ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

“ESTUDIO Y DISEÑO DE MPLS PARA UNA EMPRESA DE TELECOMUNICACIONES CELULAR”

PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE INGENIERO EN ELECTRÓNICA Y TELECOMUNICACIONES

MYRIAN ELIZABETH NARANJO PLAZA
cristal82194@yahoo.com.mx
SANDRA PAULINA PAREDES ULLOA
paredessand@gmail.com

DIRECTOR: ING. CARLOS NOVILLO.
carlos.novillo@epn.edu.ec

QUITO, JUNIO DEL 2008

DECLARACIÓN

Nosotras, MYRIAN ELIZABETH NARANJO PLAZA y SANDRA PAULINA PAREDES ULLOA, declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que hemos consultado las referencias bibliográficas que se incluyen en este documento.

En lo concerniente a "Soporte de Garantía de Servicio (GoS) a flujos privilegiados en MPLS" [1], se deja constancia de que estos estudios son anteriores a la presente tesis y que no son el objeto de la misma. Así mismo, parte de las pruebas realizadas a la red diseñada se realizaron utilizando el simulador OpenSimMPLS [2]. Se incluyen algunas capturas de la simulación de la red diseñada y figuras explicativas que por motivos didácticos corresponden a documentos y conceptos previamente investigados [3].

Cabe recalcar, que el objetivo de la nuestra tesis es "El estudio y diseño de MPLS para una red celular" con datos obtenidos de una empresa de comunicaciones celular, la que nos facilitó la Información de equipos configuraciones y sondeos de tráfico real y con la que tenemos un convenio NDA (para no divulgación de datos reales).

---

[1] Guarantee of Service (GoS) Support over MPLS using Active Techniques. WSEAS Transaction on Computers (WSEAS International Journal), issue 6, volume 3, December 2004. ISSN 1109-2750.

[2] Multiplatform and Opensource GoS/MPLS network simulator. II European Modeling and Simulation Symposium (EMSS2006). International Mediterranean Modelling Multiconference (I3M2006). 4-6 October 2006, Barcelona (SPAIN). ISBN 84-8408-346-2. Puede saberse más sobre esta arquitectura y sobre el simulador, en la web del proyecto http://www.gitaca.com/opensimmpls

[3] Propiedad de José Luis González Sánchez y Manuel Domínguez Dorado, de la Universidad de Extremadura, ESPAÑA.

A través de la presente decaración cedemos nuestros derechos de propiedad intelectual correspondientes a este trabajo, a la Escuela Politécnica Nacional, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

MYRIAN NARANJO PLAZA

SANDRA PAREDES ULLOA

CERTIFICACIÓN

Certifico que el presente trabajo fue desarrollado por MYRIAN ELIZABETH NARANJO PLAZA y SANDRA PAULINA PAREDES ULLOA, bajo mi supervisión.

ING. CARLOS NOVILLO
DIRECTOR DE PROYECTO

AGRADECIMIENTOS

Agradecemos al Ing. Carlos Novillo por su apoyo incondicional durante el desarrollo de este trabajo, a nuestras familias por su comprensión a lo largo de toda nuestra carrera profesional, a la empresa que nos proporcionó la información necesaria para este Proyecto de Titulación y a todos aquellos cuyos trabajos y software sirvió como herramienta y/o fuente para el desarrollo de este proyecto.

DEDICATORIA

Dedicamos este proyecto a nuestros compañeros de Facultad, esperando que les sirva de punto de partida y guía durante su carrera, con la esperanza de que sea la base para estudios posteriores sobre el tema por las implicaciones y potencialidades que las redes implementadas con MPLS tienen en la actualidad, y particularmente en las comunicaciones celulares.

A nuestras familias, ya que sin su continuo apoyo y esfuerzo no hubiese sido posible alcanzar nuestras metas personales y laborales mientras realizábamos este trabajo.

CONTENIDO

Capítulo I: Descripción de red MPLS

En este capítulo se presenta una descripción genérica y detallada de la tecnología de red MPLS, que nos servirá como base fundamental para el desarrollo del presente Proyecto de Titulación.
De la misma forma, revisaremos el proceso de convergencia fijo-móvil como primer paso para la migración de tecnología a una red celular.

Capítulo II: La seguridad y las Redes Privadas Virtuales (VPN)

En este capítulo se toma en cuenta algunas partes fundamentales de la red como son: señalización, seguridades y el sustento lógico de la misma, abarcando modelos genéricos de configuración de red; incluyendo las especificaciones necesarias de diseño sobre redes privadas virtuales.

Capítulo III: Ingeniería de tráfico (TE) y Obtención de Estadísticas

Se abordarán los siguientes temas: Túneles (tipos, establecimiento, creación y configuración), Ingeniería de Tráfico (requisitos, comparación con ATM, atributos de tráfico y de recursos; encaminamiento, señalización RSVP-TE y análisis de tráfico en la red) y Estadísticas (herramientas de planificación para toma de decisiones futuras y optimización).

Capítulo IV: **Diseño de una red MPLS y la configuración de los equipos**

Se realizará el diseño de la red con MPLS presentando esquemas, diagramas y configuración de todos los equipos de la red.

Capítulo V: QoS y Establecimiento de clases de servicio

Se abordarán los siguientes temas: QoS (utilización de técnicas y herramientas de calidad de servicio; mecanismos de convergencia de datos con aplicaciones en tiempo real; prevención y recuperación de errores) y Clases de Servicio (terminología, categorización de servicios diferenciados a ofrecer).

Capítulo VI: Propuesta de monitoreo de tráfico con la herramienta computacional OpenSimMPLS

En este capítulo se realizará la descripción de OpenSimMPLS, sus características y funcionamiento básico, los requisitos para la implementación de OpenSimMPLS y la propuesta de monitoreo de tráfico con calidad de servicio línea.

Capítulo VII: Simular con Network Simulator (NS), el funcionamiento de la red diseñada.

En este capítulo se realizará la descripción de NS (Network Simulator), sus características y funcionamiento; la implementación en la red diseñada con su respectiva simulación y exposición de los resultados obtenidos.

Capítulo VIII: Conclusiones y recomendaciones

RESUMEN

MPLS (Multi Protocol Label Switching) es un mecanismo de encapsulamiento de paquetes muy eficiente utilizado en la actualidad. La razón de su efectividad es que éste utiliza etiquetas en lugar de paquetes para transportar datos. Los paquetes MPLS pueden transportarse sobre otras tecnologías de capa dos.

Por esta razón, en una empresa de telecomunicaciones celular donde se requiere una tecnología que permita la entrega de servicios como: VPN (Virtual Private Network), QoS (Calidad de Servicio) e Ingeniería de Tráfico; es altamente recomendable el empleo de MPLS en el núcleo de red.

MPLS no solamente integra servicios de valor agregado a las comunicaciones, sino que además extiende la capacidad de negocio de una empresa ya que permite ofertar nuevas oportunidades en el manejo ancho de banda e integración de servicios sobre la red.

Para el desarrollo del presente tema, se utilizó una infraestructura aproximada a la de una red celular real, así como también se emplearon datos de tráfico cercanos a los que se transportan por esta red para obtener un análisis efectivo.

Adicionalmente, se utiliza simuladores para complementar el estudio donde se permite integrar el diseño propuesto (topología y configuración) con el análisis de Calidad de Servicio en la red de telecomunicaciones celular.

PRESENTACIÓN

Las nuevas tecnologías invaden hoy en día las empresas que buscan mejorar y optimizar sus redes de comunicaciones. La tecnología MPLS (Multi Protocol Label Switching) se despliega en el núcleo de red del proveedor de servicios, proporcionando a éste un mayor control sobre la Calidad de Servicio, la Ingeniería de Tráfico y las Redes Privadas Virtuales; además mejora la utilización del ancho de banda a la vez que reduce los requisitos a los equipos de comunicación de los clientes que están conectados a un servicio sobre MPLS.

Una red MPLS, puede transportar múltiples protocolos distintos y de manera simultánea, esta tecnología se está transformando en el elemento clave de estrategia de comunicaciones en las empresas, y en el principal impulsor de la convergencia y conectividad de redes multiservicio.

MPLS como solución convergente de conectividad, es un entorno de servicio administrado de extremo a extremo, donde las aplicaciones como voz y datos son soportadas sobre una misma plataforma. Adicionalmente, con MPLS el proveedor puede ofrecer servicios adicionales que las tecnologías tradicionales como ATM no soportan.

Bajo este escenario, MPLS incrementa las oportunidades de negocio para servicios VPN, servicios de clase Premium, líneas dedicadas virtuales, además que presenta servicios IP diferenciados en categorías como oro (entrega y latencia garantizada), plata (entrega garantizada) y bronce (mejor esfuerzo).

Con esta clasificación de tráfico, incluso se facilita el cobro de tarifas sobre estos servicios, dado que es de suma importancia para el cliente que el servicio por el que está pagando le garantice alta disponibilidad de la infraestructura de red incluso cuando exista congestión en la misma, asegurando que no se descarte la información que no puede ser transmitida como en otras tecnologías.

Entre las oportunidades de servicio se debe tomar en cuenta: redundancia, servicios centralizados, acceso a Internet, acceso remoto, etc. Varios proveedores que buscan disminuir sus costos de operación al momento de vender sus servicios han optado por implementar MPLS.

MPLS ha recibido mucha atención desde su introducción en el mundo de las redes. A pesar de que en este trabajo no se ha logrado profundizar en las herramientas que presta esta tecnología, se plantean las bases necesarias para entender el funcionamiento de MPLS y desarrollar el diseño aplicado a una empresa de telecomunicaciones celular.

ÍNDICE

CAPÍTULO I

“DESCRIPCIÓN DE RED MPLS”

CAPÍTULO I

1. DESCRIPCIÓN DE RED MPLS

1.1. FUNDAMENTOS DE MPLS

1.1.1. DEFINICIÓN

MPLS (Multi-Prototocol Label Switching), es un estándar emergente del IETF[4] que surgió para unificar diferentes soluciones de conmutación multinivel, propuestas por distintos fabricantes a mitad de los 90's.

Como concepto, MPLS tiene dos aspectos que tomar, el primero que como protocolo es bastante sencillo y el segundo que las implicaciones que supone su implementación real son enormemente complejas.

Según el interés que se ponga, a la hora de explicar sus características y utilidad, MPLS se puede presentar como:

1. Un sustituto de la conocida arquitectura IP sobre ATM;
2. Un protocolo para hacer túneles (sustituyendo a las técnicas habituales de "tunneling"[5]), ó
3. Una técnica para acelerar el encaminamiento de paquetes, incluso para eliminar por completo el routing.

En realidad, MPLS hace un poco de todo eso, ya que integra sin discontinuidades los niveles 2 (enlace) y 3 (red), combinando eficazmente las funciones de control del routing con la simplicidad y rapidez de la conmutación de nivel 2.

[4] IETF: Fuerza de Trabajo de Ingeniería de Internet.
[5] Tunneling: Arquitectura diseñada para suministrar los servicios necesarios para implementar cualquier esquema de encapsulación punto a punto estándar.

Todas las soluciones de conmutación multi-nivel (incluido MPLS) se basan en dos componentes básicos comunes:

- La separación entre las funciones de control (routing[6]) y de envío (forwarding[7]), lo que implica una evolución en la manera de construir y gestionar estas redes.
- El paradigma de intercambio de etiquetas para el envío de datos.

Los problemas que resuelve son los que presentan las soluciones actuales de IP sobre ATM, tales como la expansión sobre una topología virtual superpuesta, así como la complejidad de gestión de dos redes separadas y tecnológicamente diferentes.

Al combinar en uno solo lo mejor de cada nivel (la inteligencia del routing con la rapidez del switching), MPLS ofrece nuevas posibilidades en la gestión de backbones, así como en la provisión de nuevos servicios de valor añadido.

Si bien es cierto que MPLS mejora notablemente el rendimiento del mecanismo de envío de paquetes, éste no era el principal objetivo del grupo del IETF.

Los objetivos establecidos por ese grupo en la elaboración del estándar eran:

- MPLS debía funcionar sobre cualquier tecnología de transporte, no sólo ATM.
- MPLS debía soportar el envío de paquetes tanto unicast como multicast.
- MPLS debía ser compatible con el Modelo de Servicios Integrados del IETF, incluyendo el protocolo RSVP[8].
- MPLS debía permitir el crecimiento constante de Internet.
- MPLS debía ser compatible con los procedimientos de operación, administración y mantenimiento de las actuales redes IP.

[6] Routing : es decir el control de la información sobre la topología y tráfico en la red.
[7] Forwarding : es decir el envío en sí de datos entre elementos de la red.
[8] RSVP: Protocolo de reserva de recursos: Protocolo que hace posible la reserva de recursos a través de una red IP. Las aplicaciones que se ejecutan en los sistemas finales IP pueden usar RSVP para indicarle a los otros nodos la naturaleza (ancho de banda, fluctuación de fase, ráfaga máxima, etc.) de los flujos de paquetes que desean recibir.

MPLS no perseguía eliminar totalmente el encaminamiento convencional de capa 3 conocido como prefijos de red, ya que el encaminamiento tradicional de nivel 3 siempre sería un requisito en Internet por los siguientes motivos:

- El filtrado de paquetes en los cortafuegos (FW) de acceso a las LAN corporativas y en los límites de las redes de los NSPs[9] es un requisito fundamental para poder gestionar la red y los servicios con las necesarias garantías de seguridad. Para ello se requiere examinar la información de la cabecera de los paquetes, lo que impide prescindir del uso del nivel 3 en ese tipo de aplicaciones.

- No es probable que los sistemas finales (hosts[10]) implementen MPLS. Necesitan enviar los paquetes a un primer dispositivo de red (nivel 3) que pueda examinar la cabecera del paquete para tomar luego las correspondientes decisiones sobre su envío hasta su destino final. En este primer salto se puede decidir enviarlo por routing convencional o asignar una etiqueta y enviarlo por un LSP[11].

- Las etiquetas MPLS tienen solamente significado local (es imposible mantener vínculos globales entre etiquetas y hosts en todo el Internet). Esto implica que en algún punto del camino algún dispositivo de nivel 3 debe examinar la cabecera del paquete para determinar con exactitud por donde lo envía: por routing convencional o entregándolo a un LSR[12], que lo expedirá por un nuevo LSP.

- Del mismo modo, el último LSR de un LSP debe usar encaminamiento de nivel 3 para entregar el paquete al destino, una vez suprimida la etiqueta, como se verá seguidamente al describir la funcionalidad MPLS.

[9] (NSPs) Proveedores de servicios de red.
[10] Terminal de red.
[11] El algoritmo de intercambio de etiquetas permite así la creación de "caminos virtuales" conocidos como LSP (Label-Switched Paths)
[12] Label-Switched Router (LSR)

1.2. FUNCIONAMIENTO BÁSICO

1.2.1. ELEMENTOS FUNDAMENTALES DE MPLS

1. Label-Switched Router (LSR)

En términos generales un LSR es un router capaz de soportar MPLS. El LSR como tal, es un dispositivo que implementa la conmutación de etiquetas, generalmente se encuentra ubicado en el medio de la red y es capaz de enviar datagramas. En las primeras versiones de MPLS, el LSR se comparaba con un switch ATM que cumplía exactamente las mismas funciones dentro del campo VPI/VCI.

2. Label-Switched Path (LSP)

El LSP es el camino de conmutación de etiquetas, establecido entre uno o más LSR's, en uno de los niveles de la jerarquía que siguen los paquetes de una FEC particular.

3. Label Distribution Protocol (LDP) u otro protocolo como Protocolo de Reserva de Recursos (RSVP)

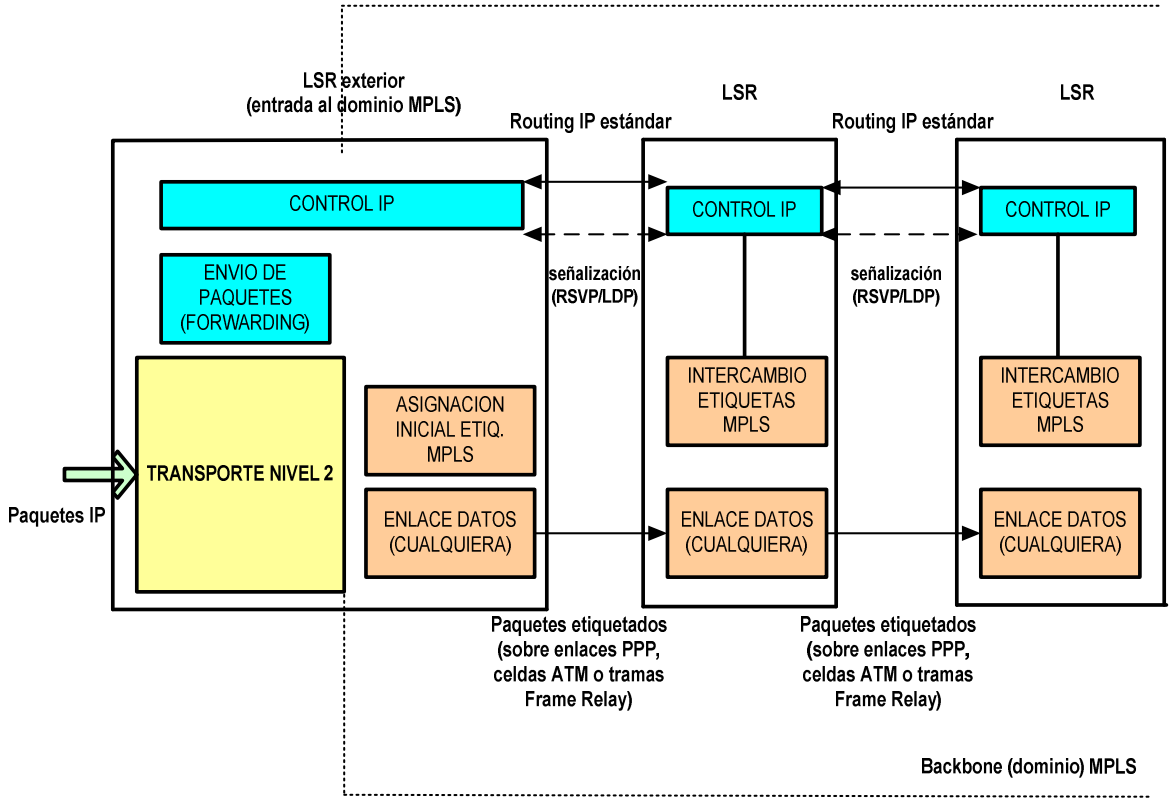
LDP es un conjunto de reglas que sirven para que un LSR le informe a otro el significado de las etiquetas utilizadas para el reenvío del tráfico entre ellos.

