principio se puede pensar que IP utiliza el paquete mas pequeño, pero esto causa serios problemas de performance, cuando se transfiere archivos grandes, los grandes paquetes son más eficientes que los chicos. Por lo tanto lo que es deseable es usar el tamaño mas largo posible.

Supongamos que contamos con dos host en diferentes redes Ethernet (capaces de transmitir paquetes de 1500 octetos) conectadas a través de una red que las vincula pero que transmite paquetes de 200 octetos. La máquina origen transmite un datagrama de 1500 octetos. Cuando éste paquete llega al link de 200 octetos debe ser fragmentado a este número para poder llegar a la red destino y ser reensamblado en el host destino. A este proceso se lo llama Fragmentación y Reensamblado.

7. ARP: Address Resolution Protocol

El ARP es un protocolo para resolver el mapeo de direcciones IP a direcciones de nivel 2. Trabaja en forma paralela a IP. Describiremos el funcionamiento de este protocolo mediante un ejemplo:

Supongamos que tenemos un Host 128.6.4.194 (A) y nos queremos conectar con el Host 128.6.4.7 (B). Verificamos primero que B se encuentre

sobre la misma red, entonces se examina la tabla de ARP local para verificar que existe la dirección Ethernet asociada a la dirección IP en cuestión, si es así se envía el datagrama.

Ahora supongamos que el host no encuentra la dirección Ethernet asociada en la tabla de ARP local, entonces se utiliza el protocolo ARP para enviar un Request. Todos los Host escuchan el ARP Request. Cuando un host interpreta que el ARP Request es para él, responde. Esta respuesta se hace mediante un ARP Reply informando al que originó el request la dirección Ethernet del que responde. El host origen salvará la información en la tabla de ARP local para enviar futuros paquetes directamente.

La mayoría de los host tratan a las tablas de ARP como una cache y limpian periódicamente las entradas que no son usadas.

Notemos que la forma de enviar en ARP Request es por medio de un paquete de "broadcast". Los ARP request no se pueden enviar directamente a un Host determinado. Muchos host utilizan los ARP Request que le arriban para actualizar su propia tabla ARP.

8. Sistema de Dominios

Generalmente, el software de red utiliza direcciones IP de 32 bits para enviar datagramas, sin embargo los usuarios prefieren utilizar nombres en lugar de números. De esta manera podríamos contar con una base de datos que permita asociar nombres a direcciones IP, esto implicaría tener una tabla con las direcciones-nombres del resto de los Host. Esta solución es simple si contamos con una red pequeña, pero se vuelve poco práctica para redes de gran tamaño.

En el caso de redes grandes en lugar de tablas se tiene un conjunto de servidores de nombres interconectados.

Los servidores de nombres forman un árbol que se corresponde con una estructura institucional. Estas instituciones conforman el sistema de dominios y delegan la autoridad sobre los nombres a instituciones jerárquicamente inferiores en el árbol del sistema de dominios.

Un ejemplo de un nombre es AYELEN.INFO.UNLP.EDU. Este nombre representa a una computadora del Departamento de Informática de la Universidad Nacional de La Plata. Para saber donde se encuentra EDU, debemos consulta a un servidor raíz. EDU mantiene a las instituciones educacionales. El servidor raíz cuenta con varios servidores para EDU, por lo tanto debemos consultar a EDU acerca del servidor para UNLP y así sucesivamente hasta completar la dirección. Cada uno de estos niveles es conocido como "dominio".

Afortunadamente, generalmente no es necesario realizar éste procedimiento. Notemos que el servidor de nombres raíz es el servidor de nombres del nivel mas alto de los dominios tal como EDU. De esta manera un simple query sobre el server raíz nos llevará a UNLP. Además el software recuerda las consultas anteriores, esto permite recordar dónde buscar los servidores para un nombre dado. Cada una de éstas piezas de información tiene un tiempo de vida asociado (del orden de días), luego de éste tiempo la información expira y debe ser obtenida nuevamente.

Cada nombre de dominio es un nodo en una base de datos. El nodo puede tener registros que especifican un número de propiedades diferentes (por ejemplo: direcciones IP, tipo de computadora y una lista de servicios provistos para una computadora). Un programa puede preguntar por una pieza específica de información, o por toda la información acerca de un nodo dado. También es posible definir un alias para un nodo de la base de datos.

El sistema de dominios también puede usarse para almacenar información acerca de los usuarios, listas de mail y otros objetos.

Existe un standard que define la operación sobre las base de datos, tales como protocolos usados para realizar consultas sobre ellas. Cada red debe ser capaz de realizar tales consultas.

IP Versión 6

1. Introducción

Con los cambios producidos en Internet y los negocios sobre la red, la versión 4 de IP se está tornando obsoleta. Hoy en día Internet se ha convertido en un entorno muy rico en aplicaciones multimediales. Todo esto apoyado en el auge de la Web. Al mismo tiempo las redes de corporaciones han pasado de usar simples e-mail y file transfer a usar complejas aplicaciones Cliente-Servidor. Todos estos desarrollos sobrepasaron las capacidades para soportar o satisfacer las necesidades de funciones y servicios de las redes basadas en IP. Un ambiente de trabajo en Internet necesita soportar Tráfico de Tiempo Real, esquema de control de congestión flexible y características de seguridad. Ninguno de estos requisitos es fácilmente alcanzado con el existente estándar de IP, la versión 4 de IP (IPv4). Otra de las causas que impulsó el desarrollo de un nuevo estándar está dado por el gran crecimiento que tiene Internet y como consecuencia de esto las direcciones de IPv4 de 32 bits resultan insuficientes para tal requerimiento.

Este nuevo estándar para IP es oficialmente conocido como IPv6, ésta nueva versión del protocolo aumenta las capacidades de la versión actual (IPv4) e incorpora nuevos conceptos y funcionalidades. El objetivo de este capítulo es describir estos aspectos.

Los cambios de la versión 4 a la versión 6 se pueden agrupar de la siguiente forma:

- *Extensión en la capacidad de direccionamiento:* IPv6 incrementa el tamaño de direcciones IP de 32 bits a 128 bits. Esto permite más niveles de jerarquías de direccionamiento, un número mayor de nodos direccionables y un modo simple de autoconfiguración de direcciones (Esta capacidad provee asignación dinámica de direcciones IPv6).
- *Simplificación en el formato del Header:* Algunos campos del header IPv4 han sido eliminados o hechos opcionales para reducir el costo de los casos más comunes del procesamiento del paquete y para limitar el costo del ancho de banda del header IPv6.
- *Mejor soporte para extensiones u opciones:* Se introdujeron cambios en la manera de codificar las opciones del header IP para hacer más eficiente el "forward" de paquetes, límites menos estrictos sobre la

longitud del campo opciones, y mayor flexibilidad para agregar opciones en el futuro.

- *Capacidad de etiquetamiento de flujo:* Una nueva capacidad es agregada para habilitar el etiquetamiento de paquetes pertenecientes a un flujo particular para los cuales el enviador pide que se lo maneje en forma especial, tal como "Servicio de Tiempo Real".
- *Capacidad de autenticación y privacidad:* Extensiones para soportar autenticación, integridad de datos y opcionalmente confidencialidad de datos.

2. Formato del Header

Versión	T.Class	Flow Label	
Payload Length		Next Header	Hop Limit
Source Address			
Destination address			

A continuación se describen los campos del Header:

Versión: 4 bit. Indica la versión de IP.

T.Class (Traffic Class): 8 bit. Clase de Tráfico.

Flow Label: 20 bits. Nivel de Flujo.

Payload Length: Un entero sin signo de 16 bits. Representa la longitud del resto del paquete en octetos.

Next Header: Selector de 8 bits. Identifica el tipo de Header que esta inmediatamente seguido al de IPv6. Usa los mismos valores de IPv4.

Hop Limit: Un entero sin signo de 8 bits. Decrementado en 1 cada vez que pasa un nodo. Si llega a 0 se descarta.

Source Address: Es la dirección del origen. 128 bits.

Destination Address: Es la dirección del destino. 128 bits.

3. Headers de las Extensiones de IPv6

Cierta información opcional se codifica en Headers separados que son puestos entre el Header de IPv6 y el campo de datos. Estas extensiones se identifican por valores diferentes en el campo de Next Header. Con solo una excepción, estos Headers no son procesados por los nodos a través del camino. Además, estos Headers deben ser procesados en el estricto orden en que aparecen en el paquete. La excepción antes mencionada hace referencia al Header de Hop-by-Hop Option que lleva información que debe ser procesada y examinada por todos los nodos en el camino (incluido origen y destino). Si este Header esta presente debe ir inmediatamente después del Header de IPv6. La presencia de éste se indica con el valor 0 en el campo Next Header del Header de IPv6.

Extensiones definidas:

- Hop-by-Hop Options
- Routing (type 0)
- Fragment
- Destination Options
- Authentication
- Encapsulating Security Payload

3.1 Orden de los Headers Extention

Cuando se usa mas de un Header Extention en el mismo paquete se recomienda que estos aparezcan en el siguiente orden:

IPv6 Header
Hop-by-Hop Options Header
Destination Options Header
Routing Header
Fragment Header
Authentication Header
Encapsulating Security Payload Header
Destination Options Header
Upper-Layer Header

Todos los Header pueden aparecer solamente una vez, a excepción del Destination Options Header que puede aparecer 2 veces, o antes del Routing Header o antes del Upper-Layer Header. Los nodos deben estar preparados para intentar procesar los Headers en cualquier orden y sin importar la cantidad de veces que aparezcan en un mismo paquete, excepto el Hop-by-Hop Options Header.

3.2. Options

Dos de los Headers (Hop-by-Hop Options Header y Destination Options Header) pueden llevar un número variable de TLV (type-length-value) para codificar las opciones. Tiene el siguiente formato:

Option Type	Opt. Data Len	Option Data
-------------	---------------	-------------

Option Type: 8 bits. Identificador del tipo de opción.

Opt Data Len: 8 bits. Entero sin signo. Es la longitud del campo de datos

Option Data: Campo de longitud variable. Son los datos.

Todas las opciones deben ser procesadas en el orden estricto en que fueron puestas.

Los valores del campo del Option Type están codificados de tal manera que si un nodo no reconoce uno de estos tipos, los dos primeros bits le indican que debe hacer con el paquete:

00-Saltar el procesamiento del Header.

01-Descarta el paquete.

10-Descarta el paquete y manda un mensaje de error, independientemente de la dirección de destino.

11-Descarta el paquete y manda mensaje de error si la dirección destino no es dirección de multicast.

El tercer bit indica cuando los datos en campo de datos pueden cambiar en el camino. Si es 0 no puede cambiar. Si es 1 puede cambiar.

3.3. Hop-by-Hop Options Header

Usado para transportar información opcional que debe ser examinada por todos los nodos del camino que recorre el paquete. Se identifica con el campo Next Header en el Header del paquete igual a cero.

Tiene el siguiente formato:

Next Header	Hrd Ext Len	
OPTIONS		

Next Header: 8 bit. Indica el tipo de Header que le sigue.

Hdr Ext Len: Entero sin signo. 8 bits. Da la longitud en octetos del Hop-by-Hop Options no incluyendo los primeros 8 octetos.

Options: Campo de longitud variable, que contiene uno o más TLV.

3.4. Routing Header

Es usado por el origen para listar uno o más nodos intermedios para alcanzar el destino. Es muy similar a las opciones de ruteo de IPv4.

Este Header se setea con el número 43 en campo Next Header del Header anterior.

Tiene el siguiente formato:

Next Header	Hdr Ext Len	Routing Type	Segment Left
Type-Specific Data			

Next Header: 8 bit. Indica el tipo de Header que le sigue.

Hdr Ext Len: Entero sin signo de 8 bits. Da la longitud en octetos del Routing Options no incluyendo los primeros 8 octetos.

Routing Type: 8 bits. Identificador de una variante particular del Routing Header.

Segment Left: Entero sin signo de 8 bits. Número de segmentos restantes (los que faltan visitar).

Type-Specific Data: Campo de longitud variable. Es el campo de datos.

Cuando un nodo procesa un Header de Routing con un valor de Routing Type no conocido pueden pasar dos cosas:

- Si el Segmento Left está en 0 (cero) el nodo debe ignorar este Header y seguir procesando el siguiente Header.
- Si el Segment Left no está en 0 (cero) se descarta un paquete y se manda un mensaje de error.
- Si después de procesar el Routing Header de un paquete recibido, el nodo intermedio determina que el paquete va a ser "reenviado"

a un link cuyo MTU es menor que el tamaño del paquete, el nodo debe descartarlo y enviar un paquete de control a la dirección destino.

El tipo 0 (cero) tiene el siguiente formato:

Next Header	Hdr Ext Len	Routing Type = 0	Segment L
Reserved			
Address 1			
Address 2			
.			
.			
.			
Address n			

Next Header: 8 bit. Indica el tipo de Header que le sigue.

Hdr Ext Len: Entero sin signo de 8 bits. Indica la longitud del Header en unidades de 8 octetos, sin incluir los primeros 8 octetos. Para el Type 0 del Routing Header, indica dos veces el número de direcciones en el Header.

Routing Type: Valor 0. 8 bits.

Segment Left: Entero sin signo de 8 bits. Número de nodos que quedan por visitar.

Reserved: 32 bits. Inicializado a 0 para transmisiones. Ignorado en la recepción.

Address (1..n): Vector de direcciones de 128 bits numerados de 1 a n.

Las direcciones Multicast no deben aparecer en el Header de Routing, ni en el campo Destination Address de un paquete que acarree un Header de Routing de tipo 0.

El Routing Header se examina sólo en los nodos identificados en la dirección destino del Header Principal.

Algoritmo de ruteo que procesa el Routing Header, cuando este es de tipo 0:

```

IF Segment Left = 0
Procesar el próximo Options Header
ELSE
IF Hdr Ext Len es impar
Envía un mensaje de error y descarta el paquete
ELSE
    Calcular N: el nº de direcciones es el Routing Header,
    dividiendo Hdr Ext Len por 2
    IF Segment Left más grande que N
    Envía un mensaje de error y descarta el paquete
ELSE
    Decrementar Segment Left en 1
    Calcular I: índice de la próxima dirección en el
    vector de direcciones restando N menos la
    cantidad que restan (Segment Left)
    IF Address(i) o el Destination Address es multicast
    Descarto el paquete
ELSE
IF IPv6 Hop Limit es menor o igual a 1
    Enviar mensaje de error y descartar
    paquete
ELSE
    Decremento Hop Limit en 1
    Esta listo para transmitir al destino

```

3.5. Fragment Header

El Fragment Header es usado por un origen IPv6 para mandar paquetes más grandes que el MTU del camino.

A diferencia de IPv4, en IPv6 la fragmentación es realizada por el nodo origen.

Este Header es identificado con el valor 44 en el campo Header anterior.

Formato del Header:

Next Header	Reserved	Fragment Offset	Res	M
Identification				

Next Header: 8 bits. El tipo del Header inicial de la parte fragmentable.

Reserved: 8 bits reservados. Inicializados a 0 para la transmisión. Es ignorada por el receptor.

Fragment Offset: Entero sin signo de 13 bits. Es la posición relativa de los datos del fragmento en el paquete original de la parte fragmentable.

Res: 2 bits. Es reservado. Inicialmente en 0 para la transmisión. Es ignorado por el fragmento.

M: 1 = más fragmentos.
0 = último fragmento.

Identificación: 32 bits.

Para cada paquete que va a ser fragmentado el nodo origen genera un valor de identificación. La identificación debe ser diferente de otro paquete fragmentado *recientemente** con la misma dirección origen y destino. El destino final mencionado es el destino final de la conexión y no el próximo destino a alcanzar en el caso de aparecer el Routing Header.

- **Recientemente:** significa el máximo tiempo de vida de un paquete incluyendo tiempo de tránsito del origen al destino y el tiempo que se tarda en reensamblar los fragmentos del mismo paquete. Sin embargo, el nodo origen puede no conocer el máximo tiempo de vida del paquete, en este caso el valor de identificación se mantiene como un contador de 32 bits incrementado cada vez que un paquete va a ser fragmentado.

El paquete original consta de dos partes, una que se puede fragmentar y otra que no. La parte que no es fragmentable consiste del Header IPv6 más algunos Options Header que deben ser procesados por los nodos en la ruta hasta el destino. La parte fragmentable consiste del resto del paquete con algunos Header Options que solo se procesarán en el destino final.

Formato de fragmentos:

Parte no fragmentable	Header Fragmentable	Primer Fragmento
Parte no fragmentable	Header Fragmentable	Segundo Fragmento
Parte no fragmentable	Header Fragmentable	Ultimo Fragmento

La longitud de los fragmentos es múltiplo de 8 octetos, salvo el último (que puede ser menor).

Cada fragmento esta compuesto de:

- 1) La parte No fragmentable del paquete original con la longitud del Payload del Header IPv6 original cambiada por la longitud de este

fragmento (excluyendo la longitud del Header IPv6), y el campo Next Header del último Header de la parte No fragmentable cambiada a 44.

2) Fragment Header:

- El valor de Next Header que identifica el primer Header de la parte fragmentable del paquete original.
- Un Fragment Offset conteniendo el Offset del paquete (en unidades de 8 octetos) relativo al comienzo de la parte fragmentable del paquete original.
- M flag:
 - 0 el fragmento es el último.
 - 1 El fragmento es del medio.
 - 2 Valor de identificación.



BIBLIOTECA
FAC. DE INFORMÁT
U.N.L.P.

3) Fragmento propiamente dicho.

La longitud de los fragmentos debe ser elegida de tal forma que los paquetes que quepan en el MTU del camino.

Reensamblado:

Reglas del reensamblado:

Un paquete original es reensamblado sólo desde fragmentos que tiene la misma dirección origen y destino y la misma identificación del paquete.

La parte No fragmentable del paquete a ser reensamblado consiste de todos los Header sin incluir el Fragment Header del primer fragmento, con las siguientes 2 modificaciones:

- El campo Next Header del último Header de la parte No fragmentable se obtiene desde el campo Next Header del Fragment Header del primer fragmento.
- La longitud del Payload del paquete original se calcula de la siguiente manera:

(8* Offset del último fragmento) + la longitud del último fragmento + longitud de los Options Header de la parte No fragmentable.

Condiciones de error que se pueden presentar al reensamblar paquetes:

- Si dentro de los 60 segundos de recibir el primer fragmento no se completan la totalidad de los paquetes, el reensamblado se abandona y todos los paquetes recibidos son descartados. Se envía un mensaje de error al origen.

- Si la longitud de un fragmento no es múltiplo de 8 y el M-flag del fragmento es 1 se descarta el fragmento y se envía un mensaje de error.
- Si la longitud del Offset de un fragmento son tales que la longitud del Payload del paquete reensamblado supera los 64 K de octetos, el fragmento debe ser descartado y se envía un mensaje de error.

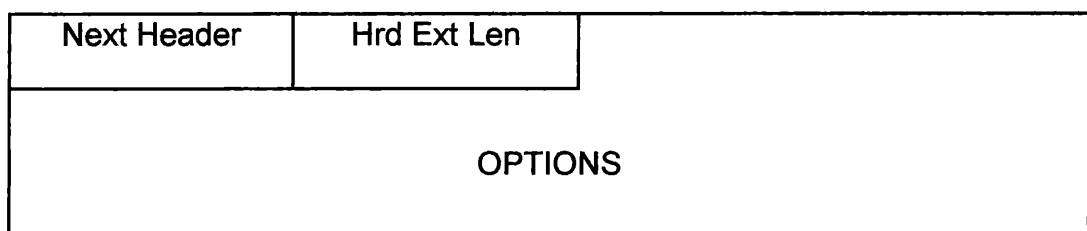
Condiciones que no son esperadas que ocurran, pero si ocurren no se consideran error:

- El n° y contenido de los Header que preceden al Fragment Header de los diferentes fragmentos de un mismo paquete pueden diferir. Solo los Header que están en el primer fragmento son los que se retienen para el reensamblado del paquete original.
- El valor del campo Next Header que se toma en cuenta para el reensamblado es el que aparece en el primer fragmento (pueden diferir entre los diferentes fragmentos).

3.6. Destination Options Header

El Destination Options Header es usado para acarrear información opcional para ser analizada en el destino final. Se identifica por el valor de Next Header 60 (igual que IPv4).

Tiene el siguiente formato:



Next Header: 8 bits selector. Identifica el tipo del Header que sigue inmediatamente al Destination Header.

Hdr Ext Len: Entero sin signo de 8 bits. Longitud del Destination Options Header en unidades de 8 octetos. No incluye los primeros 8 octetos.

Options: Campo de longitud variable. La longitud es tal que la longitud del Header debe ser múltiplo de 8 octetos. Contiene uno o más opciones TLV codificadas.

Hay 2 posibilidades de codificar la información opcional: una en el Destination Options Header y otra en un Header separado.

3.7. No Next Header

El valor 59 en el campo Next Header de un Header IPv6 o en cualquier extensión header indica que no sigue nada. Si el campo longitud del Header IPv6 indica presencia de octetos después del header que tiene el Next Header en 59, estos octetos son ignorados aún cuando el paquete es “forwardado” (no se modifican).

4. Consideraciones del tamaño del paquete

IPv6 requiere que cada link en Internet tenga un MTU de 1280 octetos o mayor. En un link que no pueda cubrir un paquete de 1280 octetos en una pieza, la capa inferior debe proveer la fragmentación y el reensamblado.

Se recomienda que los nodos IPv6 implementen Path MTU Discovery para descubrir y tomar las ventajas de tener un MTU mas grande que 1280 octetos. En implementaciones pequeñas de IPv6 se envía paquetes de 1280 octetos omitiendo la implementación del Path MTU Discovery.

Un protocolo de nivel superior que dependa de la fragmentación de IPv6 no debe enviar fragmentos que al ser reensamblados sean más grande que 1.500 octetos. A no ser que el origen conozca explícitamente que el destino puede reensamblar paquetes de ese tamaño.

En el caso de que un Router traductor IPv6-IPv4 tenga un MTU menor que 1280 y debe fragmentar, el origen debe proveer un Fragment Header en donde especifica el número de identificación del fragmento, que va a ser usada por los fragmentos IPv4.

- El Path MTU Discovery debe ser ejecutado aún en los casos donde un Host “piense” que un destino esta enganchado al mismo link que él.
- Los Links que tienen un MTU configurable deben ser seteados con al menos 1280 octetos y se recomienda hacerlo con 1500 o más para evitar fragmentación a nivel de IPv6 en caso de encapsulamiento.

5. Etiquetado de Flujo

El campo de Flow Label es de 20 bits en el Header IPv6, es usado por el origen para indicar un manejo especial para el paquete a los router IPv6.

Los Host o Router que no soportan la función del campo Flow Label tienen que setear a 0 este campo cuando crean el paquete, no se debe modificar el campo cuando se “forwardea”, el campo es ignorado cuando se recibe el paquete.

Un flujo es una secuencia de paquetes enviados desde un origen particular a un destino particular (unicast o multicast), para el cual el origen desea un manejo especial en los Router intermedios.

Un flujo es identificado unívocamente a través de la dirección origen y el Non-Zero Flow Label.

El Flow Label es asignado por el nodo origen. Los Label son elegidos en forma Pseudo-Random con un rango de 1 a FFFFF Hexa. El uso del

Random es para hacer que el Label funcione como una clave Hash en el Router para observar el estado asociado con el flujo.

Todos los paquetes pertenecientes al mismo flujo deben ser enviados con el mismo origen, destino y Label de flujo.

Si alguno de los paquetes incluye un Header de Hop-by-Hop entonces todos deben ser originados con el mismo contenido del Header (sin incluir el campo de Next Header).

Si uno de los paquetes incluye un Routing Header entonces todos los paquetes deben ser originados con los mismos contenidos en todos los Header Extension hasta e incluyendo el Routing Header (excluyendo Next Header del Routing Header).

A los router o destinos se les permite verificar estas condiciones, si se detecta una violación, pueden mandar un paquete de error al origen.

Los Router son libres de setear (set up) el estado del manejo de flujo para algunos de los flujos, aún cuando la información provista para el flujo no lo establezca explícitamente vía un protocolo de control, un Hop-by-Hop Options, u otro significado.

Este procedimiento puede incluir determinar el próximo salto (next hop interface) y otras acciones tales como modificar un Hop-by-Hop Options, avanzar el puntero y direccionar en un Routing Header, o decidir como encolar el paquete basado en su campo de prioridad.

Un router puede elegir "recortar" los resultados de algún tipo de procesamiento y "cachear" esa información, usando la dirección origen más el label del flujo como clave. Luego los paquetes con la misma dirección origen y label de flujo serán manejadas de acuerdo a la información de la cache, en vez de examinar todos aquellos campos que pueden asumirse que no han cambiado desde el primer paquete del flujo.

La información que es guardada en el Router para manejar el estado del flujo debe ser descartada en no más de n segundos (configurables) después que ha sido establecida, independientemente de si continúan arribando paquetes del mismo flujo. Si luego de descartar la información en la cache, llega un paquete con la misma dirección origen y Label de flujo, este se procesa normalmente y puede recrear la entrada en la cache.

El tiempo de vida del estado del manejo de flujo (información en la cache) es seteado explícitamente, por ejemplo por un protocolo de control o un Hop-by-Hop Options, y debe ser especificado como parte del mecanismo de seteo.

Un origen no puede reusar un Label de flujo dentro del tiempo de vida de algún flow-handling state.

Cuando un nodo se cae y luego reinicia, se debe tener cuidado de no usar un Label de flujo que podría estar siendo usado por algún flujo cuyo tiempo de vida no haya expirado. Esto puede ser solucionado guardando el Label de flujo que podrá ser recordado después de la caída o ateniéndose de usar algún Label de flujo hasta que el máximo tiempo de vida de algún flujo establecido previamente haya expirado.

6. Clases de Tráfico

El campo Traffic Class (8 bits) del Header en IPv6, permite al origen identificar la prioridad de sus paquetes. Los siguientes requerimientos generales se aplican sobre el campo *Traffic Class*:

- El valor default debe ser cero para todos los bits.
- Los nodos que soportan un uso específico (experimental o eventualmente estándar) de alguno o todos los bits de este campo tienen permitido cambiar el valor de éstos bits en los paquetes que ellos originen, reenvíen o reciban como lo requiera el uso definido. Los nodos deberían ignorar y no modificar aquellos bits de éste campo para los cuales no soportan un uso específico.
- Un protocolo de nivel superior no debe asumir que el valor que tienen los bits de éste campo en un paquete recibido es igual al valor de éstos bits cuando el paquete fue enviado por el origen.

NOTA: Los valores para el campo Traffic Class están bajo estudio y los disponibles son experimentales. Esta especificación toma como ejemplo los valores definidos en el RFC 1883

Los valores de prioridad son divididos en dos rangos:

- 0..7 son usados para especificar la prioridad del tráfico para la cuál el origen provee control de congestión.
- 8..15 usados para especificar la prioridad de tráfico de paquetes de tiempo real, por ejemplo.

Para control de congestión los siguientes valores de prioridad son recomendados:

- 0 – Tráfico no caracterizado.
- 1 – Tráfico "fillers".
- 2 – Transferencia de datos sin que el destino tenga que estar escuchando (e-mail).
- 3 – Reservado
- 4 – Transferencia de volumen (FTP).
- 5 – Reservado.
- 6 – Tráfico interactivo (Telnet).
- 7 – Control de tráfico Internet (protocolos de routing).

Para tráfico sin control de congestión el valor de prioridad más bajo (8) será usado para aquellos paquetes que el enviador esta dispuesto a descartar bajo condiciones de congestión y el máximo valor es el usado para indicar que el paquete no es deseable que sea descartado.

7. Condiciones de Protocolos de Nivel Superior

7.1. Checksums de Nivel Superior

Cualquier transporte o protocolo de nivel superior que incluya la dirección de IP en el cálculo del Checksum debe ser modificado para usar IPv6 ya que debe incluir una dirección de 128 bits en vez de 32 bits de IPv4.

Pseudo-Header de TCP y UDP para Ipv6:

Dirección origen	
Dirección Destino	
Upper Layer Packet Length	
Zero	Next Header

- Si el paquete contiene un Routing Header, la dirección destino usada en el pseudo-header es la del destino final. En el nodo original la dirección estará en el último elemento del Routing Header, en el receptor esa dirección estará en el campo de Destination Address del Header de IPv6.
- El valor Next Header en el pseudo-header identifica el protocolo de nivel superior, este diferirá del valor del Next Header en el header IPv6 si existen extensiones Header entre el Header de IPv6 y Header de nivel superior.
- La longitud del Upper Layer Packet usada en el pseudo-header, es la longitud del header del paquete de la capa superior mas los datos. Algunos protocolos de nivel superior llevan su propia información de longitud, para tales protocolos esto es la longitud usada en el pseudo header. Otros protocolos como TCP, no llevan su propia información de longitud, en éstos casos la longitud usada en el pseudo header es la longitud de payload del header IPv6 menos la longitud de cualquier Extention Header entre el header IPv6 y el header de nivel superior.
- A diferencia de IPv4, cuando un paquete UDP es originado por un nodo IPv6, el Checksum no es opcional. Siempre que se origine un paquete UDP, un nodo IPv6 debe calcular el Checksum UDP sobre el paquete y pseudo-header. Si el cálculo de cómo resultado

ceros, debe ser modificado a FFFF. El receptor IPv6 debe descartar los paquetes que contienen Checksum en cero y logear el error.

7.2. Tiempo de vida máximo del paquete

A diferencia de IPv4, los nodos IPv6 no requieren un tiempo de vida máximo, por esta razón el campo "Time to Live" de IPv4 ha sido cambiado por "Hop Limit" en IPv6.

7.3. Máximo tamaño de Payload de nivel superior

En IPv4 el MSS de TCP se calcula como el máximo tamaño de paquete (un valor por defecto o por Path MTU Discovery) menos 40 octetos (20 para la mínima longitud del Header IPv4 y 20 para la mínima longitud del Header TCP). Cuando se usa TCP sobre IPv6, el MSS se calcula como el máximo tamaño del paquete menos 60 octetos (la mínima longitud del Header IPv6 es de 20 octetos mayor al mínimo Header de IPv4).

7.4. Respondiendo a paquetes que acarrean Routing Header

Los paquetes enviados por un protocolo de nivel superior en respuesta a un paquete que acarrea un Routing Header no deben incluir headers de éste tipo que fueron automáticamente derivado por reversa del Routing Header recibido, a menos que la integridad y autenticidad de la dirección origen y el Routing Header recibido hayan sido verificados. Es decir, solamente los siguientes tipos de paquetes son permitidos en respuesta a un paquete recibido conteniendo un Routing Header:

- Paquetes que no acarrean Routing Headers.
- Paquetes que acarrean Routing Headers que no fueron derivados por reversa de los routing Headers Recibidos.
- Paquetes que acarrean Routing Headers que no fueron derivados por reversa de los routing Headers Recibidos si y solo si la integridad y autenticidad de la dirección origen y el Routing Header recibido hayan sido verificados.

8. Arquitectura del Direccionamiento en IPv6

8.1. Direcciones IPv6

Las direcciones IPv6 son identificadores de 128 bits para interfaces o conjuntos de interfaces. Existen 3 tipos de direcciones:

Unicast: es un identificador para solo una interface.

Anycast: es un identificador para un conjunto de interfaces. Un paquete enviado a una dirección anycast es entregado a una de las interfaces identificadas por esa dirección.

Multicast: un identificador para un conjunto de interfaces. Un paquete enviado a una dirección multicast es entregado a todas las interfaces identificadas por esa dirección.

8.2. Modelo de Direccionamiento

Todas las direcciones son asignadas a interfaces y no a nodos. Todas las direcciones unicast correspondientes a todas las interfaces pertenecientes a un nodo pueden ser utilizadas para identificarlo. A una interface se le puede asignar múltiples direcciones IPv6 de cualquier tipo.

Existen dos excepciones a este modelo, a saber:

- Una sola dirección se puede asignar a varias interfaces físicas si la implementación trata a estas múltiples interfaces físicas como una única interface cuando las presenta a nivel Internet.
- Los Routers pueden tener interfaces no numeradas, es decir que no tengan direcciones IPv6 asignadas, en los links punto a punto para eliminar la necesidad de configuración manual de direcciones,

Representación de Direcciones

Provider-Based Unicast Address	010
Link Local Use Adresses	1111 1110 10
Site Local Use Adresses	1111 1110 11
Multicast Adresses	1111 1111

8.3. Direcciones Conocidas

Dirección "no especificada": es la dirección 0:0:0:0:0:0:0:0 ("todos ceros"), nunca puede ser asignada a ningún nodo. Indica la ausencia de una dirección. No debe ser usada como una dirección destino en los headers de un datagrama IP o en los headers de ruteo (como destino intermedio).

Dirección de Loopback: es la dirección unicast 0:0:0:0:0:0:0:1, esta dirección puede ser usada por un nodo para enviarse datagramas IP a si mismo, nunca debe ser asignada a una interface. No debe ser usada como dirección origen en un datagrama IP enviado fuera de un nodo.

Un datagrama con una dirección destino igual a la de "loopback" nunca debe ser enviada fuera del nodo.

Provider-based-Unicast Address (Global):

3	n-bits	m-bits	p-bits	125-1-m-p bit
010	Registry-Id	Provider-Id	Subscriber-Id	Intra-Subscrib

La parte de mas alto orden de la dirección es asignada a los registradores, quienes asignan porciones del espacio de dirección para proveedores, quienes a entonces asignan el espacio de dirección a los suscriptores, etc. El identificador de registro permite identificar el registro que asigna al proveedor una porción de direcciones. El término "prefijo de registro" se refiere a la parte de más orden de la dirección incluyendo el identificador de registro.

El ID del proveedor identifica un proveedor específico, de la misma manera, el ID del suscriptor identifica a los suscriptores, y éstos últimos administran a la porción intra-suscriptor.

Direcciones Unicast: es de uso local: Existen definidas dos tipos de direcciones unicast de uso local, ésta son: Local Link y Site Link. Las primeras son para uso local dentro de un solo Link y tienen el siguiente formato:

10 bits	n-bits	118-n bits
1111 1110 10	0	Interface ID

Las direcciones de Link Local son diseñadas para ser usadas en direccionamientos sobre un Link simple y para propósitos tales como autoconfiguración de direcciones, neighbor discovery, o cuando no existen routers presentes.

Los Routers no deben dejar pasar ningún paquete con una dirección origen de Link Local.