1.2.2. FUNCIONAMIENTO DEL ENVÍO DE PAQUETES EN MPLS

La base del MPLS está en la asignación e intercambio de etiquetas expuesto anteriormente, que permiten el establecimiento de los caminos LSP por la red. Los LSPs son simplex por naturaleza (se establecen para un sentido del tráfico en cada punto de entrada a la red); para el tráfico dúplex requiere dos LSPs, uno en cada sentido. Cada LSP se crea a base de concatenar uno o más saltos (hops) en

los que se intercambian las etiquetas, de modo que cada paquete se envía de un "conmutador de etiquetas" (Label-Swiching Router) a otro, a través del dominio MPLS. Un LSR no es sino un router especializado en el envío de paquetes etiquetados por MPLS.

FIGURA 1.1. ESQUEMA FUNCIONAL DE MPLS



Fuente: www.cisco.com

Elaborado por: Las autoras

MPLS separa las dos componentes funcionales de control (routing) y de envío (forwarding). Del mismo modo, el envío se implementa mediante el intercambio de etiquetas en los LSPs, como se ve en la Figura 1.1.

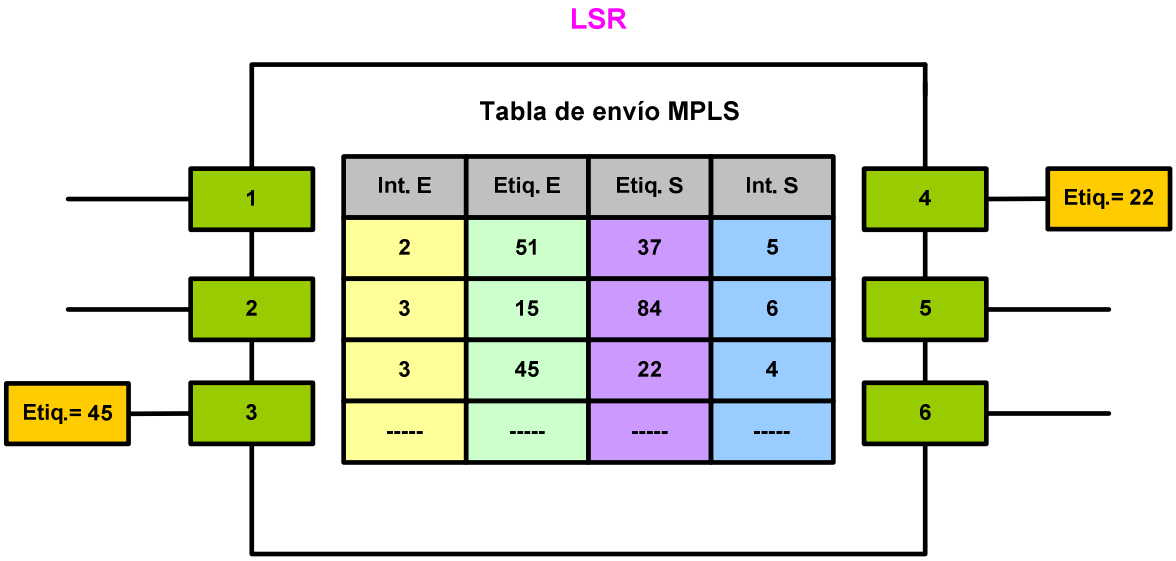
Sin embargo, MPLS no utiliza ninguno de los protocolos de señalización ni de encaminamiento definidos por el Foro ATM; en lugar de ello, en MPLS o bien se utiliza el protocolo RSVP[13] o bien un nuevo estándar de señalización (el Label Distribution Protocol, LDP). Pero, de acuerdo con los requisitos del IETF, el

[13] RSVP: Protocolo de reserva de recursos: Protocolo que hace posible la reserva de recursos a través de una red IP. Las aplicaciones que se ejecutan en los sistemas finales IP pueden usar RSVP para indicarle a los otros nodos la naturaleza (ancho de banda, fluctuación de fase, ráfaga máxima, etc.) de los flujos de paquetes que desean recibir.

transporte de datos puede ser cualquiera. Una red IP habilitada para MPLS es ahora mucho más sencilla de gestionar que la solución clásica IP/ATM. Ahora ya no hay que administrar dos arquitecturas diferentes a base de transformar las direcciones IP y las tablas de encaminamiento en las direcciones y el encaminamiento ATM: esto lo resuelve el procedimiento de intercambio de etiquetas MPLS. El papel de ATM queda restringido al mero transporte de datos basado en celdas. Para MPLS esto es indiferente, ya que puede utilizar otros transportes como Frame Relay, o directamente sobre líneas punto a punto.

Un camino LSP es el circuito virtual que siguen por la red todos los paquetes asignados a la misma FEC. Al primer LSR que interviene en un LSP se le denomina de entrada o de cabecera y al último se le denomina de salida o de cola. Los dos están en el exterior del dominio MPLS. El resto, entre ambos, son LSRs interiores del dominio MPLS. Un LSR es como un router que funciona a base de intercambiar etiquetas según una tabla de envío. Esta tabla se construye a partir de la información de encaminamiento que proporciona la componente de control. Cada entrada de la tabla contiene un par de etiquetas entrada/salida correspondientes a cada interfaz de entrada, que se utilizan para acompañar a cada paquete que llega por esa interfaz y con la misma etiqueta (en los LSR exteriores sólo hay una etiqueta, de salida en el de cabecera y de entrada en el de cola). En la figura 1.2. se ilustra un ejemplo del funcionamiento de un LSR del núcleo MPLS. A un paquete que llega al LSR por la interfaz 3 de entrada con la etiqueta 45 el LSR le asigna la etiqueta 22 y lo envía por el interfaz 4 de salida al siguiente LSR, de acuerdo con la información de la tabla.

FIGURA 1.2. DETALLE DE LA TABLA DE ENVIO DE UN LSR



Fuente: www.cisco.com

Elaborado por: Las autoras

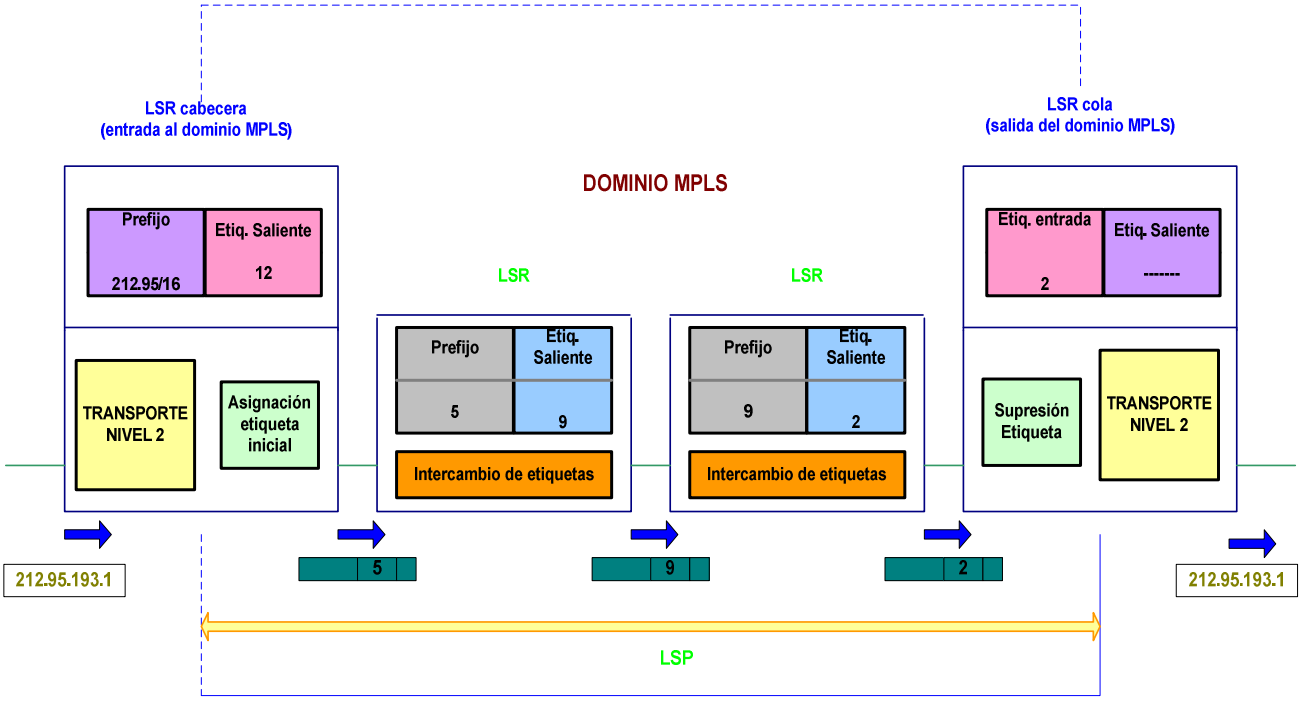
El algoritmo de intercambio de etiquetas requiere la clasificación de los paquetes a la entrada del dominio MPLS para poder hacer la asignación por el LSR de cabecera.

En la figura 1.3., el LSR de entrada recibe un paquete normal (sin etiquetar) cuya dirección de destino es 212.95.193.1. El LSR consulta la tabla de encaminamiento y asigna el paquete a la clase FEC definida por el grupo 212.95/16. Así mismo, este LSR le asigna una etiqueta (con valor 5 en el ejemplo) y envía el paquete al siguiente LSR del LSP. Dentro del dominio MPLS los LSR ignoran la cabecera IP; solamente analizan la etiqueta de entrada, consultan la tabla correspondiente (tabla de conmutación de etiquetas) y la reemplazan por otra nueva, de acuerdo con el algoritmo de intercambio de etiquetas. Al llegar el paquete al LSR de cola (salida), ve que el siguiente salto lo saca de la red MPLS; al consultar ahora la tabla de conmutación de etiquetas quita ésta y envía el paquete por routing convencional.

Como se ve, la identidad del paquete original IP queda enmascarada durante el transporte por la red MPLS, que no "mira" sino las etiquetas que necesita para su

envío por los diferentes saltos LSR que configuran los caminos LSP. Las etiquetas se insertan en las cabeceras MPLS, entre los niveles 2 y 3. Esto permite que MPLS funcione sobre cualquier tipo de transporte: PPP, LAN, ATM, Frame Relay, etc. Por ello, si el protocolo de transporte de datos contiene ya un campo para etiquetas (como ocurre con los campos VPI/VCI de ATM y DLCI de Frame Relay), se utilizan esos campos nativo para las etiquetas. Sin embargo, si la tecnología de nivel 2 empleada no soporta un campo para etiquetas p. ej. enlaces PPP o LAN), entonces se emplea una cabecera genérica MPLS de 4 octetos, que contiene un campo específico para la etiqueta y que se inserta entre la cabecera del nivel 2 y la del paquete (nivel 3).

FIGURA 1.3. EJEMPLO DE ENVÍO DE UN PAQUETE POR UN LSP



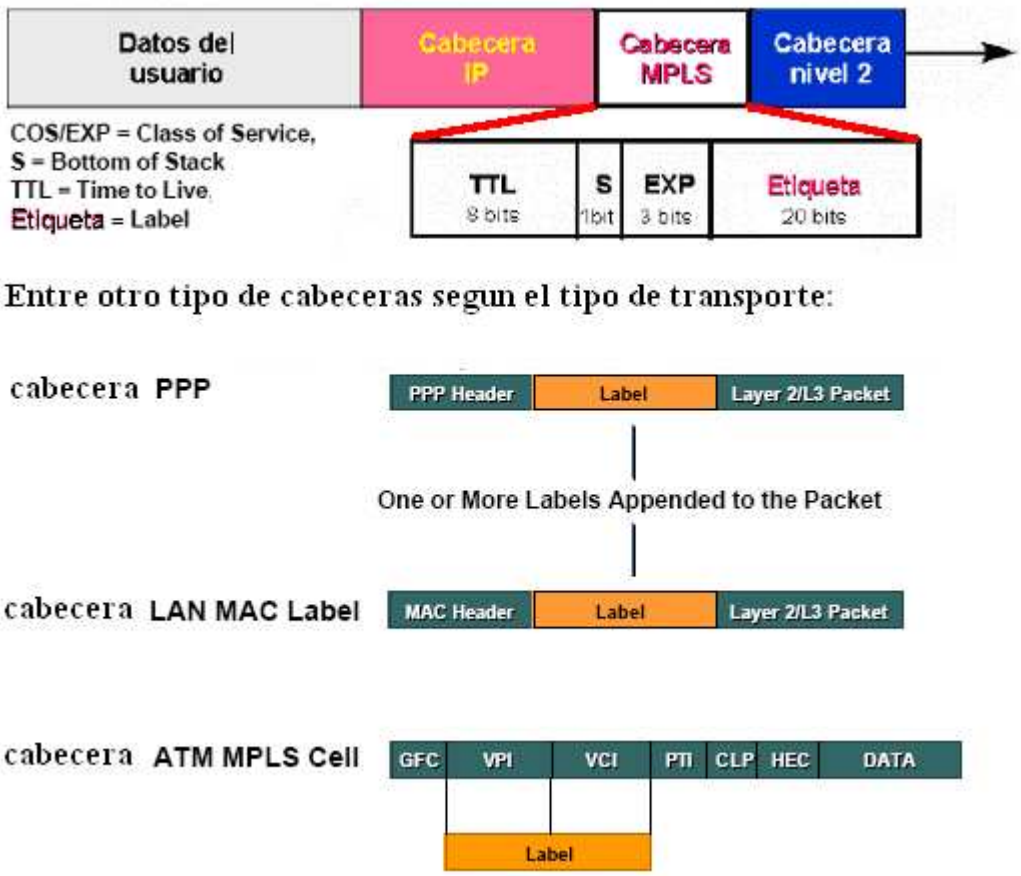
Fuente: www.cisco.com

Elaborado por: Las autoras

En la figura 1.4. se representa el esquema de los campos de la cabecera genérica MPLS y su relación con las cabeceras de los otros niveles. Según se muestra en la figura, los 32 bits de la cabecera MPLS se reparten en: 20 bits para la etiqueta MPLS, 3 bits para identificar la clase de servicio en el campo EXP (experimental, anteriormente llamado CoS), 1 bit de stack para poder apilar etiquetas de forma jerárquica (S) y 8 bits para indicar el TTL (time-to-live) que sustenta la

funcionalidad estándar TTL de las redes IP. De este modo, las cabeceras MPLS permiten cualquier tecnología o combinación de tecnologías de transporte, con la flexibilidad que esto supone para un proveedor IP a la hora de extender su red.

FIGURA 1.4. ESTRUCTURA DE UNA CABECERA GENÉRICA MPLS



Fuente: www.cisco.com

Elaborado por: Las autoras

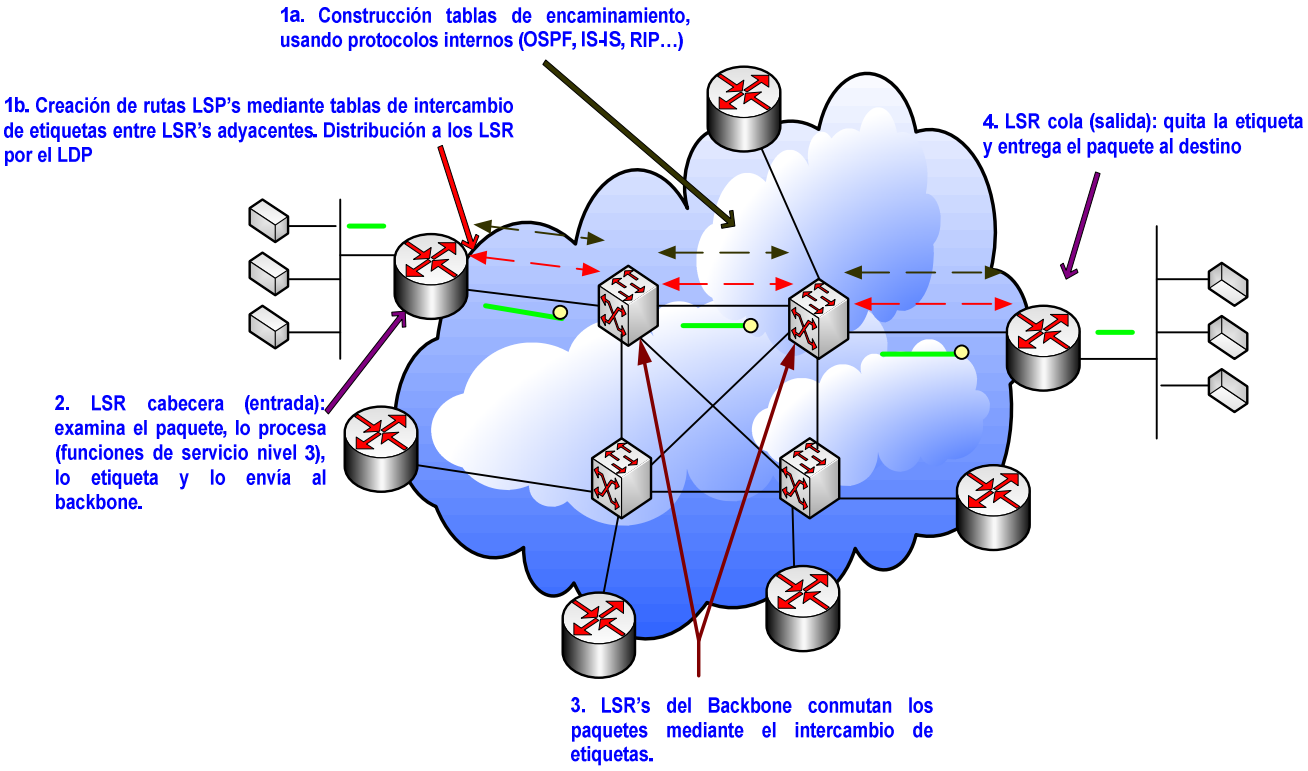
1.2.3. FUNCIONAMIENTO GLOBAL MPLS

El esquema global de funcionamiento es el que se muestra en la figura 1.5., donde quedan reflejadas las diversas funciones en cada uno de los elementos que integran la red MPLS. Es importante destacar que en el borde de la nube MPLS tenemos una red convencional de routers IP. El núcleo MPLS proporciona una arquitectura de transporte que hace aparecer a cada par de routers a una distancia de un sólo salto.

Funcionalmente es como si estuvieran unidos todos en una topología mallada directamente o por PVCs ATM. Ahora, esa unión a un solo salto se realiza por MPLS mediante los correspondientes LSPs (puede haber más de uno para cada par de routers). La diferencia con topologías conectivas reales es que en MPLS la construcción de caminos virtuales es mucho más flexible y que no se pierde la visibilidad sobre los paquetes IP.

Esto abre enormes posibilidades a la hora de mejorar el rendimiento de las redes y de soportar nuevas aplicaciones de usuario.

FIGURA 1.5. FUNCIONAMIENTO DE UNA RED MPLS



Fuente: www.cisco.com

Elaborado por: Las autoras

1.2.4. CONTROL DE LA INFORMACIÓN EN MPLS

1. Cómo se generan las tablas de envío que establecen los LSPs.
2. Cómo se distribuye la información sobre las etiquetas a los LSRs.

El primero, está relacionado con la información que se tiene sobre la red: topología, patrón de tráfico, características de los enlaces, etc. Es la información de control típica de los algoritmos de encaminamiento. MPLS necesita esta información de routing para establecer los caminos virtuales LSPs. Lo más lógico es utilizar la propia información de encaminamiento que manejan los protocolos internos IGP (OSPF, IS-IS, RIP...) para construir las tablas de encaminamiento (recuérdese que los LSR son routers con funcionalidad añadida). Esto es lo que hace MPLS precisamente: para cada "ruta IP" en la red se crea un "camino de etiquetas" a base de concatenar las de entrada/salida en cada tabla de los LSRs; el protocolo interno correspondiente se encarga de pasar la información necesaria.

El segundo aspecto se refiere a la información de "señalización", pero siempre que se quiera establecer un circuito virtual se necesita algún tipo de señalización para marcar el camino, es decir, para la distribución de etiquetas entre los nodos.

Sin embargo, la arquitectura MPLS no asume un único protocolo de distribución de etiquetas; de hecho se están estandarizando algunos existentes con las correspondientes extensiones; unos de ellos es el protocolo RSVP[14] del Modelo de Servicios Integrados del IETF. Pero además, en el IETF se están definiendo otros nuevos, específicos para la distribución de etiquetas, tal es el caso del Label Distribution Protocol (LDP[15]).

1.3. EVOLUCIÓN

Para entender mejor las ventajas de la solución MPLS, vale la pena revisar los esfuerzos anteriores de integración de los niveles 2 y 3 que han llevado finalmente a la adopción del estándar MPLS.

[14] RSVP: Protocolo de reserva de recursos: Protocolo que hace posible la reserva de recursos a través de una red IP. Las aplicaciones que se ejecutan en los sistemas finales IP pueden usar RSVP para indicarle a los otros nodos la naturaleza (ancho de banda, fluctuación de fase, ráfaga máxima, etc.) de los flujos de paquetes que desean recibir.

[15] Establece un mapa de etiquetas de destino de red, definido en RFC 3035 and 3036 con un envió de datos de clase equivalente y FEC.

A mediados de los 90's, IP fue ganando terreno como protocolo de red a otras arquitecturas en uso[16]. Hay que recordar que los backbones IP de los NSP's, estaban construidos basados en routers conectados por líneas dedicadas T1/E1[17] y T3/E3[18].

El crecimiento explosivo del Internet había generado un déficit de ancho de banda en aquel esquema de enlaces individuales. La respuesta de los NSP's fue el incremento del número de enlaces y de la capacidad de los mismos; planteando la necesidad de aprovechar mejor los recursos de red existentes, sobre todo la utilización eficaz del ancho de banda de todos los enlaces.

Con los protocolos habituales de encaminamiento (basados en métricas del menor número de saltos), ese aprovechamiento del ancho de banda global no resultaba efectivo. Había que idear otras alternativas de ingeniería de tráfico; estos esfuerzos trataban de combinar, de diversas maneras, la eficacia y la rentabilidad de los conmutadores ATM con las capacidades de control de los routers IP.

A favor de integrar los niveles 2 y 3 estaba el hecho de las infraestructuras de redes ATM en (1995-97) ofrecía entonces una buena solución a los problemas de crecimiento de los NSP's.

Por un lado proporcionaba mayores velocidades (155 Mpbs) y; por otro, las características de respuesta determinísticas de los circuitos virtuales ATM posibilitaban la implementación de soluciones de ingeniería de tráfico. El modelo de red "IP sobre ATM" pronto ganó adeptos entre la comunidad de NSP's, a la vez que facilitó la entrada de los operadores telefónicos en la provisión de servicios IP y de conexión a Internet al por mayor.

[16] (SNA, IPX, AppleTalk, OSI...).
[17] T1: Servicio de portadora WAN digital que transmite datos formateados a 1.544 Mbps a través de una red de conmutación telefónica, comúnmente utilizada en los EE.UU. E1: Esquema de transmisión digital de WAN utilizado en Europa, que lleva datos a una velocidad de 2.048 Mbps.
[18] T3: Servicio de transmisión digital a una velocidad de 44.736 Mbps en los EE.UU. E3: en Europa de 34.368 Mbps.

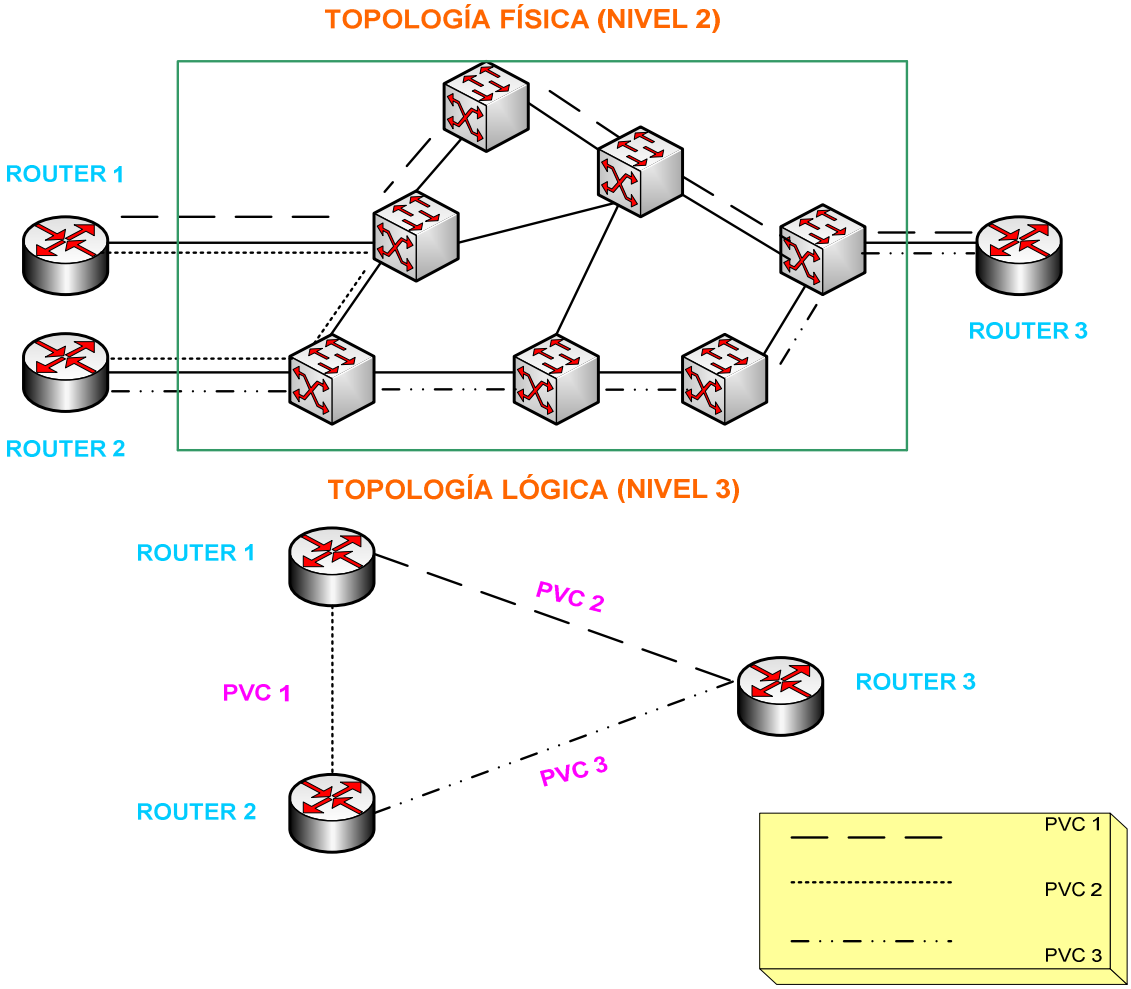
El funcionamiento IP/ATM supone la superposición de una topología virtual de routers IP sobre una topología real de conmutadores ATM. El backbone ATM se presenta como una nube central (el núcleo) rodeado por los routers de la periferia.

Cada router se comunica con el resto mediante los circuitos virtuales permanentes (PVC's) que se establecen sobre la topología física de la red ATM. Los PVC's actúan como circuitos lógicos y proporcionan la conectividad necesaria entre los routers de la periferia.

Estos, sin embargo, desconocen la topología real de la infraestructura ATM que sustenta los PVC's. Los routers ven los PVC's como enlaces punto a punto entre cada par.

En la figura 1.6., se representa un ejemplo en el que se puede comparar la diferencia entre la topología física de una red ATM con la de la topología lógica IP superpuesta sobre la anterior.

FIGURA 1.6. TOPOLOGÍA FÍSICA ATM Y TOPOLOGÍA LÓGICA IP SUPERPUESTA

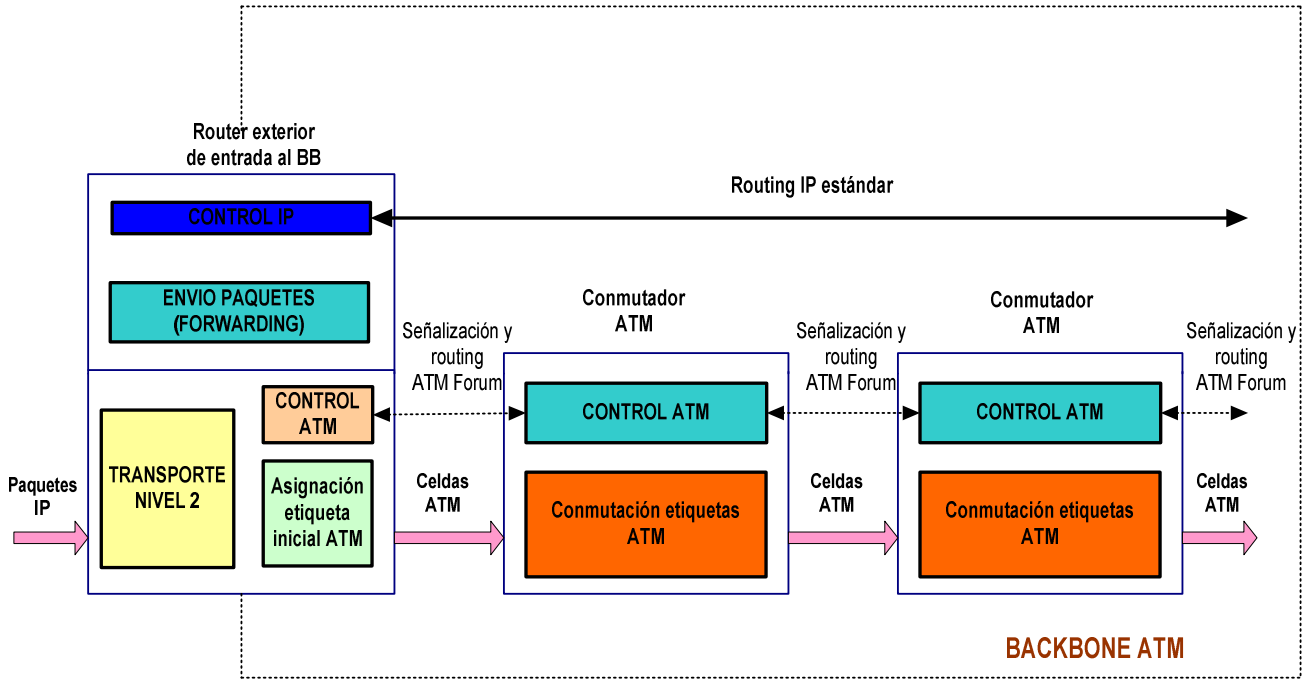


Fuente: www.cisco.com

Elaborado por: Las autoras

En la figura 1.7. se representa el modelo IP/ATM con la separación de funciones entre lo que es routing IP en el nivel 3 (control y envío de paquetes) y lo que es conmutación en el nivel 2 (control/señalización y envío de celdas). Aunque se trata de una misma infraestructura física, en realidad existen dos redes separadas, con diferentes tecnologías, con diferente funcionamiento y; lo que quizás es más sorprendente, concebidas para dos finalidades totalmente distintas.

FIGURA 1.7. MODELO FUNCIONAL IP SOBRE ATM



Fuente: www.cisco.com

Elaborado por: Las autoras

El modelo IP/ATM tiene también sus inconvenientes: hay que gestionar dos redes diferentes, una infraestructura ATM y una red lógica IP superpuesta, lo que supone a los proveedores de servicio unos mayores costes de gestión global de sus redes. Existe, además, lo que se llama la "tasa impuesta por la celda", un overhead aproximado del 20% que causa el transporte de datagramas IP sobre las celdas ATM y que reduce en ese mismo porcentaje el ancho de banda disponible.

Por otro lado, la solución IP/ATM presenta los típicos problemas de crecimiento exponencial n x (n-1) al aumentar el número de nodos IP sobre una topología completamente mallada. Por ejemplo, en una red con 5 routers externos con una topología virtual totalmente mallada sobre una red ATM. Son necesarios 5 x 4 = 20 PVC's (uno en cada sentido de transmisión). Si se añade un sexto router se necesitan 10 PVC's más para mantener la misma estructura (6 x 5 = 30). Un

problema adicional del crecimiento exponencial de rutas es el mayor esfuerzo que tiene que hacer el correspondiente protocolo IGP[19].

El modelo IP/ATM, si bien presenta ventajas evidentes en la integración de los niveles 2 y 3, lo hace de modo discontinuo, a base de mantener dos redes separadas. MPLS, logra esa integración de niveles sin discontinuidades.

Por esta razón se buscó un estándar que pudiera funcionar sobre cualquier tecnología de transporte de datos en el nivel de enlace. Dando lugar a que el Grupo de Trabajo de MPLS (IETF) estableciera en 1977, como objetivo la adopción de un estándar unificado e ínteroperativo denominado MPLS.

1.4. PROTOCOLOS DE DISTRIBUCIÓN DE ETIQUETAS

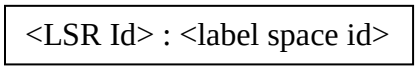
1.4.1. LDP (LABEL DISTRIBUTION PROTOCOL)

El que se encuentra definido en los documentos RFC 3035 y RFC 3036 del Network Working Group para distribuir las etiquetas o labels por los LSRs (Label Switching Routers) y soportar envíos MPLS a lo largo de los routed paths.

Esquema LDP:

LDP esta formado por seis octetos que identifican el espacio de una etiqueta el LSR. Los primeros cuatro octetos identifican al LSR con un valor global único de 32-bit de Id de ruteo asignada al LSR. Los otros dos octetos identifican el especifico espacio de etiqueta del llamado platform-wide label spaces, siempre encerados.

FIGURA 1.8. DIAGRAMA



Fuente: www.cisco.com

Elaborado por: Las autoras

[19] IGP: Protocolo de gateway interior: Protocolo de internet que se utiliza para intercambiar información de enrutamiento dentro de un sistema autónomo. IGRP, OSPF y RIP son ejemplos de IGP de internet comunes.

Se puede tener múltiples etiquetas de espacio manejadas por un mismo LSR.

1.4.2. RSVP (PROTOCOLO DE RESERVA DE RECURSOS)

RSVP es un protocolo de señalización que reserva la capacidad solicitada por un flujo en todos los routers del camino. Este protocolo requiere guardar información de estado en todos los routers que conforman el trayecto, y se caracteriza por ser un servicio orientado a conexión.