Las direcciones de Site Local tienen el siguiente formato:

10 bits	n-bits	m-bits	118-n-mbits
1111 1111 11	0	Subred id	Interface id

Las direcciones de Site Local pueden ser usadas por sitios u organizaciones que todavía no están conectadas a la Internet Global. Estos no necesitan pedir un prefijo de dirección global de Internet. Los routers no deben "forwardear" ningún paquete con una dirección origen de Site Local fuera del Site.

Direcciones Anycast: Una dirección anycast es una dirección que es asignada a mas de una interface. Las direcciones anycast son tomadas del espacio de direcciones unicast usando cualquiera de los formatos definidos, por lo tanto las direcciones anycast son indistinguibles de las direcciones unicast.

Cuando una dirección unicast es asignada a mas de una interface se convierte un una dirección anycast. Los nodos a los cuales se les va asignar una dirección anycast deben ser explícitamente configurados.

Existen dos restricciones para el uso de las direcciones anycast:

Una dirección anycast no puede ser usada como origen en un paquete IPv6.

Una dirección anycast no debe ser asignada a un host, éstas solo pueden asignarse a los routers.

Dirección Anycast Requerida: La dirección anycast de subred-router es predefinida y tiene el siguiente formato:

n-bits	128-n bits
Subred Prefix	000000000...0

El prefijo de subred en una dirección anycast es el prefijo que identifica un link específico. Los paquetes enviados a esta dirección son entregados a un router en la subred. Todos los routers tienen que soportar esta dirección anycast para las subredes a las cuales tienen interfaces conectadas.

Direcciones Multicast: Una dirección multicast es un identificador para un grupo de nodos. Un nodo puede pertenecer a varios grupos multicast. El formato de una dirección de este tipo es el siguiente:

8	4	4	112 bits
1111 1111	Flags	Scop	Group ID

Colocando 1111 1111 en el comienzo de la dirección indica que la misma es multicast.

Flags: es un conjunto de cuatro flags 000T, los tres bits de más alto orden son seteados a cero.

T = 0 indica una dirección multicast asignada permanentemente (asignada por la autoridad de la Internet Global).

T = 1 indica una dirección multicast no permanentemente asignada (transitiva).

Scop: es un valor usado para limitar el alcance del grupo multicast. Los valores son:

- 0 Reservado
- 1 Node-Local Scope
- 2 Link-Local Scope
- 3 (unassigned)
- 4 (unassigned)
- 5 Site-Local Scope
- 6 (unassigned)
- 7 (unassigned)
- 8 Organization-Local Scope
- 9 (unassigned)
- A (unassigned)
- B (unassigned)
- C (unassigned)
- D (unassigned)
- E Global Scope
- F Reservado

Group ID identifica el grupo multicast, ya sea de manera permanente o transitiva dentro de un alcance dado. El significado de las direcciones multicast permanentemente asignadas es independiente del valor del alcance.

Por ejemplo el grupo de los NTP servers tiene un grupo ID de 43 hex entonces:

FF01:0:0:0:0:0:0:43 significa todos los NTP servers sobre el mismo nodo emisor.

FF02:0:0:0:0:0:0:43 significa todos los nodos sobre el mismo link del emisor.

FF05:0:0:0:0:0:0:43 significa todos los nodos sobre el mismo sitio del emisor.

FF0E:0:0:0:0:0:0:43 significa todos los nodos en la Internet.

La dirección multicast de nodo solicitado es computada como una función de la dirección unicast del nodo (o anycast). La dirección multicast de nodo solicitado se forma tomando los 32 bits de mas bajo orden de la dirección y se les agrega el prefijo de 96 bits FF02:0:0:0:0:1.

Por ejemplo: la dirección multicast de nodo solicitado correspondiente a la dirección IPv6 4037::01:800:200E:8C6C es FF02::01:800:200E:8C6C.

8.4. Direcciones requeridas para un nodo

Un host tiene que reconocer las siguientes direcciones como propias:

- La dirección de Link Local para cada interface.
- Las direcciones unicast asignadas.

- La dirección de loopback.
- La dirección multicast de todos los nodos.
- Las direcciones multicast de nodo solicitado para cada dirección unicast o anycast asignada.
- La dirección multicast de todos los otros grupos a los que pertenece.

8.5. Direcciones requeridas por un Router

- Las direcciones de Link Local para cada interface.
- Todas las direcciones unicast asignadas.
- La dirección de loopback.
- La dirección anycast de subred – router para los link para los cuales tiene interface.
- Todas las otras direcciones anycast que el router tiene configurada.
- Las direcciones multicast de todos los nodos.
- Las direcciones multicast de todos los routers.
- Las direcciones multicast de nodo solicitado para cada dirección unicast o anycast asignada.
- La dirección multicast de todos los otros grupos a los que pertenece.

Los únicos prefijos de direcciones que están predefinidos para una implementación son:

- La dirección no especificada.
- La dirección de loopback.
- Los prefijos unicast.
- Los prefijos multicast.
- Los prefijos de uso local: Link – Local, Site – Local.
- Dirección multicast predefinida.
- Prefijos compatibles con IPv4.

9. Neighbor discovery

Este protocolo resuelve un conjunto de problemas relacionados con la interacción de nodos atachados a un mismo link.

Los problemas que resuelve serian:

- Router Discovery: Como descubrir routers atachados al mismo link.
- Prefix Discovery:
- Parameters Discovery:
- Autoconfiguración de direcciones
- Resolución de direcciones
- Determinación de próximo salto
- Detección de vecino no alcanzable

- Detección de direcciones duplicadas
- Redirect

Este protocolo define 5 tipos de paquetes ICMP diferentes; la Solicitud de router, Aviso de router, Solicitud de vecino, Aviso de vecino y mensaje de redirect. Cada mensaje tiene el siguiente propósito:

- *Solicitud de router*: Cuando una interface se habilita un host puede enviar este mensaje para así recibir un aviso de router.
- *Aviso de router*: El router avisa su presencia y los parámetros de Internet a través de un mensaje de este tipo. Este mensaje puede contener los prefijos usados en un link como también el valor de hop limit, etc.
- *Solicitud de Vecino*: es enviado por un nodo para determinar la dirección de link layer de un vecino o para determinar que el vecino todavía está alcanzable vía la dirección almacenada. También se utiliza para la detección de direcciones duplicadas.
- *Aviso de vecino*: Mensaje de respuesta a una solicitud de vecino. Un nodo también puede enviar un aviso que no responde a ninguna solicitud, para avisar el cambio de su dirección de link layer.
- *Redirect*: usado por los routers para avisar a los hosts de un mejor primer salto para alcanzar un destino.

En los links con capacidad de multicast cada router periódicamente envía un aviso de router anunciando su disponibilidad. Un host recibe los avisos de router desde todos los routers construyendo así una lista de default routers. Los routers generan avisos de routers frecuentemente así los hosts pueden aprender de su presencia en un tiempo relativamente corto pero no tan frecuentemente como para detectar la falla de alguno (Se utiliza el algoritmo de unreachability para detectar las fallas). Los avisos de routers contienen una lista de prefijos para determinación on-link y/o configuración autónoma de direcciones; los flags asociados al prefijo especifican el uso particular del mismo. Los hosts utilizan los prefijos on-link anunciados, para construir y mantener una lista que es usada en la decisión de cuando un destino es on-link o está más allá de un router. Notemos que un destino puede estar on-link aun cuando no fue cubierto por un prefijo on-link anunciado, en este caso el router puede enviar un redirect al emisor informando que el destino es un vecino.

Los avisos de routers (y los flags por prefijo) permiten a los routers informar a los hosts como ejecutar la autoconfiguración de direcciones (por ejemplo pueden especificar cuando un host puede usar DHCP versión 6 o configuración automática). Los avisos de routers también contienen parámetros de Internet tal como el límite en los saltos que un host debe utilizar en los paquetes salientes y opcionalmente parámetros de link como el MTU.

Los nodos resuelven direcciones enviando un paquete multicast de solicitud de vecino, este paquete se envía a la dirección multicast de nodo solicitado de la dirección destino. El destino retorna su dirección en paquete unicast de aviso de vecino. Los mensajes de solicitud de vecino

son también utilizados, para determinar si mas de un nodo esta utilizando la misma dirección unicast.

Además con neighbor discovery se pueden manejar las siguientes situaciones: cambio de dirección de link, carga balanceada, direcciones anycast, avisos de proxy.

9.1. Modelo conceptual de un Host

Estructura de datos conceptual: Un host deberá mantener las siguientes piezas de información para cada interface.

- *Neighbor Cache*: un conjunto de entradas de vecinos individuales, a los cuales recientemente se les envió tráfico. Las entradas son identificadas por la dirección IP on-link unicast del vecino y contiene la dirección de link layer correspondiente, un flag indicando cuando el vecino es un router o un host y un puntero a una cola de paquetes esperando para completar la resolución de direcciones. Una entrada también contiene información usada por el algoritmo de neighbor unreachability detection incluyendo el estado de alcanzabilidad, el numero de pruebas sin contestar y el tiempo que el próximo evento de neighbor unreachability detection va a ejecutarse.
- *Destinación Cache*: Un conjunto de entradas destino para los cuales fue enviado recientemente mensajes, incluyendo ambos, destinos on-link y off-link, y provee un nivel de indirección dentro de la neighbor cache. La destination cache mapea una dirección destino IP a la dirección IP de próximo salto vecino. Esta cache es actualizada con la información aprendida de los mensajes de redirección. En las entradas puede ser útil guardar información tal como el path MTU.
- *Lista de prefijos*: Una lista de prefijos que define el conjunto de direcciones que están on-link. Las entradas son creadas con la información recibida en los avisos de router. Cada entrada tiene asociado un timer de invalidación usada para descartar el prefijo. Existe un valor "infinito" que tiene validez a menos que un nuevo valor finito sea recibido en un aviso de router posterior. El prefijo de link local esta en la lista de prefijos con un valor infinito. Los avisos de router no deben modificar el tiempo de invalidación de un prefijo de link-local.
- *Lista de default routers*: una lista de los routers a los cuales se les puede enviar paquetes. Las entradas en esta lista apuntan a entradas en la neighbor cache. Cada entrada también tiene Asociado un timer de invalidez (tomado de los avisos de los routers). Usado para borrar las entradas que no han sido reconfirmadas.

La neighbor cache contiene información mantenida por el algoritmo de unreachability detection, una pieza clave de la información es el estado de alcanzabilidad del vecino que puede tomar uno de estos 5 valores: incompleto, alcanzable, delay, stale y probe.

9.2. Algoritmo Conceptual de Envíos

Cuando se envía un paquete a un destino, un nodo usa una combinación de la destination cache, la prefix list y la default router list para determinar la dirección apropiada de próximo salto, esta operación se conoce como determinación del Next Hop.

Una vez que la dirección de próximo salto se conoce, se consulta la neighbor cache para determinar la información de nivel link del vecino.

La operatoria para este algoritmo para una dirección unicast dada es la siguiente: el emisor ejecuta un mapeo del prefijo con la prefix list para determinar cuando el destino está on-link u off-link. Si el destino está on-link la dirección de próximo salto es igual a la dirección destino del paquete, de otra manera el emisor elige un router de la lista de default routers, en el caso que esta lista esté vacía, el emisor asume que el destino está on-link.

Por razones de eficiencia la determinación de next hop no se ejecuta para cada paquete que se envía, sino que los resultados de una determinación de next hop, se guarda en la correspondiente entrada de destination cache. Cuando un nodo tiene un paquete para enviar primero examina la destination cache, sino existe una entrada para el destino el algoritmo de determinación de next hop es invocado para crear una entrada en la destination cache.

Una vez que la dirección de próximo salto se conoce, el emisor examina la neighbor cache para consultar la información de link layer. Si no existe una entrada el emisor crea una, setea su dirección a incompleto, inicia la resolución de la dirección y encola los paquetes esperando que esta finalice.

Para interfaces con capacidades multicast la resolución de direcciones consiste, en enviar una solicitud de vecino y esperar el aviso de vecino. Cuando el aviso de vecino es recibido la dirección de link se guarda en la entrada de la neighbor cache y los paquetes encolados son transmitidos. Para paquetes multicast el next hop es siempre la dirección destino y es considerada on-link.

Cada vez que una entrada en la neighbor cache es accedida cuando se transmite un paquete unicast, el emisor chequea la información relacionada con la alcanzabilidad del vecino.

El next hop determination se realiza la primera vez que se envía tráfico a un destino, mientras la comunicación al destino continúe estable, la información de la entrada en la destination cache es utilizada. Si en algún punto la comunicación se corta, (determinado por el algoritmo de unreachability detection) la determinación de próximo salto puede que se necesite ejecutar nuevamente.

Notemos que cuando un nodo rehace la determinación de next hop no necesita descartar la entrada en la destination cache ya que contiene información que le puede ser útil como por ejemplo el path MTU.

IP Versión 6 sobre una LAN ATM

1. Introducción

El principal problema que encontramos cuando se encapsula IPv6 sobre Redes ATM, es que la tecnología de éstas redes no provee servicios multicast no orientados a conexión a nivel de Link, y mas aún, ATM no provee en forma inherente una estructura para las diferentes necesidades de IPv6, tales como Neighbor Discovery, Address Configuration, etc., mientras que IPv6 se basa en gran medida en las capacidades multicast no orientadas a conexión para resolver dichas necesidades.

En la versión anterior a IPv6, IPv4, las funciones de resolución y configuración de direcciones están ubicadas en el nivel de Link, es decir que los protocolos que resuelven este comportamiento no son parte de los protocolos básicos de IP, pero forman parte de cada nivel de Link, por ejemplo ARP para medios broadcast, ATMARP para redes ATM. De esta manera cada nivel de Link nuevo tendría que definir protocolos propietarios para resolución de direcciones, un ejemplo de este caso lo constituye el protocolo de resolución de direcciones ATMARP. Contrariamente en IPv6 la resolución y configuración de direcciones están ubicadas en el nivel de red en vez de estar en el nivel de Link, es decir, los protocolos de Neighbor Discovery que utiliza IPv6 para llevar a cabo la tarea de detección de Vecinos y Routers son parte integral del nivel de Red de IPv6 y cualquier mecanismo que sea usado para encapsular IPv6 en ATM deberá adaptarse a los protocolos de Neighbor Discovery. Debido a esto en IPv6 todos los niveles de Link deben manejar todos los paquetes de Neighbor Discovery y usarlos para resolución de direcciones, detección de routers y configuración de direcciones. En el caso de que no se use Neighbor Discovery se requerirá la modificación del protocolo de Red IPv6 para que pueda operar en el Link específico.

Por otro lado, en IPv6 se presenta una nueva característica que es la posibilidad de manejar prioridades en el ruteo de paquetes, y teniendo en cuenta que en ATM las prioridades se ofrecen a través de los QoS negociados para una conexión, surge la posibilidad de aprovechar ambas características y definir una correspondencia "*prioridad IPv6 - QoS ATM*", que sea utilizada no solo para el ruteo de paquetes IPv6 sino que también puedan utilizarse para la distribución de éstos paquetes dentro de un Grupo de Vecinos.

2. Términos Relacionados

2.1 Terminología de IPv6

Nodo: un dispositivo que implementa IPv6, los nodos pueden estar conectados a otros tipos de redes.

Router: un nodo que reenvía paquetes IP que no son específicamente direccionados para él.

Host: cualquier nodo que no es un router.

Link: un medio sobre el cual los nodos se pueden comunicar a nivel de Link (ejemplo Ethernet, ATM).

Neighbor: nodos “enganchados” al mismo Link (vecinos).

Interface: el punto de conexión de un nodo a un Link (tarjeta).

Dirección Unicast: un identificador para una sola interface.

Dirección Multicast: un identificador para un conjunto de interfaces. Un paquete enviado a una dirección multicast es entregado a todas las interfaces identificadas por esa dirección.

Dirección Link Layer: un identificador de Link Layer para una interface (dirección de nivel dos).

Dirección Link Local: una dirección que tiene alcance de Link solamente. Puede ser usada para alcanzar a los vecinos. Todas las interfaces tienen una dirección de Link Local.

Dirección de Site Local: una dirección que tiene el alcance limitado al Site Local.

Dirección Global: una dirección que no tiene límites en su alcance.

Token de Interface: un identificador dependiente del Link para una interface, éste es al menos único por Link.

Dirección Anycast: un identificador para un conjunto de interfaces. Un paquete enviado a una dirección anycast es entregado a una de las interfaces identificadas por esa dirección.

2.2. Terminología ATM

Punto-a-Punto (PTP): una conexión que conecta directamente dos nodos ATM. Un mismo conjunto de nodos puede tener múltiples PTP con cualquier otro.

Punto-a-Multipunto (PTM): una conexión ATM que conecta un nodo con varios nodos. El mismo conjunto de nodos puede tener múltiples PTM con otros.

Dirección de Grupo: éste tipo de direcciones es específico de la UNI 4.0.

Registración de Dirección: el proceso por el cual un nodo ATM informa a la red ATM de su dirección de Link Layer. Un nodo puede registrar múltiples direcciones y éstas pueden tomar diferentes formas.

Canal Multicast: es un canal punto-a-multipunto definido utilizando la interface UNI 4.0.

2.3. Nuevos Términos

Grupo de Vecinos: conjunto de nodos en una LAN ATM, donde cada nodo puede alcanzar a otro mediante la dirección de Link Local o por medio de un broadcast.