Es un protocolo diseñado principalmente para tráfico multicast y aunque no es un protocolo de routing, utiliza los protocolos que sí lo son para su funcionamiento.

Los componentes que deben tener los routers para implementar RSVP son los siguientes:

- Control de admisión: comprueba si la red tiene los recursos suficientes para cumplir con una petición.
- Política de control: determina si el usuario tiene los permisos adecuados para la petición realizada. Esta comprobación se puede realizar consultando una base de datos mediante el protocolo COPS (Common Open Policy Service).
- Clasificador de paquetes: clasifica los paquetes en categorías de acuerdo a la QoS a la que pertenecen. Cada categoría tiene una cola y un espacio propio para buffers en el router.
- Organizador de paquetes: organiza el envío de los paquetes dentro de cada categoría (cada cola).

Si no se cumplen con las condiciones pedidas se rechaza la llamada o control de admisión. Uno de los principales problemas de este protocolo es que los routers deben mantener información sobre muchos flujos y por tanto una gran cantidad de información de estado.

En hardware los fabricantes no han desarrollado implementaciones eficientes de RSVP, debido a su elevado costo. Sin embargo RSVP puede desarrollar un papel

eficiente en la red de acceso donde los enlaces son de baja capacidad y los routers soportan pocos flujos. En el caso de MPLS, este protocolo es útil ya que esta tecnología de red no maneja números de flujos muy elevados.

1.5. NORMALIZACION

1.5.1. IETF

El IETF es una organización internacional de normalización de carácter abierto cuyos objetivos son contribuir al desarrollo de Internet en aspectos fundamentales como: encaminamiento, transporte y seguridad.

Otra de sus funciones principales es velar por el desarrollo de la arquitectura de red y los protocolos que soporta la misma. Por lo tanto, el IETF es la comunidad que aprueba, investiga y desarrolla los estándares que en un futuro se utilizarán en Internet.

1.5.2. MPLS FORUM

El MPLS FORUM es un grupo abierto con acceso remoto vía web para opinar sobre mejoras y aplicaciones para MPLS.

1.6. APLICACIONES

1.6.1. VPN’s

Una VPN es una tecnología de red que permite una extensión de la red local sobre una red pública o no controlada, como por ejemplo Internet.

Las soluciones VPN (red privada virtual) basadas en MPLS permiten a empresas con varias sedes, red de distribuidores y empresas asociadas, interconectarse entre ellas y conectarse a Internet (acceso centralizado a Internet) a través de una red privada virtual cumpliendo los máximos requerimientos de disponibilidad y seguridad.

Cada una de las sedes o componentes de la red VPN pueden conectarse a la VPN mediante alguno de los siguientes tipos de acceso: ADSL, RDSI, SDSL y líneas dedicadas con o sin backup de hasta 155 Mbps.

A continuación se muestran las principales diferencias entre una VPN IP-Sec frente a una VPN basada en MPLS.

TABLA 1.1. RESUMEN COMPARATIVO IP-SEC Vs MPLS EN VPNs

VPN IP-Sec	VPN MPLS
Se configura a través de internet.	Se configura a través de una red privada.
Sujeta a pérdidas de paquetes, congestión y otros problemas típicos de internet.	MPLS garantiza latencias bajas y reducción de pérdidas de paquetes.
Túneles y encriptación para asegurar la integridad y confidencialidad de los datos.	Integridad y confidencialidad de los datos por VRF.
No soporta CoS ni QoS.	Permite implementar CoS y QoS.
Cada sede necesita estar conectada a internet para implementar VPN's bajo IP-Sec.	Las VPN's basadas en MPLS permiten compartir un único acceso a internet entre todas las sedes.
El cliente requiere determinado software o hardware.	No se requiere ningún software o hardware específico.

Fuente: www.cisco.com

Elaborado por: Las autoras

1.6.2. INGENIERÍA DE TRÁFICO (IT)

Todas las redes requieren un monitoreo para realizar proyecciones futuras basadas en el tráfico actual y las condiciones de la red. Para esto se usa la ingeniería de tráfico la que nos permite por medio de funciones probabilísticas, predecir: número de usuarios, condiciones críticas de red y planificar el funcionamiento de la red en función de los posibles escenarios a presentarse.
En el capítulo referente a Ingeniería de Tráfico se detallará el proceso.

1.6.3. CALIDAD DE SERVICIO (QoS)

Ofrecer calidad de servicio quiere decir garantizar el valor de uno o varios de los parámetros que definen la calidad de servicio que ofrece la red. El contrato que especifica los parámetros de calidad de servicio acordados entre el proveedor y el usuario (cliente) se denomina SLA (Service Level Agreement).

Los parámetros típicos que se deben cumplir en una red que tenga calidad de servicio son los siguientes:

- Disponibilidad: tiempo mínimo que el operador asegura que la red esté en funcionamiento.
- Ancho de Banda: indica el ancho de banda mínimo que el operador garantiza al usuario dentro de su red.
- Pérdida de paquetes: Máximo de paquetes perdidos (siempre y cuando el usuario no exceda el caudal garantizado).
- Round Trip Delay: el retardo de ida y vuelta medio de los paquetes.
- Jitter: la fluctuación que se puede producir en el retardo de ida y vuelta medio.

Existen dos tratos preferenciales que se le pueden asignar a un usuario en la red:

- Carril BUS: reservar capacidad para su uso exclusivo, denominado QoS hard.
- Ambulancia: darle a un usuario mayor prioridad que a otro.

MPLS permite asignar QoS a un cliente o a un tipo de tráfico una FEC a la que se asocie un LSP que discurra por enlaces con bajo nivel de carga.

1.6.4. CLASES DE SERVICIO (CoS)

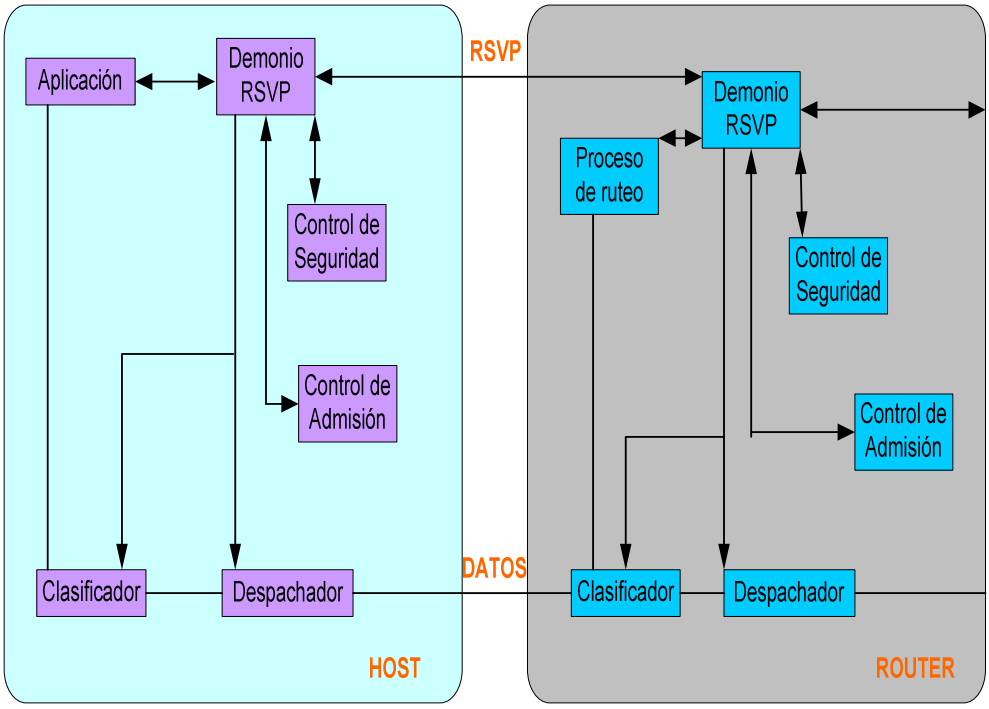
Dos arquitecturas básicas se han planteado:

1.6.4.1. IntServ SERVICIOS INTEGRADOS

Opera sobre flujos individuales reservando recursos suficientes en los routers de punta a punta, para satisfacer los requerimientos de QoS del flujo.

- Puede trabajar como unicast o multicast.
- Un router en el modelo IntServ debe ser capaz de proveer la QoS adecuada a cada flujo.
- Es necesario un protocolo para reservar recursos necesarios a lo largo de la ruta.

FIGURA 1.9. MODELO IntServ



Fuente: www.cisco.com

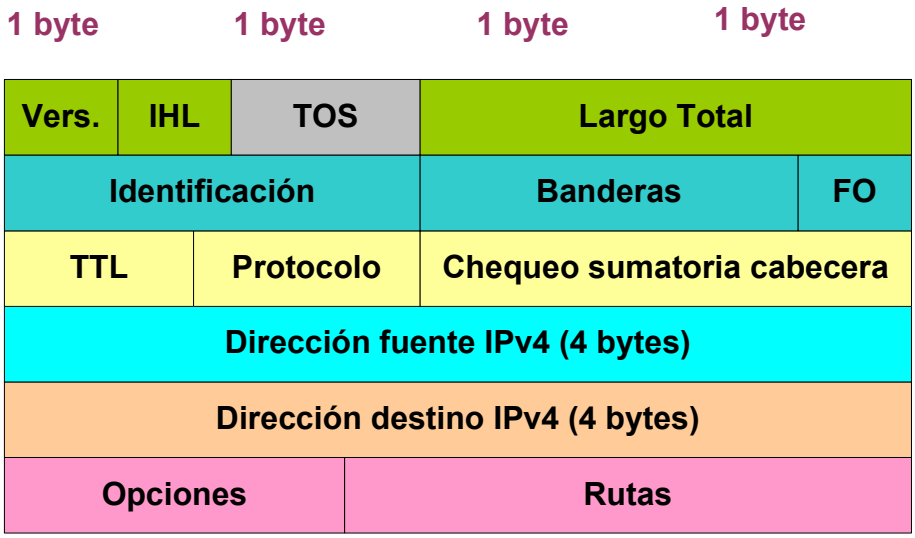
Elaborado por: Las autoras

1.6.4.2. DiffServ Servicios Diferenciados

Busca la diferenciación de servicios en IP de manera escalable y gestionable.

- Agregar el tráfico en conjuntos “grandes”
- IP: TOS à usa 6 bits para DSCP (Differentiated Service Code Point)

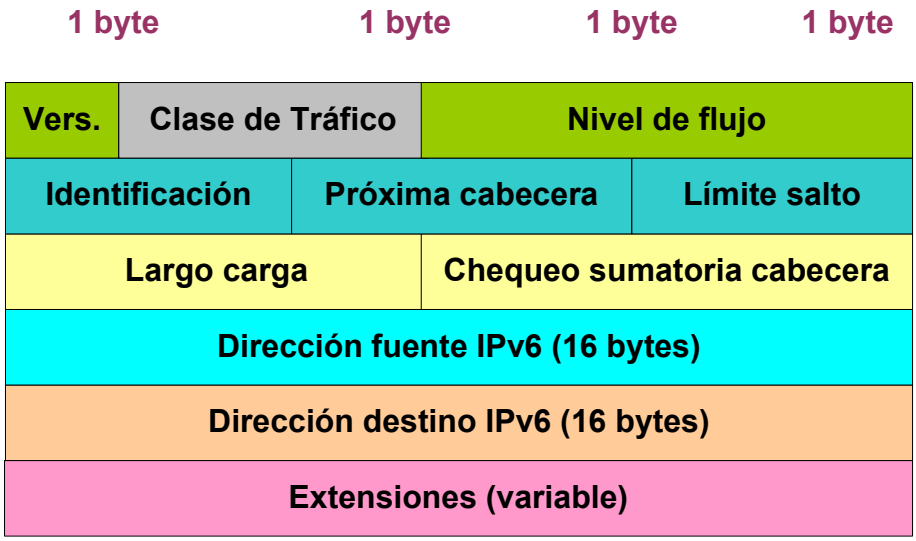
FIGURA 1.10. CAMPO TOS IPv4 (1)



Fuente: www.cisco.com

Elaborado por: Las autoras

FIGURA 1.11. CAMPO DE CLASE DE TRAFICO IPv6 (2)



Fuente: www.cisco.com

Elaborado por: Las autoras

- PHB define el tratamiento en cada nodo
- El DSCP es seteado en la frontera y en los routers internos es examinado para asociar el PHB
- La mayor complejidad residirá en los nodos exteriores
- Requiere SLAs (estático o dinámico)
- Service Level Agreement (SLA)

- Es un contrato entre un cliente y un proveedor de servicio
- Especificación de Nivel de Servicie

  Especifica el tráfico que el cliente puede mandar

  Especifica el compromiso del ISP con el cliente para los tráficos dentro y fuera del acuerdo
- Otras consideraciones contractuales

  Problemas con DiffServ: sobre agregación de tráfico por ruteo tradicional IP Es necesario realizar ingeniería de tráfico a MPLS

1.7. VENTAJAS

- Se considera a la arquitectura MPLS, base para la inclusión en la red de nuevas aplicaciones y para poder ofrecer diferentes niveles de servicio, en un entorno de mayor fiabilidad y con las garantías necesarias.

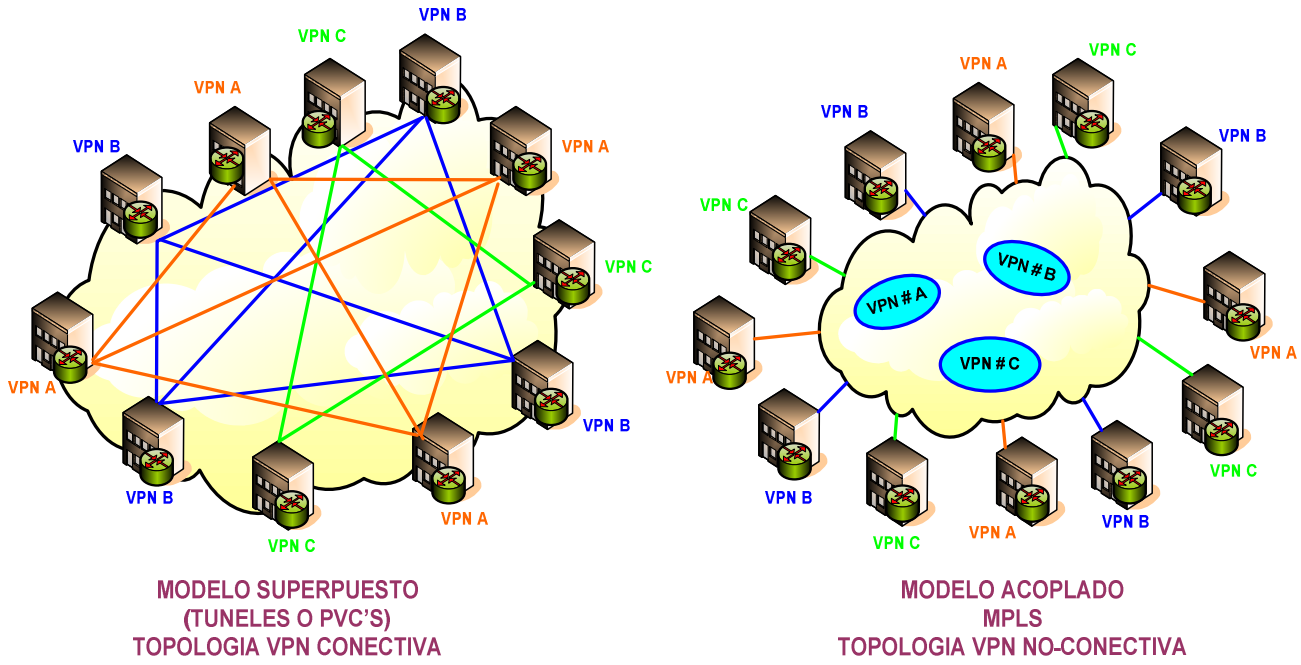
- MPLS presenta ventajas evidentes en la integración de los niveles 2 y 3 sin discontinuidades como:

  Ingeniería de tráfico, se puede hacer directamente sobre una red IP, al margen de que haya o no una infraestructura ATM por debajo, todo ello de manera más flexible y con menores costes de planificación y gestión para el administrador, y con mayor calidad de servicio para los clientes.

  Diferenciación de niveles de servicio mediante clases (CoS), MPLS está diseñado para poder cursar servicios diferenciados, según el Modelo DiffServ del IETF. MPLS se adapta perfectamente a ese modelo, ya que las etiquetas MPLS tienen el campo EXP para poder propagar la clase de servicio CoS en el correspondiente LSP.

  Servicio de redes privadas virtuales (VPN).

FIGURA 1.12. MODELO “SUPERPUESTO” (Túneles/PVC’s) Vs. MODELO “ACOPLADO” (MPLS)



Fuente: www.cisco.com

Elaborado por: Las autoras

En la figura 1.12. se presenta una comparación entre ambos modelos. La diferencia entre los túneles IP convencionales (o los circuitos virtuales) y los "túneles MPLS" (LSP’s) está en que éstos se crean dentro de la red, basados en LSP’s, y no de extremo a extremo a través de la red.

Las ventajas que MPLS ofrece para IP VPN’s son:

- Proporcionan un modelo "acoplado" o "inteligente", ya que la red MPLS "sabe" de la existencia de VPN’s (lo que no ocurre con túneles ni PVC’s).
- La provisión de servicio es sencilla: una nueva conexión afecta a un solo router tiene mayores opciones de crecimiento modular.
- Permiten mantener garantías QoS extremo a extremo, pudiendo separar flujos de tráfico por aplicaciones en diferentes clases, gracias al vínculo que mantiene el campo EXP de las etiquetas MPLS con las clases definidas a la entrada.
- Permite aprovechar las posibilidades de ingeniería de tráfico para poder garantizar los parámetros críticos y la respuesta global de la red (ancho banda, retardo, fluctuación, etc.), lo que es necesario para un servicio completo VPN.