Servidor D'Artagnan: es una estación ATM que provee los servicios necesarios para soportar la distribución de paquetes multicast.

Cliente: es la mínima unidad (router o host) que puede participar en un Grupo de Vecinos.

Capa Cliente: es una pieza de software que se encuentra en los nodos. Esta encargada de analizar los paquetes, determinar cuales son multicast para enviarlos al Servidor D'Artagnan y cuales pertenecen a una comunicación Unicast, además de determinar los QoS dependiendo de las prioridades que posean los mismos.

Capa Servidora: es una pieza de software que se encuentra en los Servidores D'Artagnan. Provee la funcionalidad necesaria para poder distribuir los paquetes Multicast que le entregan los clientes.

3. Canales

Existen tres categorías de canales definidos en este documento, cada categoría representa el tipo de conexión y los nodos involucrados en la misma, es decir:

- **Canal Punto-a-Punto entre nodos:** estos canales son los que se establecen para transmisión de datos entre nodos y se crean dinámicamente según las necesidades de los mismos. La categoría de QoS seteada para las conexiones depende de la prioridad del paquete IPv6, sin embargo los nodos participantes pueden renegociar las características de la conexión.

- **Canal Punto-a-Punto entre un nodo y el Servidor D'Artagnan:** estos canales son los que se establecen entre un nodo y el servidor D'Artagnan. Son creados en forma dinámica por el nodo que desea enviar un paquete multicast. Son unidireccionales debido a que solo transportan paquetes multicast desde el nodo hacia el Servidor. La categoría de QoS seteada para estas conexiones depende de la prioridad del paquete IPv6, sin embargo el nodo y el Servidor pueden renegociar las características de la conexión.
- **Canal Punto-a-Multipunto entre el Servidor D'Artagnan y un conjunto de nodos:** este canal se establece entre el Servidor D'Artagnan y los nodos pertenecientes al Grupo de Vecinos, y es denominado Canal Multicast. Por este canal arribarán a los nodos los paquetes multicast distribuidos por el Servidor.
- **Canal Punto-a-Punto entre Servidores D'Artagnan:** estos canales son los que se establecen para transmisión de información de Replicación entre Servidores D'Artagnan y se crean dinámicamente según las necesidades de los mismos.

4. Encapsulamiento

Dentro de este modelo todos los clientes utilizan encapsulamiento LLC/SNAP. Dependiendo de las partes que intervienen en la comunicación se definen dos formas de encapsulamiento:

- **Encapsulamiento Servidor:** Este tipo de encapsulamiento se utiliza en las comunicaciones que involucran como destino final un servidor D'Artagnan, estas son, Cliente-Servidor o Servidor-Servidor.

Este gráfico muestra como sería el header para un paquete de una comunicación como la mencionadas anteriormente.

LLC/SNAP (0xAA – 0xAA – 0x03)
OUI (0x00 – 0x00 – 0x00)
PID (0xXX – 0xXX)

Nota: El valor PID es a determinar, esto quiere decir que debe ser diferente a cualquier valor ya utilizado, pero su valor en si mismo no tiene relevancia.

- Encapsulamiento Cliente: Este tipo de encapsulamiento se utiliza en las comunicaciones que involucran como destino final un cliente, estas son, Cliente-Cliente o Servidor-Cliente.

Este gráfico muestra como sería el header para un paquete de una comunicación como la mencionadas anteriormente.

LLC/SNAP (0xAA – 0xAA – 0x03)
OUI (0x00 – 0x00 – 0x00)
PID (0xYY – 0xYY)

Nota: El valor PID es a determinar, esto quiere decir que debe ser diferente a cualquier valor ya utilizado inclusive el PID para el Encapsulamiento Servidor, pero su valor en si mismo no tiene relevancia.

5. Consideraciones Generales del Modelo

El objetivo de la especificación de este modelo, es ofrecer una solución al encapsulamiento de la versión 6 del protocolo de interconexión de redes IP (IPv6) sobre redes ATM. El mismo consiste en transportar datagramas IP sobre la capa AAL5 considerando la aplicación de ATM como reemplazo directo de las tecnologías de nivel 2 conocidas.

Dada la necesidad que tiene IPv6 de contar con una capa inferior que provea un servicio de multicast, y teniendo en cuenta que el paradigma de ATM no lo provee directamente, en esta especificación presentamos una solución para este problema.

Para abstraer a IPv6 del nivel 2 subyacente, este modelo define una nueva capa que va a estar situada entre el nivel IPv6 y la subcapa AAL5 de ATM.

Esta especificación abarca todo lo concerniente a una comunicación entre componentes de una LAN, cuyo alcance involucra el establecimiento de conexiones unicast y multicast, resolución de direcciones y manejo de prioridades.

6. Arquitectura

Como se definió en el objetivo, la solución que proponemos esta acotada específicamente a una red LAN ATM. Esta red esta compuesta por Routers y Host conectados directamente por medio de interfaces ATM a los switches.

Dado que IPv6 trabaja en gran medida con direcciones multicast, es necesario que la capa de nivel 2 provea este servicio. Teniendo en cuenta que el servicio multicast va estar dado por la capa aquí definida, encontramos ventajoso la utilización de la interface UNI 4.0 de ATM, que a diferencia de sus antecesoras provee el establecimiento de circuitos multicast.

Desde el punto de vista de IPv6, los Nodos están organizados bajo el concepto de Link. En esta especificación extendemos el concepto de Link al de Grupo de Vecinos, es decir que un Grupo de Vecinos es la visión lógica de un Link de IPv6. Para cada uno de estos Grupos existe un

Servidor D'Artagnan que provee el servicio multicast a los Nodos miembros de ese Grupo de Vecinos.

Un Link IPv6 es la mínima unidad funcional a la cual se aplica este modelo. Teniendo en cuenta que los Host miembros de un mismo Link se agrupan bajo el concepto de Grupo de Vecinos.

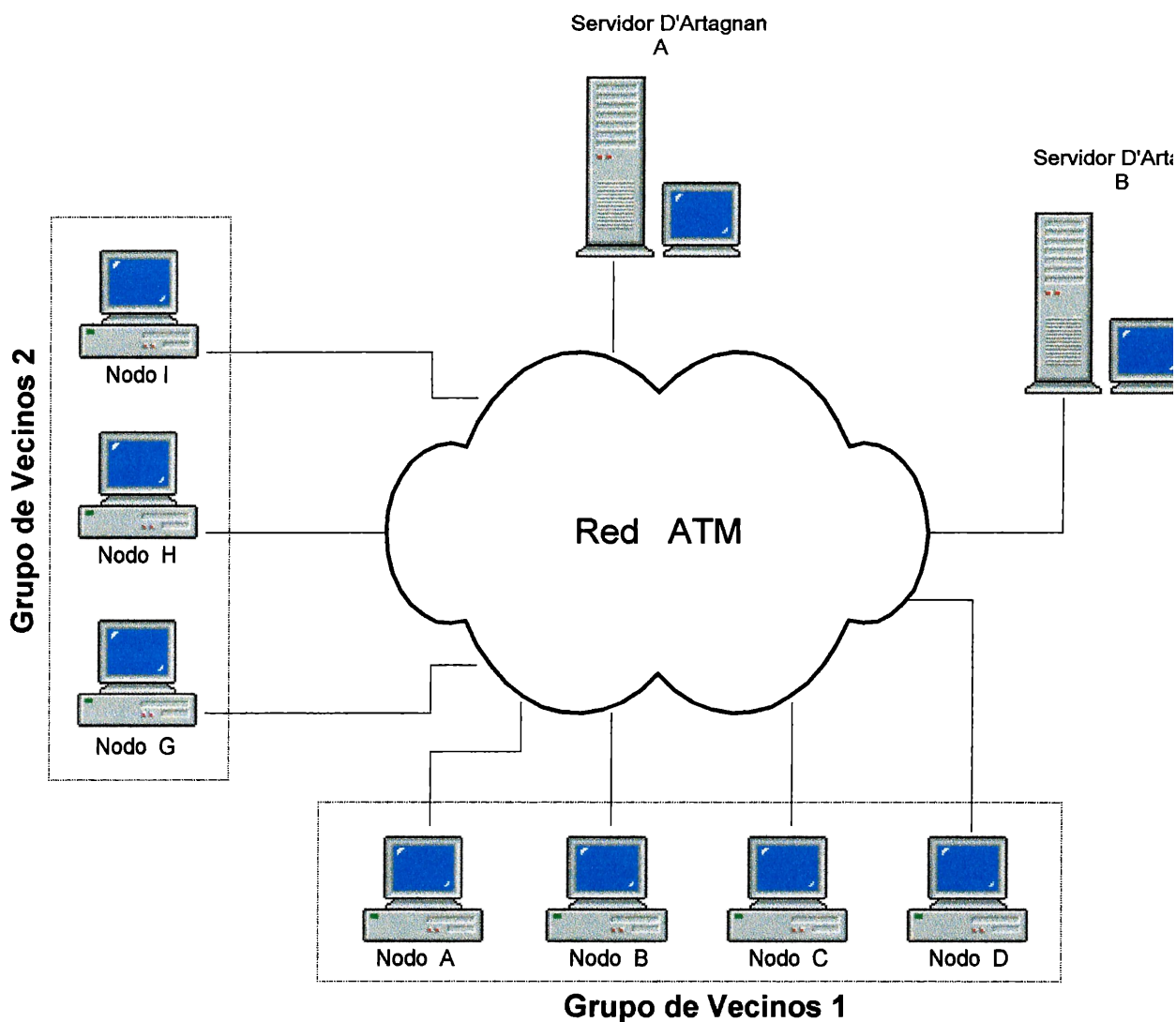


Figura 4.1

En la Figura 4.1 se muestra un ejemplo simple de configuración de la arquitectura propuesta. En el ejemplo podemos ver dos Links o Grupo de Vecinos, un servidor para cada uno de ellos (Servidor D'Artagnan A y Servidor D'Artagnan B), todos estos componentes conectados directamente a los Switches ATM.

7. Grupo de Vecinos

Como mencionamos anteriormente, un Grupo de Vecinos se desprende del concepto de Link definido en la especificación de IPv6. Un Grupo de Vecinos es una organización lógica que se da a un conjunto de Nodos de una LAN ATM teniendo en cuenta que cada uno de ellos es una entidad administrativamente separada, puesto que cada Grupo de Vecinos contiene una configuración independiente del resto de los grupos dentro de la LAN.

Dentro del mismo Grupo de Vecinos los Nodos se comunican directamente entre ellos, quedando la comunicación a un Nodo fuera del grupo por medio de un Router. Como consecuencia directa de la relación uno a uno que existe entre Grupo de Vecinos y Link, los Routers constituyen los únicos elementos que pueden pertenecer a varios Grupo de Vecinos. Notemos que de esta definición se obtiene un paralelismo entre los conceptos de Grupo de Vecinos y el actual concepto de Subred IP.

Cada Grupo de Vecinos se caracteriza por contar con un Servidor D'Artagnan que trabaja cooperativamente con los Host y Routers para resolver determinados problemas en un ambiente orientado a conexión tales como la resolución de direcciones, distribución de paquetes multicast, detección de routers, etc.

8. Modelo del Servicio de Multicast

Recordemos que básicamente un servicio multicast da la posibilidad a un Nodo de enviar un mismo paquete a varios Nodos al mismo tiempo, teniendo como casos particulares el envío de un paquete a un Nodo (Servicio Unicast) y a todos los Nodos (Servicio Broadcast).

Para proveer el Servicio Multicast definimos dos entidades, una cliente y una servidora, que trabajan en forma cooperativa. Estas entidades se encuentran instaladas en unidades operativamente diferentes, las primeras (clientes) se encuentran en los Host y Routers, y las segundas (servidores) en los Servidores D'Artagnan. La Capa Cliente se encuentra ubicada entre la capa de IPv6 y la subcapa AAL5 de ATM, mientras que la Capa Servidora se ubica Sobre AAL5, pero no interactúa con ningún protocolo de nivel superior. En este modelo el servidor puede no ser un Host IP, en cuyo caso solo proveerá conectividad a nivel ATM. Esta conectividad consiste del canal punto-multipunto que el server tendrá con todos los miembros del Grupo de Vecinos correspondiente y de los canales que los Nodos establecerán dinámicamente con el servidor para enviarle los paquetes con dirección destino multicast. Si se quisiera definir al servidor como un Host IP, solo tendríamos que agregarle la capa Cliente; esto es así debido a la independencia funcional que existe entre ambas capas.

El envío de paquetes unicast, ya sea dentro del Grupo de Vecinos o fuera de él se realiza sin la intervención del Servidor D'Artagnan.

La Figura 4.2 muestra la disposición de las capas de este modelo:

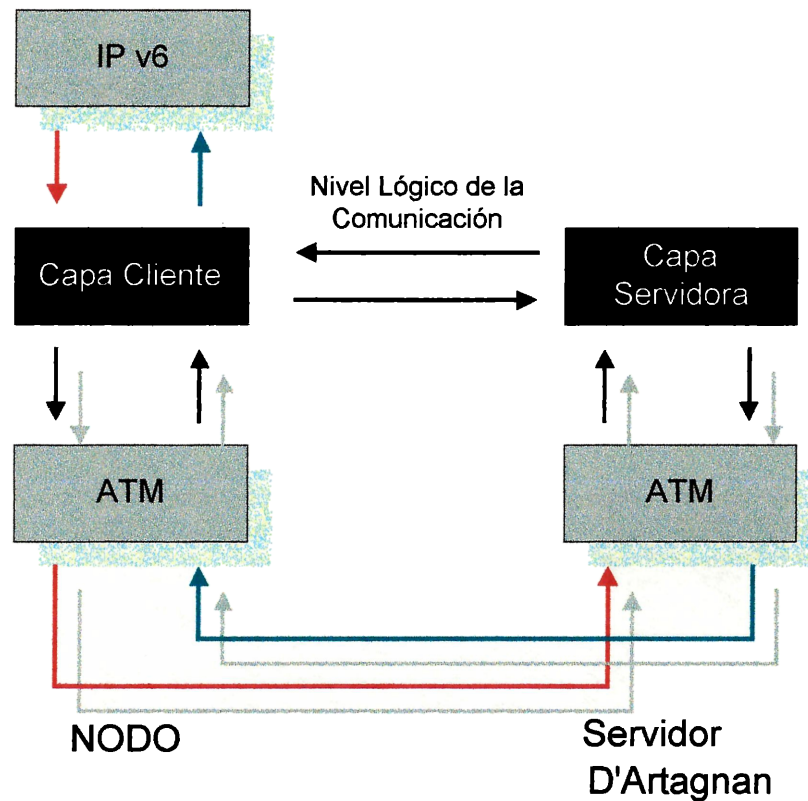


Figura 4.2

8.1. Servidor D'Artagnan

8.1.1 Características del Servidor D'Artagnan

Este servidor es el encargado de conectarse con todos los otros componentes del grupo (Hosts y Routers), a través de un canal punto-multipunto para el cual el servidor actúa como raíz del mismo. Puede existir más de un servidor dentro de un mismo Grupo de Vecinos, esto sería necesario por ejemplo, para el manejo de replicación como veremos más adelante.

La Capa Servidora que se encuentra ejecutando en el Servidor D'Artagnan, tiene la capacidad de recibir los paquetes multicast, "bufferarlos" si es necesario y enviarlos por el canal punto-multipunto que conecta el servidor con todos los componentes del Grupo de Vecinos. Por tal motivo y dado que recibe todos los paquetes multicast enviados por el resto de los miembros del grupo, es importante que cuente con una gran capacidad de buffering para no tener pérdidas en los paquetes a enviar.

Al igual que los clientes este servidor utiliza para conectarse con la red ATM la interface definida en la especificación UNI 4.0.

8.1.2 Descripción de la funcionalidad de la Capa Servidora

Esta Capa es la encargada de recibir los paquetes de los Host o Routers y reenviarlos por el canal multicast (punto-multipunto) al resto de los miembros del grupo. Estos paquetes son recibidos a través de los canales punto a punto que generan estos clientes con el server(se crean dinámicamente según se necesiten). En algunas situaciones esta Capa puede utilizar un buffer definido para almacenar los paquetes entrantes para luego poder enviarlos por el canal multicast sin ocasionar perdida alguna.

En forma mas espaciada esta capa también es la encargada de intercambiar información de replicacion con otra capa residente en un Servidor D'Artagnan diferente.

Notemos que la funcionalidad principal de esta Capa es muy simple, pero resulta de vital importancia para poder proveer al protocolo IPv6 de un servicio multicast.

8.2 Cliente

8.2.1 Características del Cliente

Los clientes IP que operan en una LAN definida en los términos de este modelo deben contar con las siguientes características:

- Todo los clientes están conectados a la red ATM en forma directa.
- Todos los miembros utilizan en la conexión ATM la interface defina en la especificación UNI 4.0.
- Todos los clientes deberán conocer la dirección ATM del Servidor D'Artagnan de multicast antes de comenzar a operar como estaciones IP. La forma propuesta en este modelo es teniendo un parámetro local definido en la estación.
- Todos los clientes de un Grupo de Vecinos deben ser capaces de comunicarse vía ATM entre sí.
- La comunicación entre componentes de diferentes Grupo de Vecinos se hace a través de Routers.
- Un Router puede pertenecer a múltiples Grupos de Vecinos, teniendo para cada uno una configuración diferente, es decir que cada conexión del Router a un Grupos de Vecinos deberá respetar todos los puntos aquí definidos.
- Todos miembros IPv6 utilizan el protocolo Neighbor Discovery para descubrir la presencia de otros miembros, determinar su dirección de link, encontrar Routers y para mantener información de alcanzabilidad acerca de los vecinos activos

como se define en el RFC 2461 (Neighbor Discovery for IP versión 6).

- Todos los miembros utilizan la arquitectura de direccionamiento IP definida en el RFC 2373 (IP versión 6 Addressing Architecture).
- Todos los miembros de la red IPv6 que se están conectando a través de este modelo deben pertenecer a algún Grupo de Vecinos.
- Todos los paquetes enviados por las estaciones IPv6 serán encapsulados utilizando el encapsulamiento LLC/SNAP. Utilizamos este mecanismo para poder identificar que paquetes son para la capa cliente y que paquetes para la capa servidora.

8.2.2 Descripción de la funcionalidad de la Capa Cliente

La funcionalidad de la Capa Cliente esta dada por la detección de que paquetes tienen que ser enviados al Servidor D'Artagnan para que este luego los retransmita al resto de las máquinas y que paquetes pertenecen a una comunicación punto-a-punto.

Esta detección se realiza examinando la dirección ATM destino que IP entrega a la capa cliente de este modelo. Si la dirección es aquella que representa una multicast, entonces la Capa Cliente redirecciona el paquete hacia el Servidor de multicast para luego entregárselo a AAL5 y que este lo haga llegar al Servidor; si no es una dirección multicast, el paquete no tiene ningún tratamiento y es entregado a la capa AAL5 de ATM para que lo envíe por una conexión punto-a-punto con el destino especificado.

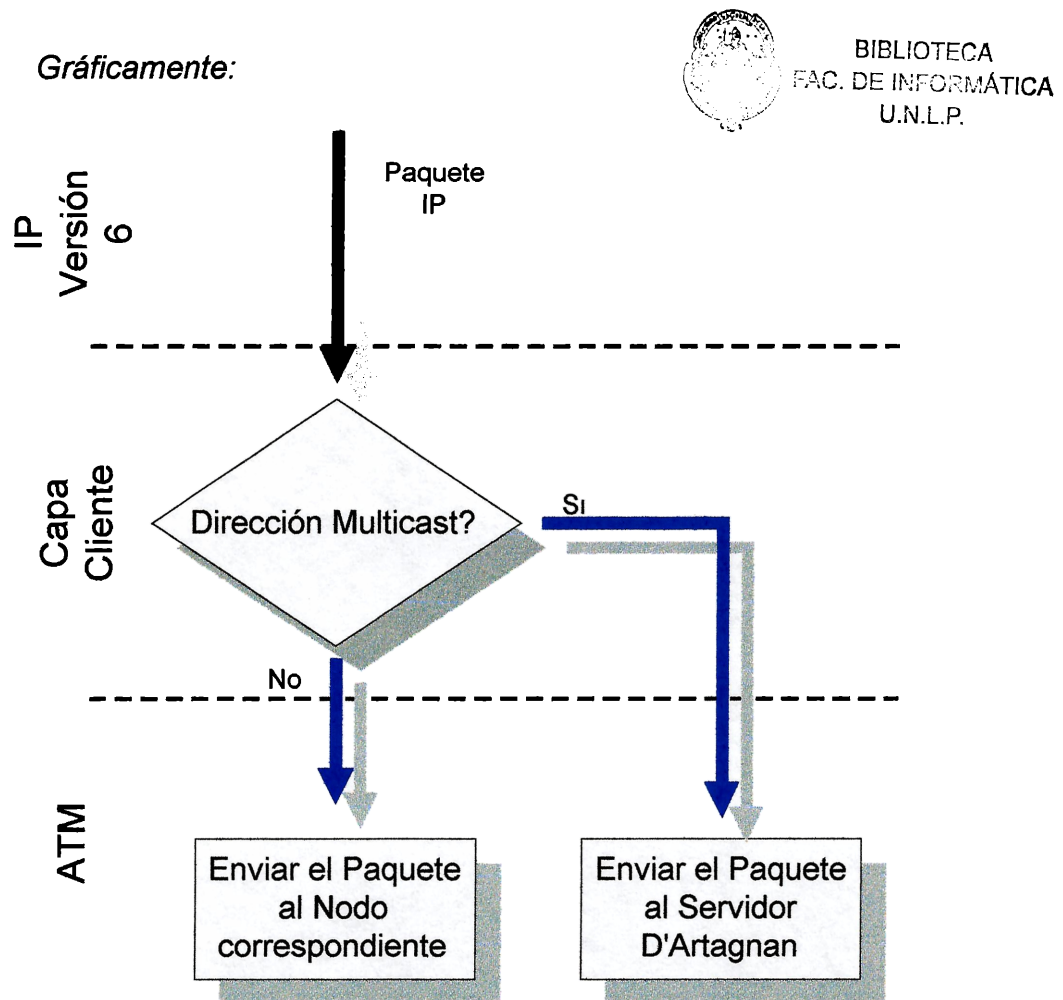


Figura 4.3

Cuando un Nodo comienza a funcionar debe proveerse automáticamente el servicio multicast, para que el protocolo IPv6 pueda utilizar los mecanismos de descubrimiento de vecino, advertencia de vecino, etc., que necesitan de este servicio para operar correctamente.

A nivel ATM, para levantar el servicio multicast la Capa Cliente debe encargarse de pedir al servidor de multicast (raíz) que incorpore al Nodo como una hoja, si la raíz todavía no había levantado el canal multicast, lo realiza, sino solo incorpora al Nodo como un miembro más del Grupo de Vecinos.

Los canales punto-a-punto de los Nodos con el Servidor D'Artagnan son dinámicos, es decir que se establecen siempre y cuando no exista ya un canal con el servidor.

En el momento de recibir los paquetes que provienen de AAL5, la Capa Cliente los retransmite a la capa superior para que esta los examine.

Observemos que el servicio de multicast propuesto establece que un paquete multicast les llegue a todos los miembros de un Grupo de Vecinos (Link IPv6), por lo tanto se deja a nivel IP la responsabilidad de aceptar o descartar los paquetes que a esa capa arriben, dependiendo de la dirección IPv6 que contengan los mismos.

9. Implementación

9.1. Estructuras de Datos

9.1.1 Estructura de Datos para el Cliente:

Todas las estructuras de datos aquí definidas deberán estar presentes con sus correspondientes valores en cada cliente antes de que estos empiecen a funcionar en la red, exceptuando los casos especificados. En caso de que alguna de estas no estuviera presente al momento de comenzar con el funcionamiento del cliente, este deberá abortar su ejecución.

- *DirATMSvr_Tbl:*

Esta tabla contiene las direcciones ATM de los Servidores D'Artagnan disponibles para un Grupo de Vecinos determinado. La disposición de las direcciones dentro de la tabla determina el orden que deben llevar los clientes en el momento de la elección del servidor D'Artagnan a conectarse, es decir primero se intentará conectar al servidor cuya dirección este primero en la tabla, luego con el segundo y así sucesivamente. Además esta tabla contiene la cantidad máxima de intentos de conexión por servidor, es decir cada cliente tendrá la posibilidad de intentar conectarse con cada servidor un numero determinado de veces.

Orden	Dirección ATM	Intentos
1	Dir ATM Svr 1	5
...
N	Dir ATM Svr n	1

- *ConexActivCli_Tbl:*

Esta tabla contiene la información de las conexiones activas del Cliente para el envío de paquetes unicast, la misma se genera dinámicamente dependiendo de las conexiones creadas por el mismo. La información contenida en ella esta compuesta por la dirección IPv6 de un Nodo, el identificador del canal creado entre el cliente y dicho Nodo y la Categoría de QoS con la cual fue creado el canal. Además en esta tabla se cuenta con un timestamp que indica la ultima vez que fue utilizada cada entrada para enviar un paquete, pudiendo de esta manera determinar cuales entradas han estado inactivas después de un determinado periodo de tiempo, para luego de pasado ese tiempo eliminarlas y liberar recursos de la red.

Dir IPv6	Canal Id	Cat. Qos	Tiempo Act.
Dir 1	Canal 1	Cat 1	Time1
...
Dir n	Canal n	Cat n	Time n

- *ConexActivSvr_Tbl:*

Esta tabla contiene la información de las conexiones activas del cliente con los servidores para el envío de paquetes multicast. La misma se genera dinámicamente dependiendo de las conexiones creadas por el mismo con dichos servidores. Esta tabla contiene la dirección ATM de un Servidor D'Artagnan, el identificador del canal creado entre el cliente y dicho servidor y la Categoría de QoS con la cual fue creado el canal. Además en esta tabla cuenta con un timestamp que indica la ultima vez que fue utilizada cada entrada para enviar un paquete, pudiendo de esta manera determinar cuales entradas han estado inactivas después de un determinado periodo de tiempo, para luego de pasado ese tiempo eliminarlas de la misma y liberar recursos de la red.

Dir ATM_Svr	Canal Id	Cat. Qos	Tiempo Act.
Dir ATM 1	Canal 1	Cat 1	Time1
...
Dir ATM n	Canal n	Cat n	Time n

- *MaxLoopsSvr_Var:*

Indica la cantidad máxima de loops que se deberá hacer sobre la tabla de servidores en caso de que ninguno de ellos acepte la conexión requerida por el cliente.

- *MaxTiempo_Var:*

Esta variable define el tiempo de vida de las entradas en las tablas *ConexActivCli_Tbl* y *ConexActivSvr_Tbl*.

- *MaxIntenUni_Var:*

Esta variable define la cantidad máxima de intentos que realiza un Cliente para enviar un paquete Unicast.

9.1.2 Estructuras de Datos para el Servidor:

Todas las estructuras de datos aquí definidas deberán estar presentes con sus correspondientes valores en cada Servidor D'Artagnan antes de que estos empiecen a funcionar, exceptuando los casos especificados. En caso de que alguna de estas no estuviera presente al momento de comenzar con el funcionamiento del Servidor, este deberá abortar su ejecución.

- *GrupoVecinos_Tbl:*

Esta tabla contiene la información acerca de los clientes a los cuales el servidor les brinda el servicio de Multicast. Esta información consta de las direcciones ATM de los clientes a los cuales el Servidor les brinda el servicio y un timestamp que indica la ultima vez que el cliente requirió del servicio Multicast, pudiendo

de esta manera determinar cuales clientes han estado inactivos (posiblemente fuera de funcionamiento) después de un determinado periodo de tiempo, para luego de pasado ese tiempo eliminarlos de la misma.

Dir ATM Cliente	Tiempo de Actualización
Dir ATM Cliente 1	Tiempo 1
...	...
Dir ATM Cliente n	Tiempo n

- *SvrCoop_Tbl:*

Esta tabla contiene las direcciones ATM de los Servidores D'Artagnan que cooperaran para ofrecer el Servicio Multicast. Es utilizada por el sistema de replicacion de datos entre los Servidores D'Artagnan. Además contiene la cantidad máxima de intentos de conexión por servidor.

Orden	Dirección ATM	Intentos
1	Dir ATM Svr 1	5
...
N	Dir ATM Svr n	1

- *TiempoInf_Var:*

Esta variable indica el periodo de tiempo en que los servidores intercambian información.

- *NuevasMaq_Var:*

Esta variable contiene la cantidad tope de maquinas nuevas, una vez que se llega a este tope se deberá intercambiar información con los demás Servidores D'Artagnan.

- *IntentosCon_Var:*

Esta variable indica la cantidad máxima de intentos de conexión de un nuevo nodo a un canal multicast.

- *MaxTiempoGrp_Var:*

Indica el tiempo máximo de vida de las entradas en la tabla de Grupo de Vecinos.

9.2 Algoritmos

9.2.1 Algoritmos relativos al Cliente

9.2.1.1 Introducción al Grupo de Vecinos

Cada vez que el cliente se "levanta" intenta establecer un canal punto a punto con el primer Servidor que figura en la tabla *DirATMSvr_Tbl* de servidores disponibles, en caso de que pueda establecer la conexión envía una solicitud de incorporación al grupo de vecinos, esta solicitud tiene el significado de pedirle al servidor la incorporación del cliente al canal multicast, para esto el cliente debe enviar un mensaje especificando como canal Multicast los últimos cuatro octetos de la dirección ATM del Servidor contactado. Una vez que el server envía la confirmación de que el cliente ha sido incorporado al canal Multicast que conforman el grupo de vecinos, este nodo esta en condiciones de empezar a operar con el protocolo de red IPv6.

En caso de no poder establecer la conexión punto a punto con el server o que el pedido de incorporación haya sido rechazado, se realizan tantas veces como se especifica en la columna *Intentos* de la tabla *DirATMSvr_Tbl* para dicho servidor, teniendo en cuenta que se considera un intento tanto el pedido de conexión punto a punto con el servidor como el pedido de incorporación del nodo al grupo de vecinos. Una vez agotada esta cantidad de intentos, el cliente deberá proceder de la misma forma con el siguiente server de la tabla *DirATMSvr_Tbl*. Asimismo, si no se tuvo éxito con ningún servidor de la lista, se repetirá este proceso tantas veces como se indica en la variable *MaxLoopsSvr_Var*. Si se agotaron todas las posibilidades entonces la Capa Cliente deberá anunciarle a IPv6 mediante un error critico que no cuenta con la funcionalidad necesaria de nivel 2 para poder ejecutarse, quedando el cliente imposibilitado de operar en la red y teniendo que ser revisado por el administrador de la misma para poder determinar el problema.

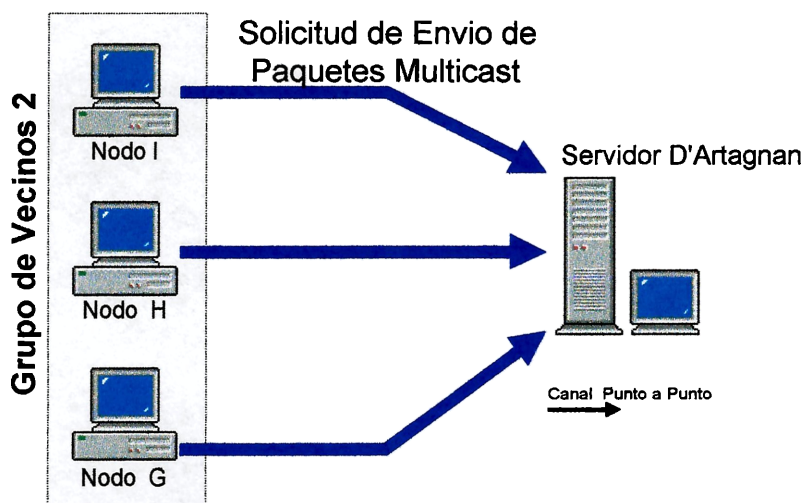


Figura 4.4

9.2.1.2 Envío de un Paquete Multicast

Cuando arriba un paquete proveniente del nivel IPv6, la Capa Cliente debe detectar las prioridades que el mismo acarrea y establecer los atributos de la categoría de QoS asociada a dichas prioridades. Después de este proceso de asociación deberá determinar el tipo de dirección destino que tiene el paquete, en el caso que se trate de una dirección de multicast la Capa Cliente deberá indicarle a ATM que cree una conexión con el Servidor utilizando los QoS determinados anteriormente. Una vez que se establece el canal con el servidor D'Artagnan, la Capa Cliente le envía el paquete para que este lo redistribuya a través del canal Multicast previamente creado. Si por algún motivo la conexión con el servidor no pudo realizarse entonces la misma se reintentará tantas veces como se indique en la tabla *DirATMSvr_Tbl*, en caso de agotarse estas posibilidades se procederá a repetir el proceso con el siguiente server de la tabla, y así sucesivamente hasta completar la lista de servidores. Si no se pudo establecer la conexión con ningún Servidor de la tabla, este proceso se repetirá tantas veces como se indique en la variable *MaxLoopsSvr_Var*. Por ultimo si en las *MaxLoopsSvr_Var* repeticiones el paquete no puede ser enviado entonces será descartado.

9.2.1.3 Envío de un Paquete Unicast

Cuando a la Capa Cliente le arriba un paquete, como fue descrito anteriormente, esta primero determina los QoS asociados dependiendo de la prioridad que acarrea el mismo. Después detecta de que tipo es la dirección destino, en caso que sea multicast se procede como se describió en 9.2.1.2, si es una dirección Unicast la Capa Cliente deberá encargarse de establecer una conexión punto a punto con el destino ATM asociado y con los QoS previamente determinados. En caso de que la conexión con el destino Unicast no se pueda realizar, el Cliente lo intentará tantas veces como se indica en la variable *MaxIntenUni_Var*, una vez agotada esta cantidad de intentos, el paquete será descartado.

9.2.1.4 Determinación de QoS

Tanto para envíos multicast como para envíos unicast la determinación de los QoS del paquete a enviar, se realiza asociando la prioridad de dicho paquete al resultado de buscar como entrada en la tabla de referencia la prioridad para obtener el correspondiente QoS, pero se deja a la negociación a nivel ATM el poder o no respetar estos QoS, esto quiere decir que si a nivel ATM se decide que determinados QoS no se pueden respetar por ciertas circunstancias momentáneas o no de la red, y la conexión se establece con QoS diferentes a los solicitados el envío de los paquetes igualmente se realizara.