1.8. CONVERGENCIA FIJO-MÓVIL

La necesidad de movilidad y servicio en áreas en las que el acceso a redes cableadas es inalcanzable por los costos que generaría su implementación, llevo al desarrollo de redes inalámbricas y una de ellas la red celular. Con el crecimiento de las redes nació también otra necesidad la de interconectividad entre las redes existentes, dando lugar a métodos de interacción.

La aplicación de nuevos estándares para una optimización de las redes cableadas y mas aún en el caso de MPLS, gracias a su independencia de la red física de transporte permitirá trasladar las ventajas ya comprobadas en redes fijas a una red celular, lo que permitirá satisfacer de mejor manera los requerimientos de los clientes facilitando de modo significativo la migración hacia la próxima generación de tecnologías de voz y datos.

CAPÍTULO II

“LA SEGURIDAD Y LAS REDES PRIVADAS VIRTUALES (VPN)”

CAPÍTULO DOS

2. LA SEGURIDAD Y LAS REDES PRIVADAS VIRTUALES (VPN)

MPLS (Multiprotocol Label Switching) proporciona un mecanismo de integración de redes heterogéneas, como IP, Frame relay y ATM; proporcionando capacidad de integrar Calidad de Servicio (QoS), VPN, abriendo un panorama de servicios que requieren menor inversión, cambios de hardware y potencia la red existente.

También se debe resaltar la necesidad de VPN's en el entorno de red empresarial actual, que tradicionalmente, al tener diversas sedes han unido sus redes locales a través de líneas dedicadas punto a punto, unas veces reales, y otras veces mediante circuitos virtuales dedicados (típicamente, mediante FR/ATM). Esta situación ha perdurado mucho tiempo, pero está claro que estamos ante una solución sub-óptima, que ha provocado que los operadores deban desarrollar y mantener dos infraestructuras totalmente diferenciadas, la de tráfico de Internet (sobre una red puramente IP) y la de circuitos (sobre ATM/FR). Esta situación es, a la vez, más compleja, más cara, y más ineficiente, puesto que no se aprovechan al máximo los recursos de la red.

En la actualidad, gracias a la aparición de MPLS, es factible fusionar ambas infraestructuras en un único dominio administrativo, pudiendo dar el servicio de Internet, el de circuitos clásicos y el de VPN sobre la misma red.

MPLS permite el aislamiento de la tecnología de nivel 2, y hacer converger desde el punto de vista de la conectividad lo mejor del mundo IP, con lo mejor del mundo de la conmutación de circuitos. Sin embargo, MPLS no resuelve por sí mismo los problemas asociados a las VPN, como la seguridad y GoS.

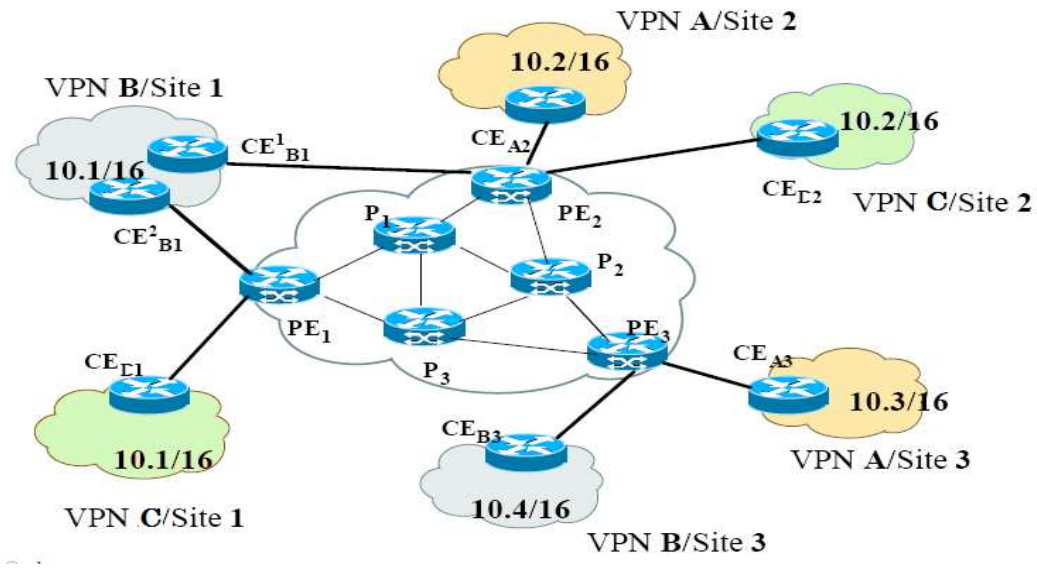
2.1. DEFINICIÓN DE VPN

Literalmente VPN = “Virtual Private Network” o Red Privada Virtual, es una red que utiliza una infraestructura pública compartida sea cual sea, para ofrecerle a un cliente las facilidades y ventajas de una red privada, por lo que cualquier red IP puede considerarse una VPN.

Las redes privadas virtuales son implementadas en routers (generalmente como parte de una solución firewall), ya que un dispositivo de VPN opera a nivel de red, a través de conexiones seguras utilizando encapsulación, encriptado y autenticación; de esta manera transportan de forma segura paquetes IP mediante Internet estableciendo túneles en ambos puntos de conexión que negocian un esquema de encriptado y autenticación previo al transporte.

Según el RFC 2547 BIS[20], se busca implementar VPN sobre MPLS para escalar un gran número de clientes, prestar mayor número de servicios de valor agregado y mejorar el uso de la infraestructura existente, con lo que se tiene un gran número de VPNs simultáneas.

FIGURA 2.1. MODELO GENERAL



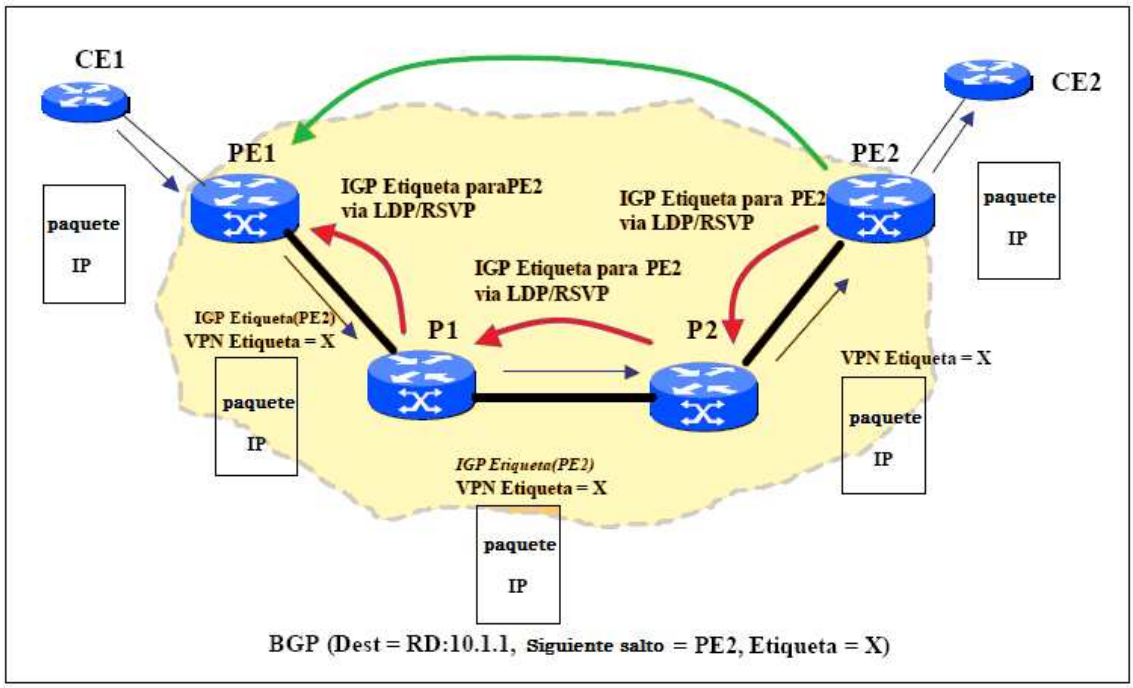
Fuente: www.cisco.com

Elaborado por: Las autoras

[20] Ver anexo

En la VPN C, si se envía un paquete de 1 a 2, CE[21]c1 lo etiqueta con un valor que representa a CEc2, durante el trayecto los demás sólo saben que es un paquete para CEc2 y solamente él retira la etiqueta y lee el resto para saber a donde enviar.

FIGURA 2.2. MODELO DE EQUIPOS QUE INTERVIENEN



Fuente: www.cisco.com

Elaborado por: Las autoras

2.2. CLASIFICACIÓN

Existe una gran variedad de VPN's, por lo que se las puede clasificar según varios aspectos, presentaremos los más relevantes:

- **SEGÚN EL PUNTO DE TERMINACIÓN**

Basadas en el CE (overlay), en este esquema se conectan CE a CE, como si el otro fuese el cliente final.

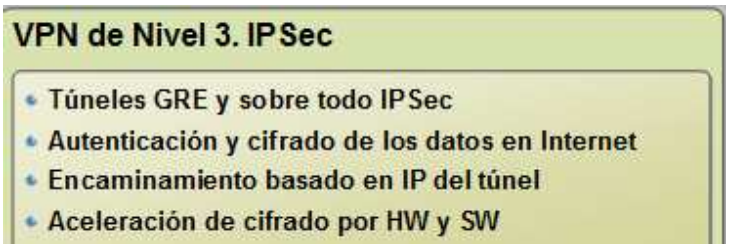
[21] CE:CLIENTE VPN
PE:PROVEEDOR VPN

Basadas en el PE (peer-to-peer), punto a punto.

Este esquema se utiliza para conectar oficinas remotas con la sede central de organización. El servidor VPN, que posee un vínculo permanente a Internet, acepta las conexiones vía Internet provenientes de los sitios y establece el túnel VPN. Los servidores de las sucursales se conectan a Internet utilizando los servicios de su proveedor local de Internet, típicamente mediante conexiones de banda ancha. Esto permite eliminar los costosos vínculos punto a punto tradicionales, sobre todo en las comunicaciones internacionales.

- **SEGÚN EL TRÁFICO DE CLIENTE TRANSPORTADO**

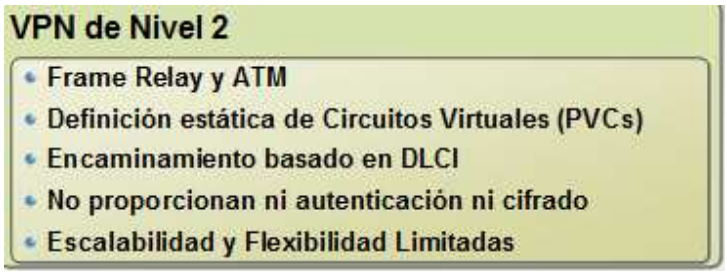
FIGURA 2.3. CARACTERÍSTICAS VPN IPSEC NIVEL 3



Fuente: www.cisco.com

Elaborado por: Las autoras

FIGURA 2.4. CARACTERÍSTICAS VPN NIVEL 2



Fuente: www.cisco.com

Elaborado por: Las autoras

- **SEGÚN EL TIPO DE RED DEL PROVEEDOR**

IP, IP/MPLS, ATM, Frame Relay, SONET/SDH, red telefónica, etc.

- **SEGÚN LA TECNOLOGÍA DE TÚNELES**

Frame Relay VC, SONET/SDH VT, PPP/Dial-up (conexiones físicas)
Túneles IPSec, L2TP, PPTP, MPLS-LSP, ATM-VP/VC

Point-to-Point Tunneling Protocol (PPTP): Permite que el tráfico IP, IPX o NetBEUI sea encriptado y encapsulado en encabezados IP para ser enviado a través de una red IP como Internet. Este Protocolo fue creado por Microsoft, y existe una implementación para Linux. Es un protocolo capa 2 que encapsula cuadros PPP en datagramas IP para ser transportados sobre una red IP, como Internet.

Layer 2 Forwarding (L2F): Es un protocolo de transmisión que permite a un servidor dial up encuadrar tráfico dial-up en PPP y transmitirlo sobre vínculos WAN a un servidor L2F. Este servidor desencapsula los paquetes y los inyecta a la red. En contraste con PPTP y L2TP, L2F no tiene un cliente definido. L2F es una tecnología propuesta por Cisco.

Layer 2 Tunneling Protocol (L2TP): Permite que el tráfico IP oIPX sea encriptado y enviado sobre cualquier medio que soporte entrega de datagramas punto-a-punto, tales como IP, X.25, Frame Relay o ATM- IP Security (IPSec) Tunnel Mode. Permite que paquetes IP sean encriptados y encapsulados en encabezados IP para ser enviados a través de una red IP.

L2TP es una combinación de PPTP y L2F. Cuando se configura para usar IP, L2TP puede ser usado como protocolo de tunneling sobre Internet, aunque también puede ser usado directamente sobre una WAN (como Frame Relay) sin una capa IP de transporte. L2TP sobre interredes IP hace uso de UDP para mantener el túnel.

- **Número de sedes conectadas**

Punto a punto: 2 sedes
Multipunto: más de dos sedes

- **SEGÚN SU EVOLUCIÓN**

1ª Generación: terminadas en el CE y basadas en líneas dedicadas que se alquilaban al proveedor.

2ª Generación: terminadas en el CE a base de circuitos virtuales ATM/Frame Relay sobre una red de conmutación de paquetes del proveedor.

3ª Generación: los proveedores ofrecen servicios para gestionar los routers del cliente usados en las terminaciones en el CE.

4ª Generación: VPN's de nivel 3 terminadas en el PE y basadas en IP/MPLS.

5ª Generación: VPN's de nivel 2 terminadas en el PE y basadas en IP/MPLS.

2.3. PRINCIPALES VALORES DE UNA VPN

2.3.1. ASPECTOS DE SEGURIDAD

2.3.1.1. TOPOLOGÍA DE LA SEGURIDAD

2.3.1.1.1. HOST-HOST

La implementación más sencilla de una VPN es de un host a otro. Por simplificar, asumiremos que los hosts están conectados por medio de ethernet a una LAN que después se conecta a Internet.

En una situación real, la comunicación se produciría a través de hubs, conmutadores, routers y nubes WAN. Si los hosts están conectados directamente mediante un cable ethernet de CAT5 (categoría 5), el único riesgo que se estaría mitigando con una VPN serían las escuchas sobre el tendido: una habilidad difícil de encontrar y de ejecutar.

En un escenario host-host, se tiene dos hosts conectados a Internet en su punto más íntimo, ya sea mediante una línea dedicada o mediante una conexión de marcado telefónico. La comunicación entre estos dos hosts no es segura y es

blanco de los hackers de Internet. Al implementar una VPN host-host, todas las comunicaciones entre ambos hosts quedan protegidas por el transporte VPN autenticado y cifrado.

2.3.1.1.2. HOST-RED

Un método fácil para ofrecer a los usuarios móviles la capacidad de conectarse con la red de la empresa es mediante una red virtual segura, o una VPN host-red.

Cada host se conecta independientemente con una LAN mediante una puerta de enlace VPN. Se autentica cada host, y los túneles VPN se inician para cada uno de ellos. El host móvil puede conectarse mediante cualquier tipo de conexión, ya sea de marcación telefónica, una conexión LAN o un enlace inalámbrico.

Este tipo se encuentra en situaciones de acceso remoto. Un usuario móvil puede tener software de VPN en su portátil y conectarse con la Intranet a través de una puerta de enlace VPN. También podemos utilizar esta topología VPN para los empleados que trabajan desde casa. El lento, pero constante, crecimiento de los clientes de ADSL y el cable hace que trabajar desde casa sea una opción atractiva.

Una VPN puede hacer que el tráfico sea privado e ilegible hasta que llega a la puerta de enlace VPN de la empresa.

2.3.1.1.3. RED-RED

La topología VPN es la red-red. En la que cada puerta de enlace se ubica en un extremo de una red y proporciona un canal de comunicación seguro entre las dos (o más) redes.

Este tipo de comunicación es el que mejor se adapta a la conexión de redes LAN separadas geográficamente, con la ventaja de esta configuración es que las LAN remotas de la VPN son transparentes para el usuario final. De hecho, las puertas de enlace VPN tienen la apariencia de routers para los usuarios.

Se puede utilizar las VPN red-red para conectar intranets, lo que hace que parezca que las redes son adyacentes. Los datos transferidos entre las intranets son confidenciales durante el tránsito. También se puede utilizar esta topología para extranets entre varias empresas, en caso de que cada empresa comparta recursos particulares sólo con los socios de negocio.

DEFENSA DE PERÍMETRO

Generalmente, los servidores VPN se encuentran situados detrás del firewall "perimetral" para proteger la red de la organización.

Existen dos clases de acceso por parte de usuarios de una red a los recursos de la otra, dependiendo de la confianza o acuerdo existente:

- Las comunicaciones entre ambos firewalls, a través de una VPN pueden ser efectuadas con acceso controlado o acceso abierto.

  En conexiones con acceso controlado, la VPN es utilizada solo para ofrecer privacidad entre ambos puntos, no existe una completa relación de confianza entre ambas partes, por lo que la comprobación de autenticidad se lleva a cabo para cada comunicación con el acceso a los recursos de la red restringido para ciertos servicios. En este caso se utiliza un firewall para controlar el acceso a la red interna.