PrilIPv6ToQoS: Tabla de asociación de Prioridades Ipv6 – Categorías de QoS

En este modelo definimos una correspondencia “*prioridad IPv6 – Categoría QoS ATM*” . El objetivo de esta asociación es aprovechar las prioridades definidas por IPv6 para el ruteo de paquetes utilizando la correspondencia para establecer canales con las características de ruteo especificadas en el paquete IP. La correspondencia entre Prioridad y Categoría QoS está basada en la característica de la información que esta siendo transportada (por ejemplo:). La siguiente tabla muestra la asociación Prioridad IPv6 – Categoría QoS empleada por este modelo.

Cod Pr.	Prioridad IPv6	Categoría QoS
0	Tráfico no caracterizado	ABR
1	Tráfico “fillers”	ABR
2	Transferencia de datos sin que el destino tenga que estar escuchando (e-mail)	VBR
3	Reservado	Null
4	Transferencia de volumen (ftp)	Nrt-VBR
5	Reservado	Null
6	Tráfico interactivo (telnet)	Nrt-VBR
7	Control de Tráfico internet (protocolos de ruteo)	ABR
8	Paquetes descartables	Dos categorías de QoS provistas por ATM pueden abarcar estas prioridades según lo requiera la característica del tráfico y/o comunicación (CBR -> 8..11 Rt-VBR -> 12..15)
9	“	
10	“	
11	“	
12	“	
13	“	
14	“	
15	Paquetes que no es deseable que sean descartados	

9.2.1.5 Determinacion del tipo de direccion IPv6

Este proceso se dispara cuando se quiere determinar el tipo de dirección destino de un paquete IPv6. Para obtener este tipo verificamos si el primer octeto de la dirección IPv6 tiene el valor FF, si es así se trata de una dirección Multicast en caso contrario se asume que la dirección es Unicast.

9.2.1.6 Mantenimiento de la Tabla *ConexActivCli_Tbl*

Esta tabla se genera dinámicamente con cada establecimiento de conexión para envío de paquetes Unicast. Cada entrada en la misma estará disponible siempre y cuando no permanezca inactiva un periodo determinado de tiempo. Este período de tiempo esta establecido en la variable *MaxTiempo_Var*.

Por cada envío de un paquete Unicast para una misma dirección IP y QoS se actualiza el timestamp correspondiente a esa entrada en la tabla pudiendo así, después de cierto tiempo, determinar que entradas están inactivas.

Una vez establecido un canal con una dirección IP y un QoS determinado, entonces se agrega una entrada a la tabla especificando canal, dirección IPv6, QoS y el timestamp correspondiente. Luego cada vez que venga un paquete que coincida (en dirección IP y QoS) con alguna de las entradas en la tabla entonces no se generara un canal nuevo sino que se reutilizara el canal que se especifica en la entrada ya que es de las características requeridas por el paquete, actualizándose así el timestamp de la misma.

9.2.1.7 Mantenimiento de la Tabla *ConexActivSvr_Tbl*

Esta tabla se genera dinámicamente con cada establecimiento de conexión para envío de paquetes Multicast. Cada entrada en la misma estará disponible siempre y cuando no permanezca inactiva un periodo determinado de tiempo. Este período de tiempo esta establecido en la variable *MaxTiempo_Var*.

Por cada envío de un paquete Multicast para un determinado QoS se toma la primer dirección que figura en la tabla *DirATMSvr_Tbl* y el QoS, se busca si existe una entrada en la tabla que coincida con estos valores. Si existe tal entrada el paquete es enviado por el canal que allí figura, actualizando el timestamp correspondiente. En caso contrario se buscará crear una conexión como se especifico en 9.2.1.2 agregándose una entrada en la tabla con los valores resultantes.

9.2.2 Algoritmos relativos al Servidor

9.2.2.1 Solicitud de incorporación de un Vecino al Grupo

Cuando arriba al Servidor un pedido de incorporación de un Cliente al Grupo de Vecinos, pueden darse dos situaciones; En el primero de los casos el Servidor no tiene creado aun el canal Multicast, por lo tanto con la información enviada por el cliente deberá crear un canal con la capacidad multicast, donde él será la raíz y el Cliente su primer hoja. En el segundo de los casos el canal Multicast ya ha sido creado por otro cliente, por lo tanto el Servidor

lo único que tiene que realizar es agregar una nueva hoja a dicho canal.

En ambos casos si la incorporación del Cliente fue satisfactoria el Server deberá incorporar una nueva entrada a la tabla *GrupoVecinos_Tbl* con la dirección ATM del Cliente y el timestamp correspondiente a la creación de la entrada. Para finalizar éste proceso el Servidor enviará un mensaje de control al Cliente informando el resultado del pedido de incorporación al canal Multicast, es decir si fue satisfactorio o rechazado.

9.2.2.2 Distribución de paquetes Multicast

Cada vez que el Servidor recibe un paquete multicast de algún miembro del Grupo de Vecinos deberá chequear de que se trata efectivamente de un paquete con dirección destino multicast, en caso de ser así busca en la tabla *GrupoVecinos_Tbl* la entrada correspondiente al Cliente originador del paquete y actualiza su timestamp, luego envía dicho paquete por el canal Multicast a los demás miembros del Grupo de Vecinos. En caso de que el Servidor detecte que el paquete no tiene como dirección destino una dirección multicast IP lo descarta.

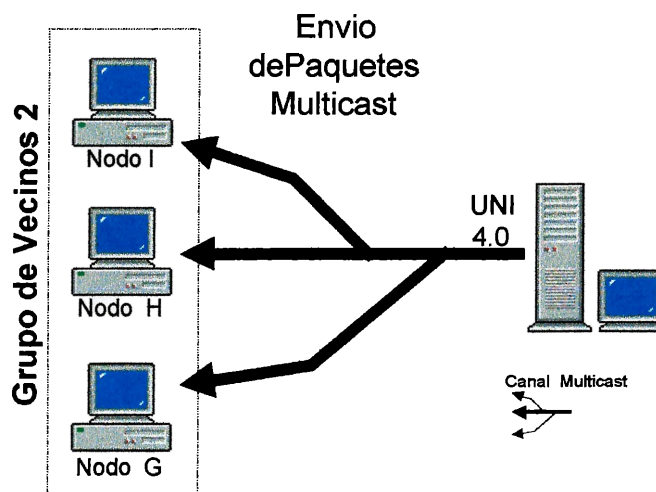


Figura 4.5

9.2.2.3 Mantenimiento de la tabla *GrupoVecinos_Tbl*

Como especificamos anteriormente el timestamp de la tabla *GrupoVecinos_Tbl* se actualiza cada vez que el Servidor D'Artagnan detecta que el vecino está Activo (mediante la recepción de paquetes multicast). Una vez superado el tiempo especificado en la variable *MaxTiempoGrp_Var*, se asume que esa entrada está "vencida" y es eliminada de la tabla. Notemos que la eliminación de una entrada en la tabla *GrupoVecinos_Tbl* implica la eliminación de ese Cliente como hoja del canal Multicast del Servidor D'Artagnan.

9.3 Pseudocódigo

9.3.1 Algoritmos de la Capa Cliente

9.3.1.1 Algoritmo para envío de Paquetes Multicast y Unicast (9.2.1.2 y 9.2.1.3)

PROCEDURE EnviarPaquete (paquete_IP, Dir_ATM)

BEGIN

delivery = false

QoS = Get_QoS (paquete_IP)

tipo_paquete = Get_Tipo_Paquete (paquete_IP)

IF tipo_paquete = *multicast* **THEN**

 cnl = GetCanalActivoSvr (QoS)

IF cnl <> -1 **THEN**

 Enviar (paquete_IP, cnl)

 UpdCnl2TblSvr (cnl, QoS)

 delivery = true

ELSE

 c_loop_tbl = 0

DO WHILE c_loop_tbl < MaxLoopsSvr_Var **AND NOT** delivery

 svr_idx = 1

DO WHILE svr_idx <= UBOUND (DirATMSvr_Tbl) **AND NOT** delivery

 c_intentos = 0

DO WHILE c_intentos < DirATMSvr_Tbl .intentos (svr_idx) **AND NOT** delivery

 cnl = EstablecerCanalSvr (DirATMSvr_Tbl .DireccionATM(svr_idx), QoS)

IF cnl <> -1 **THEN**

 Enviar (paquete_IP, cnl)

 UpdCnl2TblSvr (cnl, QoS)

 delivery = true

END IF

 c_intentos = c_intentos + 1

LOOP

 svr_idx = svr_idx + 1

LOOP

 c_loop_tbl = c_loop_tbl + 1

LOOP

END IF

ELSE /* tipo_paquete = *unicast* */

 c_intentos = 0

 cnl = GetCanalActivoCli (Dir_IP, QoS)

IF cnl <> -1 **THEN**

 Enviar (paquete_IP, cnl)

 UpdCnl2TblCli (Dir_IP, cnl, QoS)

 delivery = true

ELSE

```

DO WHILE c_intentos < MaxIntenUni_Var AND NOT delivery

    cnl = EstablecerCanalCli (Dir_ATM, QoS)
    IF cnl <> -1 THEN
        Enviar (paquete_IP, cnl)
        UpdCnl2TblCli (Dir_IP, cnl, QoS)
        delivery = true
    END IF
    c_intentos = c_intentos + 1
LOOP
END IF
END IF

IF NOT delivery THEN
    Descartar (Paquete IP)
END IF

END

```

9.3.1.2 Algoritmo para incorporarse al Grupo de Vecinos (9.2.1.1)

```

PROCEDURE AddCliente2GrpVecinos()

BEGIN
connected = false
c_loop_tbl = 0

DO WHILE c_loop_tbl < MaxLoopsSvr_Var AND NOT connected

    svr_idx = 1
    DO WHILE svr_idx <= UBOUND (DirATMSvr_Tbl ) AND NOT connected

        c_intentos = 0
        DO WHILE c_intentos < DirATMSvr_Tbl .intentos (svr_idx) AND NOT connected

            cnl = EstablecerCanalSvr (DirATMSvr_Tbl .DireccionATM(svr_idx), Null)
            IF cnl <> -1 THEN
                slt = EnviarSolicitud (cnl)
                IF slt <> -1 THEN
                    connected = true
                END IF
            END IF
            c_intentos = c_intentos + 1

        LOOP
            svr_idx = svr_idx + 1

        LOOP
            c_loop_tbl = c_loop_tbl + 1
        LOOP
            IF NOT connected THEN
                Error ("Conexion Rechazada")
            END IF
        END IF

    END

END

```

9.3.1.3 Algoritmo para determinación de QoS (9.2.1.4)

```
FUNCTION Get_QoS (paquete_IP)
BEGIN
Get_QoS = PrilPv6ToQoS.CategoriaQoS (GetPrioridadIP (paquete_IP))
END
```

9.3.1.4 Algoritmo de Determinacion del tipo de direccion IPV6 (9.2.1.5)

```
FUNCTION Get_Tipo_Paquete (paquete_IP)
BEGIN
address_ip = Get_DestinationAddress (paquete_IP)
IF SubsetStr (address_ip,1,2) = FF THEN
  Get_Tipo_Paquete = multicast
ELSE
  Get_Tipo_Paquete = unicast
END IF
END
```

**9.3.1.5 Algoritmo para actualizar las entradas a la tabla
ConexActivCli_TBL (9.2.1.6)**

```
PROCEDURE UpdCnl2TblCli (Dir_IP, cnl, QoS)
BEGIN
IF Existe_Entrada (cnl) THEN
  ConexActivCli_Tbl (cnl).Timestamp = Now
ELSE
  ConexActivCli_Tbl (new_index).Dir_IPv6 = Dir_IP
  ConexActivCli_Tbl (new_index).canal_Id = cnl
  ConexActivCli_Tbl (new_index).Cat_QoS = QoS
  ConexActivCli_Tbl (new_index).Timestamp = Now
END IF
END
```

**9.3.1.6 Algoritmo para actualizar las entradas a la tabla
ConexActivSvr_TBL (9.2.1.7)**

```
PROCEDURE UpdCnl2TblSvr (Dir_ATM, cnl, QoS)
BEGIN
IF Existe_Entrada (cnl) THEN
  ConexActivSvr_Tbl (cnl).Timestamp = Now
ELSE
  ConexActivSvr_Tbl (new_index).Dir_ATM = Dir_ATM
  ConexActivSvr_Tbl (new_index).canal_Id = cnl
  ConexActivSvr_Tbl (new_index).Cat_QoS = QoS
  ConexActivSvr_Tbl (new_index).Timestamp = Now
END IF
END
```


9.3.1.7 Algoritmo para eliminar entradas vencidas en las tablas de la Capa Cliente

```

PROCEDURE TblCli_EntriesCleaner

BEGIN

  /* Eliminacion de entradas en ConexActivSvr_Tbl */
  FOR i = 1 TO UBOUND (ConexActivSvr_Tbl)
    IF ConexActivSvr_Tbl(i).timestamp + MaxTiempo_Var < Now THEN
      DeleteEntry( i )
    END IF
  NEXT i

  /* Eliminacion de entradas en ConexActivCli_Tbl */
  FOR i = 1 TO UBOUND (ConexActivCli_Tbl)
    IF ConexActivCli_Tbl(i).timestamp + MaxTiempo_Var < Now THEN
      DeleteEntry( i )
    END IF
  NEXT i

END

```

9.3.2 Algoritmos de la Capa Servidora

9.3.2.1 Algoritmo para incorporar un Vecinos al Grupo (9.2.2.1)

```

PROCEDURE AddVecino(DirATMCli,cnlMulticast)

BEGIN

  IF cnlMulticastAct THEN
    st = addHoja(DirATMCli,cnlMulticast)
  ELSE
    st = CrearCanalMulticast(DirATMCli,cnlMulticast)
  END IF
  IF st <> -1 THEN
    UpdGrpVecTbl (DirATMCli)
    CantNuevasMaq_Var = CantNuevasMaq_Var + 1
    IF NuevasMaq_Var = CantNuevasMaq_Var THEN
      SendConectInfo
      CantNuevasMaq_Var = 0
    END IF
  END IF

END

```

9.3.2.2 Algoritmo para actualizar las entradas a la tabla *GrupoVecinos_Tbl* (9.2.2.3)

```
PROCEDURE UpdGrpVecTbl (DirATMCli)
BEGIN
IF Existe_Entrada (DirATMCli) THEN
  GrupoVecinos_Tbl (DirATMCli).Timestamp = Now
ELSE
  GrupoVecinos_Tbl(new_index).DirATM = DirATMCli
  GrupoVecinos_Tbl (new_index).Timestamp = Now
END IF
END
```

9.3.2.3 Algoritmo para la distribucion de paquetes Multicast (9.2.2.2)

```
PROCEDURE MultiDelivery (paquete_IP,DirATMCli)
BEGIN
IF Get_Tipo_Paquete (paquete_IP) = multicast THEN
  UpdGrpVecTbl (DirATMCli)
  Enviar (paquete_IP,cnlMulticast)
ELSE
  Descartar (Paquete IP)
END IF
END
```

9.3.2.4 Algoritmo para eliminar entradas vencidas en la tabla *GrupoVecinos_Tbl*

```
PROCEDURE TblSvr_EntriesCleaner
BEGIN
FOR i = 1 TO UBOUND (GrupoVecinos_Tbl)
  IF GrupoVecinos_Tbl (i).timestamp + MaxTiempoGrp_Var < Now THEN
    DeleteEntry( i )
  END IF
NEXT I
END
```

10. Seguridad

Un punto importante de esta arquitectura es que la seguridad de IP continua trabajando sin ninguna modificación. Todos los mecanismos de seguridad son mantenidos incluyendo la autenticación en detección de vecinos y routers. Esto es, los nodos solicitados pueden elegir responder a una solicitud basados en la información de autenticación contenida en el paquete solicitante. No se requiere ningún cambio para IPv6 ni ningún protocolo de seguridad de nivel superior, por lo tanto no hay pérdida de la misma.

11. Replicación

Observemos que la arquitectura planteada, define un sistema para enfrentar posibles caídas del Servidor que mantiene la conexión punto-multipunto. Este sistema trata de darle mayor robustez al modelo definido para no perder el servicio de multicast que se le ofrece a IPv6. Para otorgar seguridad al modelo definido, especificamos una solución que se basa en la utilización de dos servidores (primario y secundario) y en la replicación de los datos de la conexión que mantienen ambos Servidores con los integrantes del Grupo de Vecinos.

11.1 Intercambio de datos entre Servidores D'Artagnan

El momento en que los Servidores intercambian información, esta sujeto al análisis de dos variables, una que depende del tiempo *TiempoInf_Var* y otra que depende de la cantidad de entradas nuevas en la tabla de Vecinos *NuevasMaq_Var*. Dependiendo de que suceda primero, si se cumple el tiempo especificado o si se llega a la cantidad máxima de entradas nuevas, se envía al otro servidor mediante un mensaje de control la información contenida en la tabla de Vecinos *GrupoVecinos_Tbl*.

La forma en que los servidores procesan la información que intercambian es la siguiente:

Cada vez que un Servidor recibe información de replicación, éste intenta agregar al canal multicast a aquellos Nodos que no tiene en su tabla. Si los pudo agregar al canal multicast los inserta en la tabla de Vecinos, si no pudo realizar las incorporaciones luego de los n-intentos como se especifica en la variable *IntentosCon_Var*, los descarta y no agrega nuevas entradas en dicha tabla.