  En conexiones con acceso abierto, la VPN es configurada para que ambos firewalls tengan un acceso completo a los recursos de la otra red. No se requiere un control de autenticidad por no considerarse necesario. En este esquema, el firewall realiza la función de conectividad mediante VPN, por lo que el tráfico es privado, y la confianza que ocasiona que todos los sitios son administrados por la misma organización, bajo las mismas políticas de seguridad, se podrá permitir todos los servicios de red sobre esta VPN. De esta forma, las transmisiones están bajo la protección del firewall, por lo que el "perímetro" de seguridad de la red se extiende a los sistemas remotos conectados mediante la VPN; todos estos sistemas se encuentran virtualmente en la misma red privada con un perímetro de red virtual.

- También es posible establecer una VPN entre un firewall y un sitio remoto simple para proveer acceso privado a usuarios móviles o conexiones hogareñas. De la misma forma que las conexiones anteriores, éstas pueden ser con acceso controlado o abierto.

  Acceso controlado es útil para clientes y socios que necesiten acceso a servicios o sistemas particulares.

  Acceso Abierto es útil para empleados o socios de la organización que necesiten tener acceso a recursos compartidos, como archivos, impresoras, unidades de almacenamiento masivo, etc.; en ambos casos estos servicios o recursos están situados dentro del perímetro de seguridad de la red. Mediante VPNs todas estas operaciones pueden realizarse de forma segura.

DEFENSA DE CANAL

Confidencialidad, integridad, autenticidad.

Para cumplir las condiciones anteriores, los paquetes IP que se desean transmitir:

- Se cifran para garantizar la confidencialidad
- Se firman para garantizar la autenticidad e integridad
- El paquete resultante se encapsula en un nuevo paquete IP y se envía a través de la red insegura al otro extremo de la VPN

FIGURA 2.5. NUEVO PAQUETE IP

Fuente: www.cisco.com

Elaborado por: Las autoras

2.3.1.2. CONFIDENCIALIDAD

Dado que los datos viajan a través de un medio potencialmente hostil como Internet, los mismos son susceptibles de intercepción, por lo que es fundamental el cifrado de los mismos. De este modo, la información no debe poder ser

interpretada por nadie más que los destinatarios de la misma. Se hace uso de algoritmos de cifrado como Data Encryption Standard (DES), Triple DES (3DES) y Advanced Encryption Standard (AES[22]).

NO REPUDIO POR FIRMA

Es decir, un mensaje tiene que ir firmado, y el que lo firma no puede negar que el mensaje lo envió él.

2.3.1.3. INTEGRIDAD

La garantía de que los datos enviados no han sido alterados. Para ello se utiliza funciones de Hash. Los algoritmos de hash más comunes son los Message Digest (MD2[23] y MD5[24]) y el Secure Hash Algorithm (SHA[25]).

2.3.1.4. AUTENTIFICACIÓN

Uno de los elementos más importantes de seguridad para una VPN es identificar al usuario. Esto es esencial para determinar a que recursos la persona está autorizada a usar. IPSec permite a los dispositivos usar un procedimiento llamado

---

[22] Descritos en la pagina 20.

[23] MD2 *Message-Digest Algorithm 2*, Algoritmo de Resumen del Mensaje 2, es una función de *hash* criptográfica desarrollada por Ronald Rivest en 1989. El algoritmo está optimizado para computadoras de 8 bits. El valor *hash* de cualquier mensaje se forma haciendo que el mensaje sea múltiplo de la longitud de bloque en el ordenador (128 bits o 16 bytes) y añadiéndole un *checksum*. Para el cálculo real, se utiliza un bloque auxiliar 48 bytes y una tabla de 256 bytes que contiene dígitos al azar del número pi. Una vez que todos los bloques del mensaje alargado se han procesado, el primer bloque parcial del bloque auxiliar se convierte en el valor de hash del mensaje.

[24] MD5 es uno de los algoritmos de reducción criptográficos diseñados por el profesor Ronald Rivest del MIT. Fue desarrollado en 1991 como reemplazo del algoritmo MD4 después de que Hans Dobbertin descubriese su debilidad. La sucesión de problemas de seguridad detectados desde que, en 1996, Hans Dobbertin anunciase una colisión de hash plantea una serie de dudas acerca de su uso futuro.

La codificación del MD5 de 128 bits es representada típicamente como un número de 32 dígitos hexadecimal. El siguiente código de 28 bytes ASCII será tratado con MD5 y veremos su correspondiente hash de salida:

```
MD5("Esto si es una prueba de MD5") = e07186fbff6107d0274af02b8b930b65
```

25 La familia SHA (*Secure Hash Algorithm*, Algoritmo de *Hash* Seguro) es un sistema de funciones *hash* criptográficas relacionadas de la Agencia de Seguridad Nacional de los Estados Unidos y publicadas por el *National Institute of Standards and Technology* (NIST). Existen cuatro variantes más que se han publicado desde entonces cuyas diferencias se basan en un diseño algo modificado y rangos de salida incrementados: SHA-224, SHA-256, SHA-384, y SHA-512 (llamándose SHA-2 a todos ellos).

SHA-1 ha sido examinado muy de cerca por la comunidad criptográfica pública, y no se ha encontrado ningún ataque efectivo. SHA-0 y SHA-1 producen una salida resumen de 160 bits de un mensaje que puede tener un tamaño máximo de 2^{64} bits, y se basa en principios similares a los usados por el profesor Ronald L. Rivest del MIT en el diseño de los algoritmos de resumen de mensaje MD4 y MD5.

La codificación *hash* vacía para SHA-1 corresponde a:

```
SHA1("") = da39a3ee5e6b4b0d3255bfef95601890afd80709
```

Intercambio de Llave de Internet (Internet Key Exchange, IKE) para transferir llaves de seguridad.

También se usa:

2.3.1.4.1. CRIPTOGRAFÍA DE CLAVE SIMÉTRICA

Son aquellos que utilizan la misma clave para cifrar y descifrar un documento. El principal problema de seguridad reside en el intercambio de claves entre el emisor y el receptor ya que ambos deben usar la misma clave.

Por lo tanto se tiene que buscar también un canal de comunicación que sea seguro para el intercambio de la clave. Es importante que sea muy difícil de adivinar. Por ejemplo el algoritmo de cifrado DES usa una clave de 56 bits, lo que significa que hay 72 mil billones de claves posibles. Ya se usan claves de 128 bits que aumentan el "espectro" de claves posibles (2 elevado a 128) de forma que aunque se uniesen todos los ordenadores existentes en estos momentos no lo conseguirían en miles de millones de años.

2.3.1.4.2. CRIPTOGRAFÍA DE CLAVE PÚBLICA

También son llamados sistemas de cifrado asimétrico. Este sistema de cifrado usa dos claves diferentes. Una es la clave pública y se puede enviar a cualquier persona y otra que se llama clave privada, que debe guardarse para que nadie tenga acceso a ella.
Para enviar un mensaje, el remitente usa la clave pública del destinatario para cifrar el mensaje. Una vez que lo ha cifrado, solamente con la clave privada del destinatario se puede descifrar, ni siquiera el que ha cifrado el mensaje puede volver a descifrarlo.

Un sistema de cifrado de clave pública basado en la factorización de números primos se basa en que la clave pública contiene un número compuesto de dos

números primos muy grandes. Para cifrar un mensaje, el algoritmo de cifrado usa ese compuesto para cifrar el mensaje.

Para descifrar el mensaje, el algoritmo de descifrado requiere conocer los factores primos, y la clave privada tiene uno de esos factores, con lo que puede fácilmente descifrar el mensaje.

Se recomienda en este caso que la clave pública tenga un mínimo de 1024 bits. Para un ataque de fuerza bruta, por ejemplo, sobre una clave pública de 512 bits, se debe factorizar un número compuesto de hasta 155 cifras decimales.

2.3.1.4.3. FIRMA DIGITAL

Una firma digital (o electrónica) es un sello digital que se añade a los datos que se envían. Al igual que en un sello de garantía, sirve para comprobar si alguien ha modificado el contenido.

También sirve para que un usuario se autentifique. Existen dos procesos fundamentales:

- Firma de los datos. Se lleva a cabo por medio de la clave privada, añadiendo la firma al mensaje enviado.
- Verificación de la firma. Se utiliza la clave pública. En este proceso se comprueba que el contenido no ha sido modificado.

2.3.1.5. AUTORIDADES DE CERTIFICACIÓN

- Mecanismos de encriptado: IPsec, PPTP, T2L, PT2L
- Algoritmos de encriptado: RC2 y RC4, DES y 3DES, IDEA, CAST

El cifrado de bloques (block cipher) es un método de encriptación de datos en que una llave criptográfica y un algoritmo son aplicados a un bloque de datos (por ejemplo, 64 bits contiguos) de una vez sobre todo el grupo, en lugar de aplicarlo a un bit cada vez.

Para evitar que mensajes idénticos encriptados de la misma forma no den el mismo resultado, un vector de inicialización, derivado de un generador de números aleatorios, se combina con los datos del primer bloque y la llave. Esto garantiza que al encriptar los siguientes bloques, se genere cada vez un mensaje encriptado diferente.

El cifrado de flujos (stream cipher) es, por contraposición al cifrado de bloques, un método de encriptación en el que se aplica la clave y el algoritmo de cifrado a cada dígito binario del flujo de datos, un bit de cada vez. No se usa habitualmente en criptografía moderna.

RSA (Rivest-Shamir-Adleman) es el algoritmo de encriptación y autentificación más comúnmente usado. Fue desarrollado en 1977 por Ron Rivest, Adi Shamir y Leonard Adleman, y se incluye como parte de los navegadores de Netscape y Microsoft, así como aplicaciones como Lotus Notes y muchos otros productos.

El funcionamiento de este algoritmo se basa en multiplicar dos números primos extremadamente grandes, y a través de operaciones adicionales obtener un par de números que constituyen la clave pública y otro número que constituye la clave privada. Una vez que se han obtenido las claves, los números primos originales ya no son necesarios para nada, y se descartan.

Se necesitan tanto las claves públicas como las privadas para encriptar y desencriptar, pero solamente el dueño de la clave privada lo necesitará. Usando el sistema RSA, la clave privada nunca necesitará ser enviada. La clave privada se usa para desencriptar el código que ha sido encriptado con la clave pública. Por tanto, para enviar un mensaje a alguien, hay que conocer su clave pública, pero no su clave privada.

También se puede usar para autentificar un mensaje, firmando con la clave privada un certificado digital.

AES (Advanced Encryption Standard) es un algoritmo de encriptación para proteger información delicada, aunque no clasificada, por las agencias gubernamentales de USA y, como consecuencia, puede transformarse en el estándar de facto para las transacciones comerciales en el sector privado.

La especificación solicitaba un algoritmo simétrico usando encriptación por bloques de 128 bits de tamaño, que soportara como mínimo claves de 128, 192 y 256 bits.

DES (Data Encryption Standard) es un método de encriptación de clave privada muy usado. El gobierno de USA restringió su exportación a otros países debido a su estimación de la dificultad para reventarlo por hackers. Hay 72 cuatrillones (72,000,000,000,000,000) o más de posibles claves. Para cada mensaje, se elige una clave al azar entre todas esas posibilidades. Es un método de encriptación simétrico, lo que obliga a que tanto el emisor como el receptor hayan de conocer la clave privada. DES aplica una clave de 56 bits a cada bloque de 64 bits de datos. El proceso se puede ejecutar en diferentes modos e implica 16 turnos de operaciones.

Aunque está considerado como un algoritmo de encriptación fuerte, muchas organizaciones usan "triple DES", o sea aplicar 3 claves de forma sucesiva. Esto no quiere decir que un mensaje encriptado por DES no pueda ser reventado. DES fue desarrollado por IBM en 1977. La clave del algoritmo DES tiene un valor de 64 bits, 8 de los cuales se usan para comprobar la paridad, y 56 para el algoritmo de encriptación.
Triple DES (3DES) usa tres claves de 56 bits, un total de 168 bits. DES es un tipo de cifrado de los conocidos como Red Feistel Tradicional.

IDEA (International Data Encryption Algorithm) es un algoritmo de encriptación desarrollado en el ETH de Zurich (Suiza) por James Massey y Xuejia Lai. Usa criptografía de bloque con una clave de 128 bits, y se suele considerar como muy seguro. Está considerado como uno de los algoritmos más conocidos. El uso no comercial de IDEA es gratuito.

Blowfish es un algoritmo de encriptación que puede usarse como sustituto de DES y de IDEA. Es simétrico y encripta bloques, con una clave de longitud variable, desde 32 bits hasta 448 bits. Fue diseñado en 1993 por Bruce Schneier como una alternativa a los algoritmos existentes entonces, y con procesadores de 32 bits en mente, lo que lo hace significativamente más rápido que DES.

CAST-128 es un algoritmo de encriptación del mismo tipo que DES. Es un criptosistema SPN (Substitution-Permutation Network) que parece tener buena resistencia contra ataques diferenciales, lineales y related-key, Pertenece a la clase de algoritmos denominada como cifrados Feistel, y su mecanismo, de 4 pasos, es similar al DES.

TEA (Tiny Encryption Algorithm) es uno de los algoritmos de encriptación más rápidos y eficientes que existen. Fue desarrollado por David Wheeler y Roger Needham en el Computer Laboratory de Cambridge University. Consiste en un cifrado Feistel que usa operaciones de grupos algebraicos mixtos (ortogonales), XORs y sumas en este caso. Encripta bloques de 64 bits usando una clave de 128 bits. Parece altamente resistente al criptoanálisis diferencial y consigue difusión total (una diferencia de un bit en el mensaje original causa aproximadamente 32 bits de diferencia en el mensaje cifrado) en solamente 6 pasos. Es tan seguro como IDEA.

- Mecanismos de negociación e intercambio de claves para encriptado: ISAKMP[26], SKIP[27]
- Algoritmos utilizados para intercambiar claves para el encriptado: RSA[28], Diffie-Hellman[29]

[26] ISAKMP (Internet Security Association and Key Management Protocol) es un protocolo para el establecimiento de Asociaciones de Seguridad (SA) y claves criptográficas en un entorno de Internet.
Define los procedimientos para la autenticación de una comunicación entre pares, la creación y la gestión de las Asociaciones de Seguridad, las técnicas de generación de claves, y la amenaza de mitigación (por ejemplo, la denegación de servicio y ataques replay). Normalmente utiliza IKE para intercambio de claves, aunque otros métodos se pueden aplicar. ISAKMP está documentado en el RFC 2048:VER ANEXO

[27] Una skip list o lista por saltos es una Estructura de datos, basada en Listas enlazadas paralelas con eficiencia comparable a la de un árbol binario (tiempo en orden O(log n) para la mayoría de las operaciones).

Todos estos mecanismos deben funcionar en forma coordinada para poder integrar una eficiente funcionalidad para una VPN.

Otros valores de seguridad:

FIREWALL

Un dispositivo de seguridad que provee una barrera fuerte entre la red privada y el Internet. Puede instalárselo para restringir el número de puertos abiertos, el tipo de paquetes que pueden pasar y que protocolos son permitidos.

Una buena solución VPN debe proporcionar un dispositivo de seguridad ("firewall") repleto de funciones basado en la tecnología de Inspección de Estado de Paquetes ("Stateful Packet Inspection") y Traducción de Dirección de Red (Network Address Translation, NAT) para protección contra intrusos y ataques de Rechazo de Servicio (Denial of Service, DoS). Además, un dispositivo de seguridad ("firewall") puede proveer seguridad a nivel aplicación usando proxy y filtros para bloquear contenidos específicos de Internet.

PROTECCIÓN DE VIRUS

Los virus de computadora son una de las amenazas líderes de seguridad para redes conectadas a Internet. Los virus pueden también ser usados como mecanismo de entrega de utilerías pirata, comprometiendo la seguridad de la red, aunque un dispositivo de seguridad ("firewall") este instalado. Una buena solución de VPN debe proveer una protección de virus en tiempo real utilizando un alto

[28] El sistema criptográfico con clave pública RSA es un algoritmo asimétrico cifrador de bloques, que utiliza una clave pública, la cual se distribuye (en forma autenticada preferentemente), y otra privada, la cual es guardada en secreto por su propietario.

[29] El protocolo **Diffie-Hellman** (debido a Whitfield Diffie y Martin Hellman) permite el intercambio secreto de claves entre dos partes que no han tenido contacto previo, utilizando un canal inseguro, y de manera anónima (no autenticada). Se emplea generalmente como medio para acordar claves simétricas que serán empleadas para el cifrado de una sesión. Siendo no autenticado, es bases para varios protocolos autenticados. Su seguridad radica en la extrema dificultad (conjeturada, no demostrada) de calcular logaritmos discretos en un campo finito.

desempeño, un probador ICSA y un procesador anti-virus que revise entradas y salidas de correo electrónico.

FILTRADO DE CONTENIDO

El filtrado de contenido le permite a las compañías controlar que información puede y no puede ser accesada desde sus computadoras. El bloqueo de URL, basado en una lista de filtro con actualización frecuente, es el método preferido de filtrado de contenido porque bloquea contenido censurable mientras conserva el acceso a recursos valiosos de Internet.

2.4. CALIDAD DE SERVICIO

QoS o Calidad de Servicio (Quality of Service, en inglés) son las tecnologías que garantizan la transmisión de cierta cantidad de datos en un tiempo dado (throughput). Calidad de servicio es la capacidad de dar un buen servicio.

2.4.1. GESTIÓN

2.4.1.1. CONCEPTO DE DOMINIO

En Internet es un nombre base que agrupa a un conjunto de equipos o dispositivos y que permite proporcionar nombres de equipo más fácilmente recordables en lugar de una dirección IP numérica.

Permiten a cualquier servicio (de red) moverse otro lugar diferente en la topología de Internet, que tendrá una dirección IP diferente sin que sea de total evidencia para el usuario final.