Para aquellos Clientes que vienen en la información de replicación y que existen en la tabla se procede a actualizar el campo "*tiempo de actualización*" si el tiempo que viene es superior al especificado en la tabla.

El Servidor D'Artagnan también se encarga de mantener actualizada la tabla de vecinos eliminando aquellas entradas que no han sido utilizadas después de un tiempo determinado por la variable *MaxTiempoGrpVar*. Esta validación se hace luego del intercambio de

información entre los servidores, esto es así dado que una entrada que podría llegar a eliminar (puesto que esta "vencida") puede venir con un tiempo mas nuevo, con lo cual no tendría que bajarse la conexión con ese cliente.

11.2 Sincronización entre los Nodos y el Servidor

Mientras el Servidor primario esta disponible, los nodos envían los paquetes multicast normalmente, de acuerdo a lo especificado en la descripción del modelo, cuando el Servidor Primario deja de atender las solicitudes de los Nodos, éstos toman la dirección del Servidor Secundario y comienzan a enviarle los paquetes multicast. Esto permite que cuando el Servidor Secundario deje de funcionar, automáticamente se intente operar con el Servidor Primario.

Cabe destacar que la solución planteada, esta dada a nivel de definición, quedando la implementación de la misma sujeta a futuros estudios.

11.3. Algoritmos para la Replicación de Datos entre Servidores D'Artagnan

PROCEDURE SendConectInfo

BEGIN

PqCtrl = Armar_Paquete_de_Control (GrupoVecinos_Tbl)

cnl = EstablecerCanalSvr (DirATMSvr_Tbl .DireccionATM(svr_idx),Null)

IF cnl <> -1 **THEN**

 Enviar (PqCtrl, cnl)

END IF

END

POCEDURE ReceiveConectInfo (PqCtrl)

BEGIN

GrpTbl = Obtener_Tabla_de_Vecinos(PqCtrl)

FOR I = 1 **TO** UBOUND(GrpTbl)

 J= Exist (GrpTbl.DirATMCli(i), GrupoVecinos_Tbl)

IF j = -1 **THEN**

 AddVecino(GrpTbl.DirATMCli(i),cnlMulticast)

ELSE

IF GrupoVecinos_Tbl (j).timestamp < GrpTbl.timestamp(i) **THEN**

 GrupoVecinos_Tbl (j).timestamp= GrpTbl.timestamp(i)

END IF

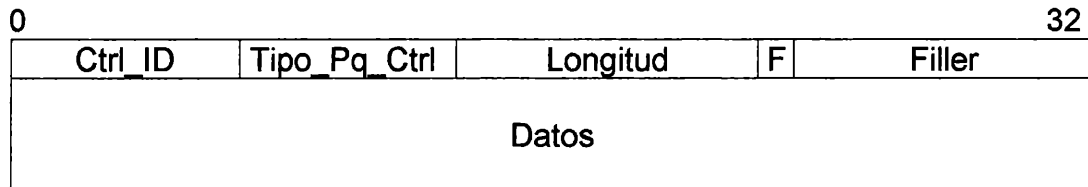
END IF

NEXT I

END

12. Mensajes de Control

Los mensajes de control son utilizados para intercambiar o enviar información interna del modelo definido para el encapsulamiento de IPv6 sobre redes ATM. La información de control es utilizada en la Solicitud de ingreso de un Nodo a un Grupo de Vecinos, en el Rechazo o Aceptación de la incorporación y en la Replicación de Información entre Servidores D'Artagnan. Para este tipo de intercambio de información definimos un paquete que tiene la siguiente estructura:



Ctrl ID (6 bits): es un identificador de paquetes de control, debe ser diferente a la versión de IP que acarrea un paquete IPv6

Tipo Pq Ctrl (6 bits): este campo identifica el tipo de paquete de control que se está transportando. Los tipos disponibles para esta especificación son:

1. Pedido de Ingreso al Grupo de Vecinos.
2. Rechazo de la Incorporación.
3. Aceptación de la Incorporación.
4. Información de Replicación.

Longitud (8 bits): este campo indica la longitud del paquete expresada en octetos

F: este flag indica si el paquete que está siendo transportado es un fragmento de un paquete.

1. El paquete es un fragmento.
0. El paquete es el fin de un paquete fragmentado o es un paquete sin fragmentar.

Filler: este campo es el "relleno" del paquete hasta completar un paquete de 1280 octetos de longitud.

Datos: el campo de datos acarrea información de control para los tipos de paquetes 1 y 4 (pedido de incorporación al grupo de vecinos e información de replicación respectivamente). A continuación se detalla la información que este campo transporta para cada tipo de paquete:

Pedido de Ingreso al Grupo de Vecinos

Dirección ATM del Nodo solicitante
Canal Multicast

Información de Replicación

Tabla de Grupo de Vecinos (Dirección ATM del Cliente, timestamp)

Conclusiones

En esta Tesis especificamos una solución que permite encapsular IPv6 sobre Redes ATM. El principal inconveniente que se presenta al atacar el encapsulamiento es que IPv6 es un protocolo “sin conexión”, mientras que ATM esta “orientado a conexión”. Este punto nos llevó a trabajar en una solución que ofreciera el servicio de broadcast, característico de los protocolos no orientados a conexión, de una manera transparente para que encapsular no implique la redefinición de ninguno de los protocolos involucrados.

Esta tesis presenta una solución al problema antes mencionado. Dicha solución esta basada en la utilización de un modelo Cliente – Servidor para realizar los servicios de distribución broadcast. El modelo se compone de dos piezas de software, una que cumple el rol de Cliente que reside en los Hosts y Routers, y otra que cumple el rol de Servidor que reside en el denominado Servidor D’Artagnan. A cada una de ellas se las denomina Capa Cliente y Capa Servidora respectivamente.

La Capa Cliente se encuentra ubicada en un nivel intermedio entre el nivel ATM y el protocolo IPv6, mientras que la Servidora se encuentra sobre ATM. Ambas capas interactúan entre si para ofrecer todos los servicios que requiere IPv6 para poder transportar paquetes unicast y multicast y para la utilización de protocolos tales como Neighbor Discovery.

La función de la Capa Cliente esta dada por el envío de los mensajes con destino multicast al Servidor D’Artagnan correspondiente, en tanto que la funcionalidad de la Capa Servidora esta dada por la distribución de éstos paquetes multicast.

El hecho de ofrecer un modelo en capas para resolver el encapsulamiento, nos permitió abstraer a IPv6 del protocolo de nivel dos subyacente (ATM en este caso).

El punto saliente en el desarrollo de esta Tesis fue proveer el servicio multicast, tan fuertemente ligado a la funcionalidad de IPv6. Para éste problema analizamos las Interfaces UNI 3.1 y UNI 4.0, optamos por la utilización de la versión más actual (4.0) puesto que ofrece la posibilidad de crear canales multicast, lo cual permite simplificar tanto los algoritmos como así también las estructuras definidas en este modelo.

En conclusión, el modelo definido en este Trabajo de Tesis utiliza un modelo en capas para abstraer a IPv6 del nivel dos, un modelo Cliente – Servidor que conjuntamente con el uso de la interface UNI 4.0 facilitan el mecanismo de proveer el servicio multicast a IPv6. Todas estas características (y las definidas con detalle en el Capítulo 4: “Encapsulamiento de IPv6 sobre Redes ATM”) nos permiten satisfacer los objetivos planteados en el comienzo de ésta Tesis.

A partir de ésta Tesis se podrían realizar futuros trabajos como la implementación de éste modelo en un entorno determinado, el estudio de factibilidad de la extensión del modelo hacia redes WAN, la posibilidad de encapsular, utilizando el modelo aquí definido, otros protocolos no orientados a conexión sobre otras redes orientadas a conexión, el estudio de la autenticación de los clientes que desean ingresar a un Grupo de Vecinos, como así también todas las investigaciones que surjan a partir de necesidades o motivaciones diferentes que tomen a este modelo como punto de partida, teniendo en cuenta que existen otros modelos tales como los especificados en el RFC 2492 o en el documento "A Framework for IPv6 over ATM", que persiguen el mismo objetivo con diferentes soluciones y podrían alimentar también a éstas futuras propuestas.

A P E N D I C E A

Encapsulamiento de IPv4 sobre ATM

El objetivo de la definición de este apéndice es proponer una solución para el encapsulamiento de la versión actual de IP (IPv4) sobre Redes ATM. Esta sección persigue las mismas premisas que el encapsulamiento para IPv6 descrito en el Capítulo 4.

Esta definición, adopta la arquitectura especificada en Classical IP para IPv4, en la que se presenta el concepto de LIS.

1. Configuración de las Subredes IP

Puesto que no existe una propuesta para resolver de una manera general el binding de direcciones IP a direcciones ATM en grandes redes, se atacó el problema para redes más acotadas (LAN ATM Privadas).

2. Concepto de Subredes Lógicas a nivel IP (LIS)

La idea se basa en un grupo de computadoras conectadas a una LAN ATM, este grupo forma un Logical IP Subnet (LIS). Múltiples LIS se pueden definir sobre una misma LAN ATM. Cada LIS funciona como una LAN separada, es decir que en un escenario LIS cada entidad administrativamente separada configura sus Host y Routers dentro de una Subred Logic IP cerrada.

3. Arquitectura de LIS

Una LIS consta de un grupo de Host conectados a una LAN ATM, un Router para interconectar las distintas LIS y servicios de broadcast/ARP.

En la Figura A.1 los Hosts A, C, D, E componen una LIS conjuntamente con F (servidor de broadcast para esta LIS) y G (Router de interconexión de LIS) y los Hosts B, H componen otra LIS con el servidor de broadcast I.

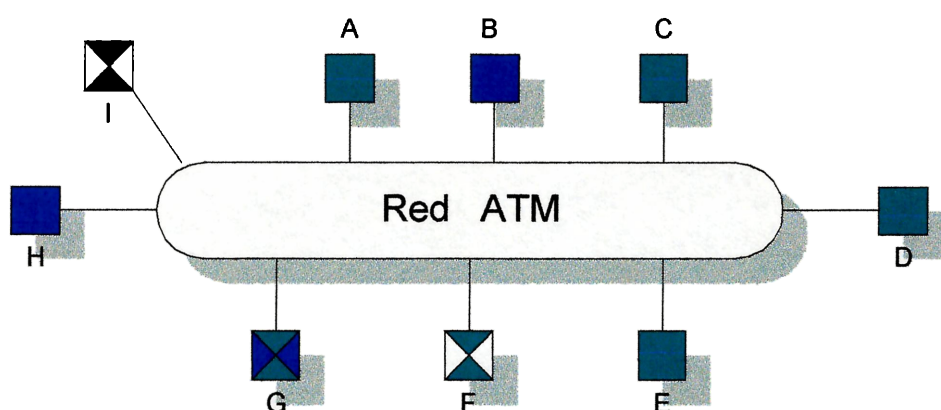


Figura A.1

4. Consideraciones Generales del Modelo

Los Host conectados a ATM se conectan directamente a otros Host en la misma LIS. La comunicación a Hosts fuera de la LIS es provista por un Router, este Router es un "punto final ATM" que es configurado como miembro de una o más LIS pudiendo resultar la misma en un número igual a la cantidad de LIS disjuntas que operan sobre la misma red ATM. Los Host de diferentes Subredes IP (LIS) deben comunicarse vía un Router IP, aún cuando puedan establecer directamente un VC ATM entre ellos.

Los requerimientos para los miembros IP que operan sobre una misma LIS:

- Todos los miembros tienen el mismo número de Red/Subred IP y la misma máscara.
- Todos los miembros de una LIS están conectados directamente a la red ATM.
- Todos los miembros deben ser capaces de comunicarse vía ATM a todos los miembros de la LIS.
- Todos los miembros deben tener un PVC definido al Servidor de Broadcast/ARP.

6. Servicio de Broadcast

Dada la naturaleza "orientada a conexión" del paradigma ATM, los mensajes de broadcast del nivel IP no pueden ser implementados directamente. Desafortunadamente, el estándar de AAL5 no provee una forma para que el receptor identifique celdas individuales de diferentes orígenes. Esto no permite un apropiado reensamble de las celdas llegadas en los frames. La solución a este problema es la implementación de un servidor de Broadcast.

En esta especificación presentamos una implementación del manejo del broadcast a través de un servidor.

6.1. Características del Servidor de Broadcast

Para atender las solicitudes de broadcast de los Host ATM, el servidor debe contar con:

- Un PVC con cada uno de los endsystems ATM de la LIS.
- Una tabla que contenga la asociación de direcciones IP a PVC. Esta tabla se genera dinámicamente, a través de los mensajes de broadcast que envía los Host para hacerse conocer cuando se “prenden”. Las entradas de la tabla se mantienen, en principio, durante un tiempo indeterminado, puesto que la información del binding dirección IP-PVC no es muy cambiante en este tipo de LAN's
- Servicio capaz de reconocer los paquetes de broadcast que están arribando al servidor, para reenviar al resto de los Host.

6.2. Mecanismo para enviar mensajes de broadcast

Cuando IP envía un mensaje de broadcast, la capa de encapsulamiento IPv4-ATM reconoce que se trata de un mensaje de broadcast y lo envía al servidor por el PVC establecido con éste.

El servicio que reside en el servidor, recibe el mensaje, lo analiza y en caso de ser un mensaje de broadcast lo reenvía hacia el resto de los Host de la LIS y actualiza la cache en la que mantiene la asociación de direcciones IP-PVC. Actualizar la tabla implica verificar que no existe una entrada para el par dirección origen del mensaje-PVC, en cuyo caso se crea una entrada en la cache para este par. Por otro lado se verifica que no exista alguna entrada para esa dirección IP con otro PVC o viceversa, en estos casos las entradas se consideran inconsistentes y deben ser eliminadas de la tabla.

7. Servicio de ARP

La idea de esta especificación fue mantener el espíritu que tiene el protocolo ARP en las implementaciones de IP sobre protocolos de nivel 2 no orientados a conexión (Ethernet). A diferencia de otras implementaciones, en este caso la dirección no se resuelve en el servidor, sino que retorna el Host correspondiente.

7.1. Características del Servidor de ARP

Las características del Servidor de ARP son análogas a las definidas para Servidor de broadcast.

7.2. Mecanismo para resolver direcciones (ARP, InARP)

Cuando un Host quiere enviar un paquete y no conoce la dirección ATM del destino, se dispara un ARP Request, el cliente del servicio ARP reconoce que se trata de un paquete de ARP y lo envía al servidor a través del PVC.

El Servidor de ARP recibe el paquete y determina que se trata de un ARP request, y lo reenvía al resto de los Host.

El Host que se reconoce la dirección IP como propia, responde con un ARP reply que es interpretado por el cliente ARP como un paquete de ARP y lo envía al servidor por el PVC. El Servidor recibe esta respuesta, la analiza y reconoce que se trata de un ARP Reply, e interpreta que debe enviar este paquete por el PVC que se corresponda con la dirección destino del paquete ARP Reply en la tabla de direcciones del servidor.

En el caso de InARP, el procedimiento es análogo al ARP.

7.3. Formato del paquete ARP

0	16	31
Hardware Type		Protocolo Type
HLEN	PLEN	Operation
Sender HA octetos		0-3
“ ”		4-7
“ ”		8-11
“ ”		12-15
“ ”		16-19
Sender IP octetos		0-3
“ ”		4-7
“ ”		8-11
“ ”		12-15
Target HA octetos		0-3
“ ”		4-7
“ ”		8-11
“ ”		12-15
“ ”		16-19
Target IP octetos		0-3
“ ”		4-7
“ ”		8-11
“ ”		12-15

Hardware Type: tipo de Tecnologia (Ej. Ethernet = 1)

Protocol Type: 0800 h = direccion IP

Operation: 1 = ARP Request

2 = ARP Reply

3 = ARP Request

4 = ARP Reply

La Figura A.1 muestra el comportamiento de un mensaje de broadcast:

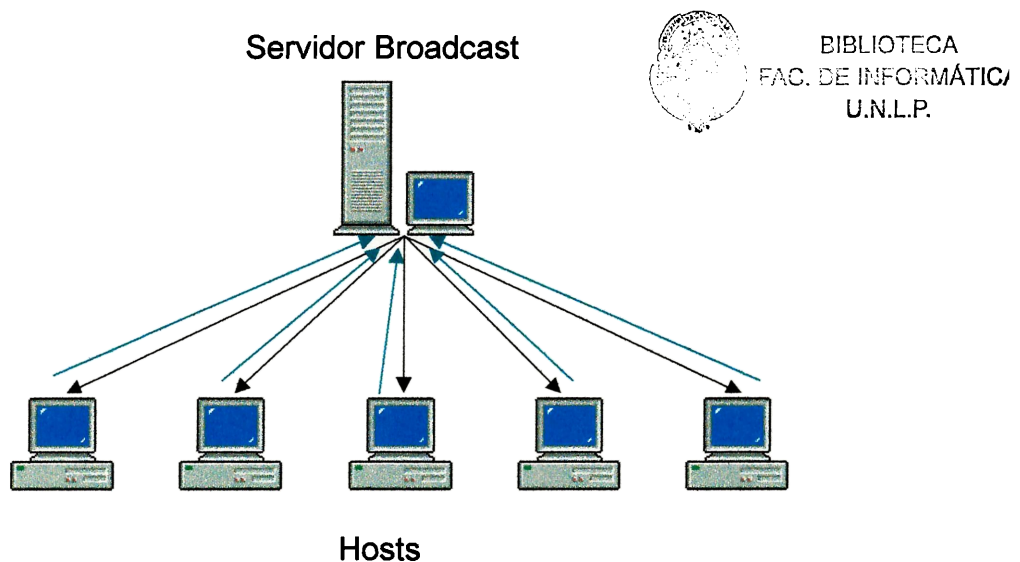


Figura A.1

8. Implementación de los Servicios de Broadcast y ARP

Básicamente la implementación de estos servicios consta de una parte Cliente, que reside en todos los miembros de las LIS, y una parte Servidora.

8.1. Cliente

Como se muestra en la Figura A.2, el Cliente se puede dividir funcionalmente en dos subcapas:

La primera subcapa denominada CLASIFICADOR analiza el tipo de los paquetes distinguiendo entre ARP, Broadcast y otros. Mediante esta distinción en el caso del envío de un mensaje se le indica a la segunda subcapa, CARTERO, a quién le debe enviar (Servidor o Host), en el caso de la recepción de un mensaje, el Clasificador, analiza el tipo de paquete y lo pasa al protocolo de capa superior correspondiente (IP, ARP).

La segunda subcapa, CARTERO, es la encargada de entregar los paquetes a quién corresponda (Servidor o Host) y además es la que interactúa con ATM.

8.2. Servidor

En el Servidor la subcapa Clasificador, analiza el tipo de paquete y además se encarga de consultar al Servidor de ARP y Broadcast cuales son los receptores del mensaje. Con esta información el Clasificador le indica al Cartero a quién/es debe enviar el mensaje.

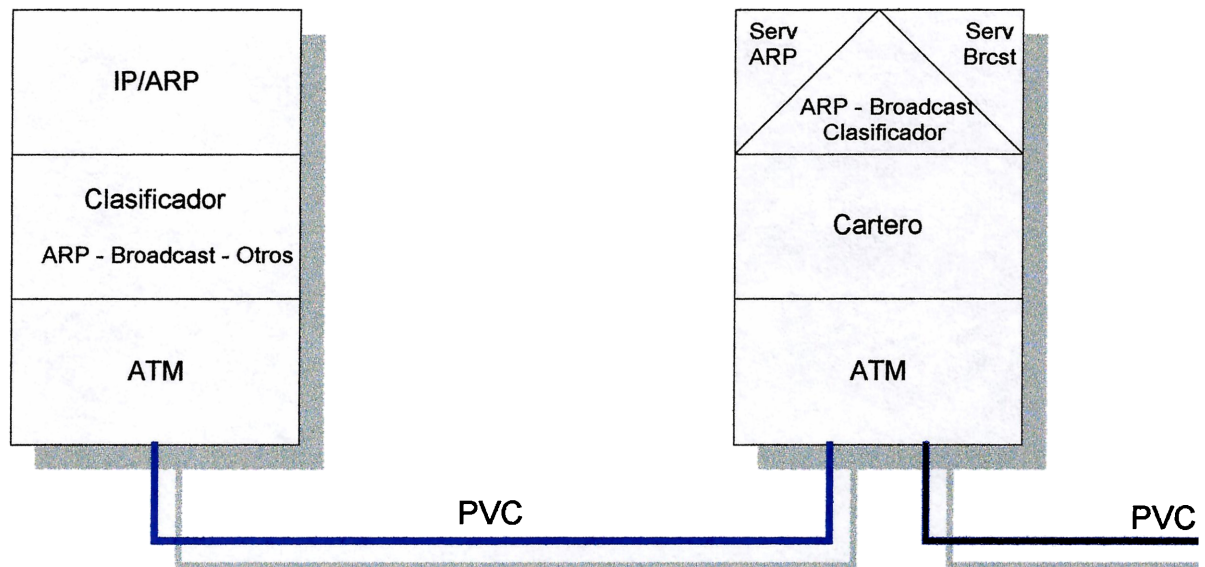


Figura A.2

A P E N D I C E B

Análisis comparativo de los Modelos para IPv4 e IPv6

Esta sección analiza algunas de las diferencias entre ambos modelos, el propuesto en el Capítulo 4, "Encapsulamiento de IPv6 sobre Redes ATM" (E – IPv6), y el comentado en el Apéndice A, "Encapsulamiento de IPv4 sobre Redes ATM" (E – IPv4).

Como punto de partida para analizar las diferencias entre ambos métodos tendríamos que subrayar las diferencias que existen entre IPv4 e IPv6 a la hora de resolver direcciones. En IPv4 las funciones de resolución y configuración de direcciones están provistas por protocolos que trabajan cooperativamente con dicha versión. Estos protocolos, en particular el ARP (Address Resolution Protocol), se ven como un conjunto de funciones que no pertenecen directamente a IPv4 y que están especialmente desarrollados para un nivel 2 determinado. Esto no sucede con el conjunto de protocolos definidos en la nueva versión de IP, IPv6, ya que las funciones de configuración y resolución de direcciones se encuentran embebidas dentro del mismo, teniendo de esta forma una independencia absoluta del nivel dos subyacente, pero a su vez requiriendo de éste un conjunto de servicios que no todas las posibles implementaciones de un nivel 2 pueden ofrecer.

Gráficamente sería:

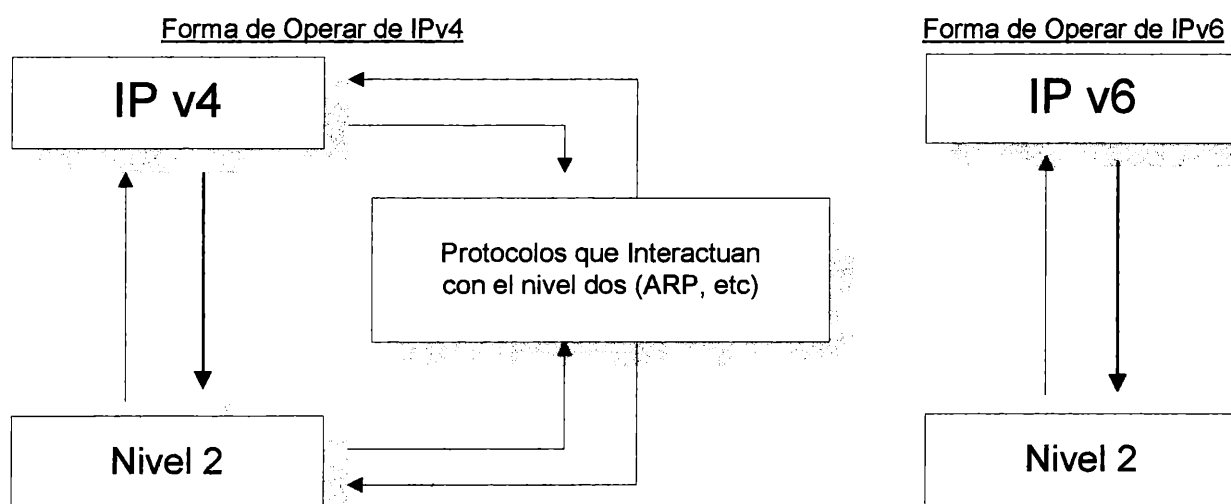


Figura B.1

La diferencia planteada anteriormente indica dónde se debe enfocar el estudio de una solución al problema del encapsulamiento. En el primer caso el punto de enfoque está centrado en proveer el servicio de Broadcast para el mecanismo de resolución de direcciones ARP, es decir, en desarrollar una versión de éste que pueda trabajar sobre el nivel 2 deseado (ATM en este caso); mientras que en el segundo caso está centrado en satisfacer los requerimientos de funcionalidad que necesita IPv6 de su nivel inmediato inferior para poder funcionar correctamente.

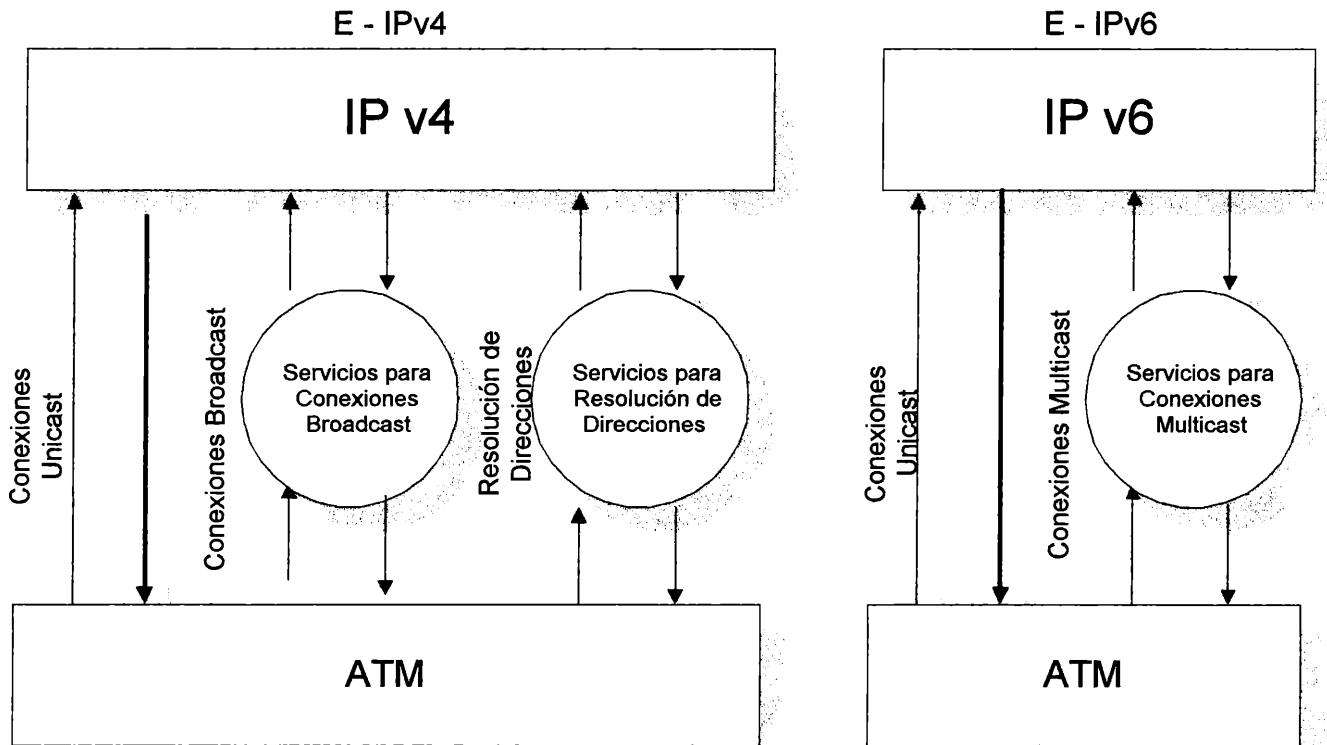


Figura B.2

La figura B.1 muestra precisamente la diferencia que existe entre ambos modelos al momento de interactuar con el nivel 2, en el primer caso E – IPv4, se pueden observar dos servicios: Servicio para Conexiones Broadcast y Servicio para Resolución de Direcciones. Mientras que en el segundo caso, E – IPv6, solo se provee el servicio para Conexiones Multicast.

En una primera instancia, la diferencia planteada para la solución al problema del encapsulamiento nos lleva a pensar que las implementaciones para ambos modelos son totalmente diferentes. Sin embargo, veremos que las arquitecturas especificadas tanto para IPv6 como para IPv4 son similares. Esta similitud consiste en que las implementaciones se basan en un servidor que ofrece determinados servicios, para el caso de encapsulamiento de IPv4 se definieron los servicios de ARP (para resolución de direcciones), y Broadcast, mientras que en el encapsulamiento de IPv6 se especificó un servicio de Multicast.

A pesar de esta similitud, las capas definidas en los servidores de cada una de las soluciones trabajan de manera completamente distinta. En E-IPv4, la capa servidora debe determinar el tipo de paquete que arriba al servidor y a partir de este análisis debe disparar los procesos para los servicios de ARP o Broadcast. Por otro lado en E-IPv6 la capa servidora esta planteada de una forma más simple puesto que no debe hacer ningún tipo de análisis sobre el tipo de paquete sino que sólo se encarga de atender las solicitudes de envío de paquetes multicast.

Asimismo, desde el punto de vista del cliente, la diferencia radica en el análisis que hacen ambas capas (Capa Cliente E-IPv4, Capa Cliente IPv6) sobre el tipo de paquete. En E-IPv4 se distingue entre tres tipos de paquetes, ARP, Broadcast y otros, mientras que en E-IPv6 se distingue entre Multicast y otros.

Cabe destacar que existe una diferencia entre las versiones UNI de ATM en las cuales se sostienen los modelos aquí presentados. Para la solución E-IPv4 se utilizó UNI 3.1 porque el espíritu de esta solución fue presentar una alternativa de encapsulamiento con los medios más usados en la actualidad. Por otro lado en E-IPv6 se buscó una solución que utilizara las herramientas en sus versiones más avanzadas, por esto se utilizó UNI 4.0 la cual provee canales Multicast.

Bibliografía

- **ATM Internetworking. Antony Alles (Cisco Systems Inc.):** Revisión de la infraestructura del Protocolo ATM. Teniendo en cuenta las características propias del protocolo como también aquellas que surgen de su integración con otras tecnologías. Mayo 1995.
- **ATM LAN Emulation – An inside Look at Version 1.0 of the LANE Specification. Bob Klessig (3Com):** Definición de los componentes claves para la implementación de LAN Emulation sobre ATM. Septiembre 1997.
- **ATM – Overview. Cisco Systems Inc.:** Descripción general de los conceptos básicos de la tecnología de ATM. Julio 1996.
- **ATM User-Network Interface (UNI) Signalling Specification Version 4.0. ATM Forum:** Especificación de los procedimientos de señalización para establecer, mantener y limpiar dinámicamente conexiones a nivel UNI. Esta definición se realiza en términos de los mensajes y los elementos de información usados para caracterizar dichas conexiones. Julio 1996.
- **Framework for IPv6 over ATM. Peter Schulter, Markus Jork, Geraldine Harter:** Presentación de una solución al problema de utilizar ATM para transportar paquetes IPv6. Junio 1996.
- **Internet Protocol Version 6 and the Digital UNIX implementation Experience. Harrington, Bound, McCann, Thomas.** Descripción de la experiencia de Digital acerca de la implementación de IP Version 6 sobre UNIX y aspectos generales de IPng. Año 1996.
- **IP Next Generation Overview. R. Hinden:** Presentación de generalidades sobre IPng. Mayo 1995.
- **IP Version 6 over Point-to-Point ATM Link. Yamamoto, Cho, Kho, A. Esaki:** Definición de un mecanismo de comunicación para intercambiar paquetes unicast y multicast de IPv6 sobre Redes ATM usadas como link punto a punto. Febrero 1998.
- **Redes ATM. Sebastian Porro (NUIA LETTERS):** Informe básico sobre la estructura de la Tecnología ATM. Julio 1997.
- **RFC 1191. Path MTU Discovery. Mogul, Deering:** Describe una técnica para descubrir dinámicamente el MTU de un camino arbitrario sobre Internet. Noviembre 1990.

- **RFC 1970. Neighbor Discovery for IP versión 6.** Narten, Nordmark, Simpson. Este documento especifica el protocolo de Neighbor Discovery para la versión 6 de IP. Agosto 1996.
- **RFC 1971. IPv6 Stateless Address Autoconfiguration.** Tompson, Narten: Especificación de los pasos que un host debe realizar para la autoconfiguración de sus interfaces. Agosto 1996.
- **RFC 2460. Internet Protocol, Version 6 Specification.** Deering, Hinden: Esta especificación es una actualización a la versión descrita en el RFC 1883. Diciembre 1998.
- **RFC 2461. Neighbor Discovery for IP versión 6.** Narten, Nordmark, Simpson. Este documento es una actualización al RFC 1970. Diciembre 1998.
- **RFC 2463. Internet Control Message Protocol (ICMPv6) for IP Version 6 Specification.** Conta, Deering. Este RFC es una actualización al documento RFC 1885. Diciembre 1998.
- **RFC 2373. IP Version 6 Addressing Architecture.** Hinden, Deering: Actualización al RFC 1884. Julio 1998.
- **SVCs – The Benefits, Questions and Effects on ATM.** Greg Wetzel (AT&T Labs): Descripción de la importancia de los SVC (Switched Virtual Connection) en la Tecnología ATM. Noviembre 1996.
- **Internetworking with TCP/IP.** Comer: Breve descripción sobre como TCP/IP, (diseñado para trabajar sobre redes no orientadas a conexión) puede ser usado sobre Redes ATM (tecnología orientadas a conexión).
- **Traffic Management Specification Version 4.0.** ATM Forum: Definición de procedimientos y parámetros relacionados con el Manejo del Trafico y la Calidad de Servicio (QoS). Abril 1996.

DONACION.....
\$.....
Fecha..... 29-9-05
Inv. E..... Inv. D..... 2009

TES
9914



BIBLIOTECA
FAC. DE INFORMÁTICA
U.N.L.P.

TES
99/4
DIF-02069
SALA



UNIVERSIDAD NACIONAL DE LA PLATA
FACULTAD DE INFORMATICA
Biblioteca
50 y 120 La Plata
catologo:info.unlp.edu.ar
biblioteca@info.unlp.edu.ar



9 780206 9904