2.4.1.2. TUNNELING

Es un método que consiste en utilizar la infraestructura de una interred (como Internet), para transportar datos de una red a otra. Los datos a ser transportados pueden ser los cuadros (o paquetes) de un protocolo diferente al que maneje la interred en cuestión, es decir, en lugar de enviar un cuadro tal y como fue

producido por el nodo que lo originó, el protocolo de tunneling (ya sea L2TP, IPSec, etc.) encapsula el cuadro en un header (encabezado) adicional que pertenece al protocolo de transporte de la interred sobre la cual se establece el túnel (por ejemplo, IP). Los paquetes encapsulados son entonces enrutados sobre la interred entre los extremos del túnel. A esa ruta lógica a través de la cual viajan los paquetes encapsulados sobre la interred se le llama 'túnel'. Cuando los paquetes (o cuadros) encapsulados llegan a su destino, el paquete es desencapsulado y reenviado a su destino final.

2.6. IPSec

Es un estándar de Internet, no funciona sobre TCP/UDP sino directamente sobre IP, permite validación de usuarios por medio de:

- Certificados X509 (SSL)
- Claves secretas compartidas por ambos extremos
- Claves RSA

Es complejo de configurar, tiene problemas con NAT, los mismos que están resueltos en las nuevas versiones (NAT-T). Para la negociación de la conexión utiliza el protocolo ISAKMP.

SISTEMAS DE SEGURIDAD EN CANAL DE COMUNICACIÓN-GENERALIDADES

Los protocolos de IPsec actúan en la capa de red, la capa 3 del modelo OSI. Otros protocolos de seguridad para Internet de uso extendido, como SSL, TLS y SSH operan de la capa de transporte (capas OSI 4 a 7) hacia arriba.

Esto hace que IPsec sea más flexible, ya que puede ser utilizado para proteger protocolos de la capa 4, incluyendo TCP y UDP, los protocolos de capa de transporte más usados. IPsec tiene una ventaja sobre SSL y otros métodos que operan en capas superiores. Para que una aplicación pueda usar IPsec no hay

que hacer ningún cambio, mientras que para usar SSL y otros protocolos de niveles superiores, las aplicaciones tienen que modificar su código.

ALGORITMO Y CLAVE

Los algoritmos criptográficos definidos para usar con IPsec incluyen HMAC-SHA1 para protección de integridad, y Triple DES-CBC y AES-CBC para confidencialidad. Más detalles en la RFC 4305[30].

2.6. MODOS DE TRABAJO: TÚNEL Y TRANSPORTE

2.6.1. MODO TÚNEL

En el modo túnel, todo el paquete IP (datos más cabeceras del mensaje) es cifrado y/o autenticado. Debe ser entonces encapsulado en un nuevo paquete IP para que funcione el enrutamiento.

El modo túnel se utiliza para comunicaciones red a red (túneles seguros entre routers, PE. para VPNs) o comunicaciones ordenador a red u ordenador a ordenador sobre Internet.

2.6.2. MODO TRANSPORTE

En modo transporte, sólo la carga útil (los datos que se transfieren) del paquete IP es cifrada y/o autenticada. El enrutamiento permanece intacto, ya que no se modifica ni se cifra la cabecera IP; sin embargo, cuando se utiliza la cabecera de autenticación (AH), las direcciones IP no pueden ser traducidas, ya que eso invalidaría el hash. Las capas de transporte y aplicación están siempre aseguradas por un hash, de forma que no pueden ser modificadas de ninguna manera, como traduciendo los números de puerto TCP y UDP.

[30] Ver en los anexos

El modo transporte se utiliza para comunicaciones ordenador a ordenador.
Una forma de encapsular mensajes IPsec para atravesar NAT ha sido definido por RFCs que describen el mecanismo de NAT-T.

2.6.2.1. COMPONENTES FUNDAMENTALES DE IPSEC: SA (SECURITY ASSOCIATION)

Dos protocolos han sido desarrollados para proporcionar seguridad a nivel de paquete, tanto para IPv4 como para IPv6:

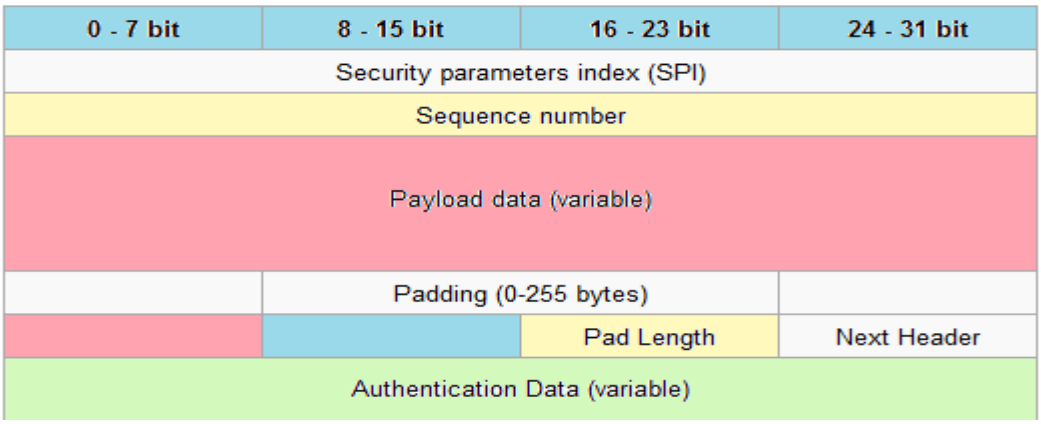
2.6.2.1.1. ENCAPSULATING SECURITY PAYLOAD (ESP)

Proporciona confidencialidad, además de opcionalmente (pero altamente recomendable) autenticación y protección de integridad.

El protocolo ESP proporciona autenticidad de origen, integridad y protección de confidencialidad de un paquete. ESP también soporta configuraciones de sólo cifrado y sólo autenticación, pero utilizar cifrado sin autenticación está altamente desaconsejado porque es inseguro.

Al contrario que con AH, la cabecera del paquete IP no está protegida por ESP (aunque en ESP en modo túnel, la protección es proporcionada a todo el paquete IP interno, incluyendo la cabecera interna; la cabecera externa permanece sin proteger). ESP opera directamente sobre IP, utilizando el protocolo IP número 50.

FIGURA 2.6. DIAGRAMA DE PAQUETE ESP



Fuente: www.cisco.com

Elaborado por: Las autoras

Significado de los campos:

(SPI)	Identifica los parámetros de seguridad en combinación con la dirección IP.
Sequence number	Un número siempre creciente, utilizado para evitar ataques de repetición.
Payload data	Los datos a transferir.
Padding	Usado por algunos algoritmos criptográficos para rellenar por completo los bloques.
Pad Length	Tamaño del relleno en bytes.
Next Header	Identifica el protocolo de los datos transferidos.
Authentication Data	Contiene los datos utilizados para autenticar el paquete.

2.6.2.1.2. AUTHENTICATION HEADER (AH)

Proporciona integridad y autenticación y no repudio, si se eligen los algoritmos criptográficos apropiados. AH está dirigido a garantizar integridad sin conexión y autenticación de los datos de origen de los datagramas IP. Además, puede opcionalmente proteger contra ataques de repetición utilizando la técnica de ventana deslizante y descartando paquetes viejos.

AH protege la carga útil IP y todos los campos de la cabecera de un datagrama IP excepto los campos mutantes, es decir, aquellos que pueden ser alterados en el tránsito.

En IPv4, los campos de la cabecera IP mutantes (y por lo tanto no autenticados) incluyen TOS, Flags, Offset de fragmentos, TTL y suma de verificación de la cabecera. AH opera directamente por encima de IP, utilizando el protocolo IP número 51.

FIGURA 2.7. DIAGRAMA DE AUTHENTICATION HEADER (AH)

0 - 7 bit	8 - 15 bit	16 - 23 bit	24 - 31 bit
Next header	Payload length	RESERVED	
Security parameters index (SPI)			
Sequence number			
Authentication data (variable)			

Fuente: www.cisco.com

Elaborado por: Las autoras

Significado de los campos:

Next header	Identifica el protocolo de los datos transferidos
Payload length	Tamaño del paquete AH.
RESERVED	Reservado para uso futuro (hasta entonces todo ceros).

(SPI) Indica los parámetros de seguridad que, en combinación con la dirección IP, identifican la asociación de seguridad implementada con este paquete.

Sequence number Un número siempre creciente, utilizado para evitar ataques de repetición.

Authentication data Contiene el valor de verificación de integridad (ICV) necesario para autenticar el paquete; puede contener relleno.

2.6.3. PROPÓSITO DE IPSEC

IPsec fue diseñado para proporcionar seguridad en modo transporte (extremo a extremo) del tráfico de paquetes, en el que los ordenadores de los extremos finales realizan el procesado de seguridad, o en modo túnel (puerta a puerta) en el que la seguridad del tráfico de paquetes es proporcionada a varias máquinas (incluso a toda la red de área local) por un único nodo.

Por ello IPsec puede utilizarse para crear VPNs en los dos modos, y este es su uso principal. Hay que tener en cuenta, sin embargo, que las implicaciones de seguridad son bastante diferentes entre los dos modos de operación.

La seguridad de comunicaciones extremo a extremo a escala Internet se ha desarrollado más despacio de lo esperado. Parte de la razón a esto es que no ha surgido infraestructura de clave pública universal o universalmente de confianza (DNSSEC[31] fue originalmente previsto para esto); otra parte es que muchos usuarios no comprenden lo suficientemente bien ni sus necesidades ni las opciones disponibles como para promover su inclusión en los productos de los vendedores.

[31] Domain Name System Security Extensions (DNSSEC), es de importancia crítica para asegurar el Internet en su conjunto, pero el despliegue se ha visto obstaculizada por la dificultad de:

- La elaboración de un estándar compatible hacia atrás-que puede aumentar con el tamaño de la Internet.
- DNSSEC despliegue de implementaciones a través de una amplia variedad de servidores DNS y la resolución de (clientes).
- Discrepancias entre los principales protagonistas, que no se ponen de acuero sobre cual es la auténtica

Como el Protocolo de Internet no provee intrínsecamente de ninguna capacidad de seguridad, IPsec se introdujo para proporcionar servicios de seguridad tales como:

- Cifrar el tráfico (de forma que no pueda ser leído por nadie más que las partes a las que está dirigido)
- Validación de integridad (asegurar que el tráfico no ha sido modificado a lo largo de su trayecto)
- Autenticar a los extremos (asegurar que el tráfico proviene de un extremo de confianza)
- Anti-repetición (proteger contra la repetición de la sesión segura).

IKE (Internet Key Exchange) es un servicio de negociación automático y de gestión de claves, usado en los protocolos IPsec.
Supone una alternativa al intercambio manual de claves. Permite, además, especificar el tiempo de vida de la sesión IPSEC, autenticación dinámica de otras máquinas, etc.

La mayoría de las implementaciones de IPsec consisten en un dominio IKE que corre en el espacio de usuario y una pila IPsec dentro del kernel que procesa los paquetes IP.

El protocolo IKE usa paquetes UDP, normalmente a través del puerto 500, y generalmente requiere entre 4 y 6 paquetes con dos turnos para crear una SA en ambos extremos. Las claves negociadas son entregadas a la pila IPsec. Se define en rfc 2409[32]

[32] Ver anexo

2.7. CONSIDERACIONES ECONÓMICAS A TENER EN CUENTA A LA HORA DE DISEÑAR UNA VPN

- Costes directos, indirectos y ocultos

La principal motivación del uso y difusión de esta tecnología es la reducción de los costos de comunicaciones directos, tanto en líneas dial-up como en vínculos WAN dedicados. Los costos se reducen drásticamente en estos casos:

- En el caso de accesos remotos, llamadas locales a los ISP (Internet Service Provider) en vez de llamadas de larga distancia a los servidores de acceso remoto de la organización. O también mediante servicios de banda ancha.
- En el caso de conexiones punto a punto, utilizando servicios de banda ancha para acceder a Internet, y desde Internet llegar al servidor VPN de la organización. Todo esto a un costo sensiblemente inferior al de los vínculos WAN dedicados.

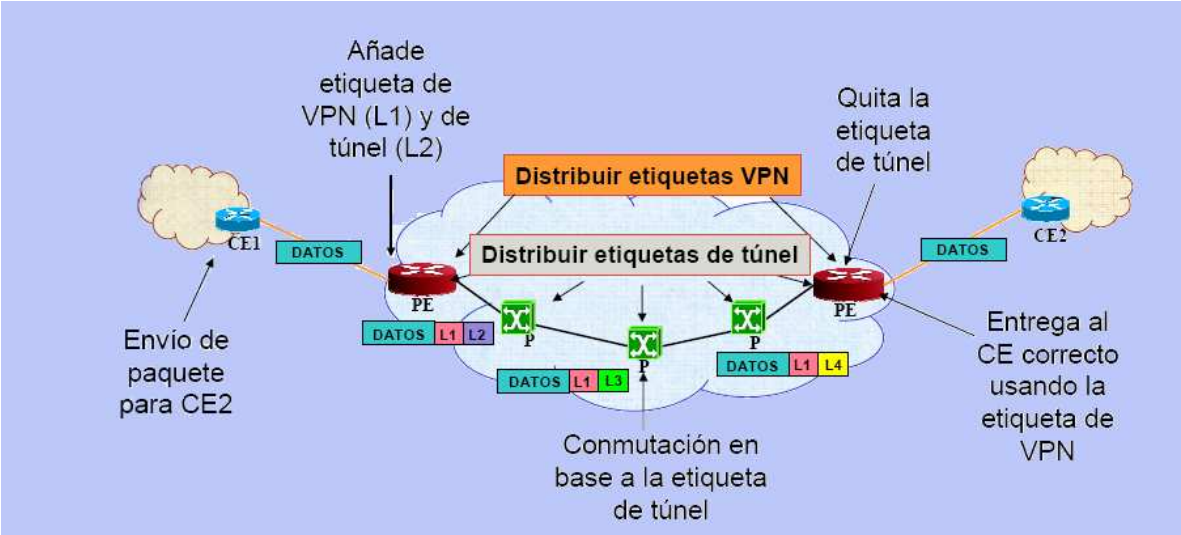
2.8. BGP

Border Gateway Protocol es un protocolo mediante el cual intercambia información de encaminamiento entre Sistemas Autónomos; conjunto de routers dirigidos por la misma autoridad y que usan un protocolo interno de distribución y actualización de información de encaminamiento (IGP).

Por ejemplo, los ISP registrados en Internet suelen componerse de varios sistemas autónomos. Actualmente entre los sistemas autónomos de los ISP se intercambian sus tablas de rutas a través del protocolo BGP. Este intercambio de información de encaminamiento se hace entre los routers externos de cada sistema autónomo. Estos routers deben soportar BGP. Se trata del protocolo más utilizado para redes con intención de configurar un EGP (external gateway protocol)

Cada sistema autónomo (AS) tendrá conexiones o, mejor dicho, sesiones internas (iBGP) y además sesiones externas (eBGP).

FIGURA 2.8. DISTRIBUCIÓN DE ETIQUETAS DE CIRCUITO VIRTUAL

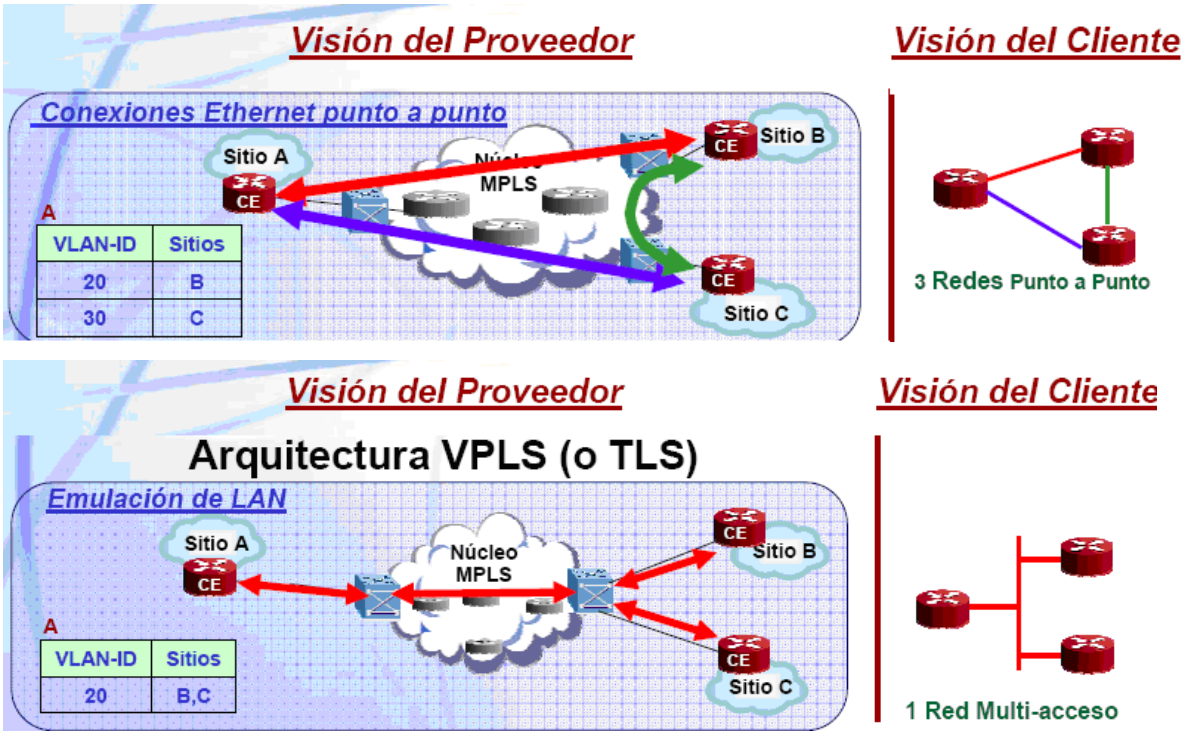


Fuente: www.cisco.com

Elaborado por: Las autoras

En esta figura 2.8. se puede ver como se intercambian etiquetas de túnel y de vpn.

FIGURA 2.9. CONEXIÓN PUNTO A PUNTO Y VPLS



Fuente: www.cisco.com

Elaborado por: Las autoras

2.9. ROUTERS VIRTUALES: VRF

Es una tecnología que se emplea en las redes de computadoras que permite que varias instancias de una tabla de enrutamiento puedan co-existir en el mismo router a la vez. Debido a que el enrutamiento de los casos es independiente, la misma o superposición de las direcciones IP se pueden utilizar sin entrar en conflicto unos con otros.

VRF podrán ejecutarse en un dispositivo de red por tener distintas tablas de enrutamiento, también la conocida como Bases de Transmisión de Información (FIBs). Por otra parte, un dispositivo de red puede tener la capacidad de configurar diferentes routers virtuales, donde cada uno tiene su propia FIB que no es accesible a cualquier otra instancia del virtual router en el mismo dispositivo.

Tecnología VRF se encuentra comúnmente en el mercado proveedor de acceso a Internet, especialmente en las configuraciones VPN MPLS.

En un entorno MPLS, sólo el Proveedor Edge (PE), tienen conmutadores de enrutamiento y conoce de estas múltiples características de enrutamiento virtual. Cliente Edge (CE) dispositivos que participar en la red MPLS VPN de enrutamiento a través de la difusión de la ruta y del PE conoce un único motor de enrutamiento ejemplo el apoyo a la aplicación de cliente VPN. Normalmente esto se hace a través de eBGP, OSPF o enrutamiento estático.

2.10. EJEMPLO

En este ejemplo, los ruteadores son configurados como PE routers (Router B y Router E) mientras CE routers (Router y Router F) y como PE routers de tipo AS frontera routers (Router C and Router D).

Un router está configurado como un router CE (utilizando la ruta declaración de los casos) en la configuración del router para B. Debido a que el intercambio de rutas VPN-IPv4, y el Router D, C se configuran como routers PE.

Configurar router B:

[Editar]
Protocolos (
RSVP (T3-0/0/0.0 interfaz;)
MPLS (De conmutación de etiqueta de ruta para routerC-(A 10.255.14.171;
Descripción "para routerC-para su uso con VPNs";)
T3-0/0/0.0 interfaz;
So-1/2/0.0 interfaz;)
(BGP Grupo al-ibgp (Tipo interior; Local-address 10.255.14.175; Familia inet-VPN (Unicast;) 10.255.14.171 vecino;))
OSPF (De la ingeniería de tráfico; Ancho de banda de referencia 4g; Área 0.0.0.0 (T3-0/0/0.0 interfaz; Interfaz lo0.0 (Pasivos;))))

Ejemplo de tipo vrf;
So-1/2/0.0 interfaz;
Ruta-10.255.14.175:9 distintivo;
Vrf-vpna importación a la importación;
Vrf-vpna exportación a la exportación;
Protocolos ((BGP Grupo al-ce (
Como peer-9;
Vecino 192.168.198.1;)))

- Opciones de política (Política de declaración de importación-vpna (
Plazo 1 (De (Protocolo BGP; Comunidad vpna-Com;)
Entonces aceptar;)
Plazo 2 (Entonces rechazar;))
Vpna declaración política-a la exportación (
Plazo 1 (De protocolo BGP; Entonces (
Comunidad añadir vpna-comm;
Aceptar;))
Plazo 2 (Entonces rechazar;))
Comunidad vpna-comm miembros objetivo: 100:1001;)

En el protocolo BGP de configuración del router para C, se debe mantener toda declaración. Cuando esta declaración se incluye, BGP debe almacenar cada ruta aprendida a través de BGP. Configurar dos sesiones BGP (configurar familia inet-vpn en ambos períodos de sesiones):

- Período de sesiones IBGP a router D (grupo al-ibgp en este ejemplo)
- EBGP período de sesiones a router B (grupo al-ebgp-pe en este ejemplo)

Interfaz t3-0/2/0 se añade al [editar protocolos mpls] nivel jerárquico, lo que permite anunciar rutas BGP con etiquetas durante el período de sesiones EBGP.

Configurar router C:

[Editar]
Protocolos (RSVP (T3-0/2/0.0 interfaz;)
MPLS (De conmutación de etiqueta de ruta para routerB-(A 10.255.14.175;
Descripción "para routerB-para su uso con vpns";)
T3-0/2/0.0 interfaz;
So-0/0/0.0 interfaz;)
(BGP Guardar todos; Grupo al-ibgp (Tipo interior; Local-address
10.255.14.171;
Familia inet-VPN (Unicast;)
10.255.14.175 vecino;)
Grupo al-ebgp-pe (Tipo externo; Familia inet-VPN (Unicast;)
192.168.197.22 vecino (Como peer-10045;)))
OSPF (De la ingeniería de tráfico; Ancho de banda de referencia 4g;
Área 0.0.0.0 (T3-0/2/0.0 interfaz; Interfaz lo0.0 (Pasivos;))))

La configuración de Router D es casi idéntica a la del router C:

[Editar]
Protocolos (RSVP (Fe-1/1/0.0 interfaz;)
MPLS (De conmutación de etiqueta de camino hacia la E-(A 10.255.14.177;
Descripción "para-routerE para vpna";)
Fe-1/1/0.0 interfaz;

So-0/1/0.0 interfaz;)
(BGP Guardar todos; Grupo al-ibgp-pe (Tipo interior;
Familia inet-VPN (Unicast;) 10.255.14.177 vecino;)
Grupo al-ebgp-pe (Tipo externo; Familia inet-VPN (Unicast;)
Como peer-10023;
192.168.197.21 vecino;))
OSPF (De la ingeniería de tráfico; Ancho de banda de referencia 4g;
Área 0.0.0.0 (Fe-1/1/0.0 interfaz;
Interfaz lo0.0 (Pasivos;))))

La configuración del router para E es muy similar a la de configuración del router B.

En el grupo al-ibgp, incluir a la familia inet etiquetados-unicast declaración a pasar etiquetados IPv4 rutas, y configurar un EBGP multihop período de sesiones para aprobar las rutas VPN-IPv4:

[Editar]
Protocolos ((BGP Grupo al-ibgp (Tipo interior;
Local-address 10.255.14.175;
Familia inet (Etiquetados-unicast (Resolver-VPN;))
10.255.14.171 vecino;)
Grupo-a distancia-pe (Multihop (TTL 10;)
Familia inet-VPN (Unicast;)
10.255.14.177 vecino (Como peer-10045;))

CAPÍTULO III

“INGENIERÍA DE TRÁFICO (IT) Y OBTENCIÓN DE ESTADÍSTICAS”

CAPÍTULO TRES

3.1. INGENIERÍA DE TRÁFICO (IT) Y OBTENCIÓN DE ESTADÍSTICAS

El objetivo básico de la ingeniería de tráfico es adaptar los flujos de tráfico a los recursos físicos de la red. La idea es equilibrar de forma óptima la utilización de esos recursos, de manera que no haya algunos que estén supra-utilizados, con posibles puntos calientes y cuellos de botella, mientras otros puedan estar infrautilizados.

A comienzos de los 90 los esquemas para adaptar de forma efectiva los flujos de tráfico a la topología física de las redes IP eran bastante rudimentarios. Los flujos de tráfico siguen el camino más corto calculado por el algoritmo IGP correspondiente. En casos de congestión de algunos enlaces, el problema se resolvía a base de añadir más capacidad a los enlaces.

La ingeniería de tráfico consiste en trasladar determinados flujos seleccionados por el algoritmo IGP sobre enlaces más congestionados, a otros enlaces más descargados, aunque estén fuera de la ruta más corta (con menos saltos).

El camino más corto entre A y B según la métrica normal IGP es el que tiene sólo dos saltos, pero puede que el exceso de tráfico sobre esos enlaces o el esfuerzo de los routers correspondientes haga aconsejable la utilización del camino alternativo indicado con un salto más. MPLS es una herramienta efectiva para esta aplicación en grandes backbones, ya que:

- Permite al administrador de la red el establecimiento de rutas explícitas, especificando el camino físico exacto de un LSP.
- Permite obtener estadísticas de uso LSP, que se pueden utilizar en la planificación de la red y como herramientas de análisis de cuellos de botella y carga de los enlaces, lo que resulta bastante útil para planes de expansión futura.

- Permite hacer "encaminamiento restringido" (Constraint-based Routing, CBR), de modo que el administrador de la red pueda seleccionar determinadas rutas para servicios especiales (distintos niveles de calidad). Por ejemplo, con garantías explícitas de retardo, ancho de banda, fluctuación, pérdida de paquetes, etc.

La ventaja de la ingeniería de tráfico MPLS es que se puede hacer directamente sobre una red IP, al márgen de que haya o no una infraestructura ATM por debajo, todo ello de manera más flexible y con menores costes de planificación y gestión para el administrador, y con mayor calidad de servicio para los clientes.

3.2. TÚNELES MPLS: TIPOS, ESTABLECIMIENTO, CREACIÓN Y CONFIGURACIÓN

TIPOS

La Ingeniería de Tráfico en MPLS combina la capacidad de tráfico de redes ATM con la flexibilidad de IP y la diferenciación de clases de servicio. Así como ATM maneja circuitos virtuales MPLS maneja LSPs llamados túneles, de esta manera el tráfico se enruta hacia un destino determinado. Este método permite un mayor control que si se enrutará tráfico tomando como base solamente la IP destino.

A pesar de las ventajas de los túneles IP sobre los PVCs, éstos son menos eficientes frente a la solución MPLS:

- Están basados en conexiones punto a punto (PVCs o túneles)
- La configuración es manual
- La provisión y gestión son complicadas; una nueva conexión supone alterar todas las configuraciones
- Plantean problemas de crecimiento al añadir nuevos túneles o circuitos virtuales
- La gestión de QoS es posible en cierta medida, pero no se puede mantener extremo a extremo a lo largo de la red, ya que no existen mecanismos que sustenten los parámetros de calidad durante el transporte.

Realmente, el problema que plantean estas IP VPNs es que están basadas en un modelo topológico superpuesto sobre la topología física existente, a base de túneles extremo a extremo (o circuitos virtuales) entre cada par de routers de cliente en cada VPN. De ahí las desventajas en cuanto a la poca flexibilidad en la provisión y gestión del servicio, así como en el crecimiento cuando se quieren añadir nuevos emplazamientos.

Con una arquitectura MPLS se obvian estos inconvenientes ya que el modelo topológico no se superpone sino que se acopla a la red del proveedor. En el modelo acoplado MPLS, en lugar de conexiones extremo a extremo entre los distintos emplazamientos de una VPN, lo que hay son conexiones IP a una "nube común" en las que solamente pueden entrar los miembros de la misma VPN. Las "nubes" que representan las distintas VPNs se implementan mediante los caminos LSPs creados por el mecanismo de intercambio de etiquetas MPLS. Los LSPs son similares a los túneles en cuanto a que la red transporta los paquetes del usuario (incluyendo las cabeceras) sin examinar el contenido, a base de encapsularlos sobre otro protocolo.

Aquí está la diferencia: en los túneles se utiliza el encaminamiento convencional IP para transportar la información del usuario, mientras que en MPLS esta información se transporta sobre el mecanismo de intercambio de etiquetas, que no ve para nada el proceso de routing IP. Sin embargo, sí se mantiene en todo momento la visibilidad IP hacia el usuario, que no sabe nada de rutas MPLS sino que ve una internet privada (intranet) entre los miembros de su VPN. De este modo, se pueden aplicar técnicas QoS basadas en el examen de la cabecera IP, que la red MPLS podrá propagar hasta el destino, pudiendo así reservar ancho de banda, priorizar aplicaciones, establecer CoS y optimizar los recursos de la red con técnicas de ingeniería de tráfico.

La diferencia entre los túneles IP convencionales (o los circuitos virtuales) y los "túneles MPLS" (LSPs) está en que éstos se crean dentro de la red, a base de LSPs, y no de extremo a extremo a través de la red. Este tipo de túnel es el utilizado en redes MPLS.

ESTABLECIMIENTO

Para la configuración de MPLS IT se deben cumplir los siguientes prerrequisitos:

- Una versión de software IOS que soporte Ingeniería de Tráfico en MPLS
- Habilitar en la red Cisco Express Forwarding (CEF)
- Un protocolo de estado de enlace (OSPF o IS-IS) como IGP
- Habilitar Ingeniería de Tráfico de manera global en el router
- Una interfaz de loopback que será utilizada como router ID
- Configuración básica de túneles e Ingeniería de Tráfico

Las versiones de Cisco que soportan IT dependiendo del protocolo utilizado son:
OSPF: software Cisco IOS 12.0(5)S y 12.0(5)T
IS-IS: software Cisco IOS 12.0(7)S y 12.0(8)T

Las versiones posteriores de Cisco (incluyendo 12.0S, 12.1, 12.2, etc.) soportan MPLS IT. La necesidad de un protocolo de estado de enlace está íntimamente ligada con el cálculo del path y su configuración como se verá más adelante.

CREACIÓN Y CONFIGURACIÓN

Si no se tiene configurado como protocolo de enrutamiento OPSF o IS-IS no será posible implementar MPLS IT. Adicionalmente, se requiere una interfaz a ser utilizada como router ID. Esta debe ser una interfase de loopback. La utilidad de tener este tipo de interfase es que la misma está siempre arriba sin importar el estado de las otras interfases.

Esta interfaz debe tener configurada una máscara /32 (255.255.255.255) y su configuración es la siguiente:

interface Loopback0
ip address 192.168.1.1 255.255.255.255

El proceso MPLS IT tiene que ser habilitado en todos los ruteadores en donde se desea que este participe en MPLS IT. Esto no tiene que ser en todos los ruteadores en la red, típicamente en algunos o todos los ruteadores del core.

La forma de configurarlo en el router es usando el comando **"mpls traffic-eng tunnels"**, de manera global así como en cada interfase por donde posiblemente pase el túnel IT.

Se recomienda no habilitarlo en interfaces que hacen cara contra los clientes, si se desea correr MPLS IT en una caja donde hay clientes conectados, solo se debe habilitar IT en las interfaces conectadas a la red.

Una propiedad de MPLS Traffic Engeniering es poder controlar por donde los cruzan los túneles, entre muchas una de ellas son los Flags de Atributos, este flag mide 32 bits y es configurado dentro de la interfaces:

Router(config-if)#mpls traffic-eng attribute-flags ?

Se pueden configurar los valores de los atributos como mejor convenga, por ejemplo se pueden decidir que un atributo en particular es una interfase satelital, o por ejemplo un link con low delay. Para la comprobación de que MPLS IT está configurado se pueden usar las siguientes opciones:

Router#show mpls traffic-eng tunnels summary
Signalling Summary:
LSP Tunnels Process: running
RSVP Process: running
Forwarding: enabled

Si MPLS IT está activo se obtendrán los resultados mostrados en el ejemplo, caso contrario se debe revisar. Otra opción utilizada para la configuración es:

Router#show mpls interfaces

Interface	IP	Tunnel	Operational
POS0/0	Yes (ldp)	Yes	Yes
POS3/0	Yes (ldp)	Yes	Yes
POS5/0	Yes (ldp)	Yes	Yes

Si la columna de túnel dice SI entonces esta interfase tiene habilitado MPLS IT.

Los comandos que se siguen para la configuración de un túnel son.

```
interface Tunnel0
        ip unnumbered Loopback0
        tunnel mode mpls traffic-eng
        tunnel destination destination-ip
        tunnel mpls traffic-eng path-option 10 dynamic
```

Los túneles MPLS IT son representados como una interfase túnel en el software Cisco IOS. Desde esta perspectiva, este tipo de túnel es igual a un túnel GRE o cualquier otro tipo de túnel que se pueda configurar.

ATRIBUTOS DE TRÁFICO Y DE RECURSOS

En lo que respecta al atributo de tráfico y recursos MPLS IT permite:

- Establecer rutas explícitas, orientando el tráfico hacia donde hay más recursos y realizando balanceo de carga.

- Permite obtener estadísticas del uso de un determinado LSP, esto se realiza mediande el análisis de los “cuellos de botella” y proyectando la futura expansión de la red.

- Ruteo Basado en Restricciones (CBR), determinando el LSP a seguir en función de los atributos de tráfico, los recursos de la red y la topología, y garantizando QoS.

SEÑALIZACIÓN RSVP-IT

Las nuevas aplicaciones que están surgiendo en Internet han producido un aumento de la necesidad de transmitir información desde un origen a múltiples destinos (multidifusión o multicast) y que esta transmisión se haga garantizado ciertos parámetros de Calidad de Servicio (QoS), por ejemplo, el retardo máximo y el número de paquetes que pueden ser descartados sin afectar a la calidad de la transmisión de la información. Esta QoS no puede ser asegurada por los protocolos TCP/IP, por lo que se han desarrollado diferentes tecnologías para superar este inconveniente, entre ellas RSVP y MPLS. RSVP (Resource Reservation Protocol) es un protocolo de señalización que para un flujo específico reserva recursos a lo largo de un camino entre el nodo origen y el nodo destino lo que le permite garantizar la QoS.

MPLS tiene especificado el funcionamiento de los protocolos de señalización CR-LDP (Constraint-Route Label Distribution Protocol) y RSVP-TE (Resource Reservation Protocol – Traffic Engineering) para asegurar parámetros de QoS, como por ejemplo, la reserva de recursos y el retardo máximo para un flujo de información.

Para aplicaciones en las que es necesario asegurar un retardo máximo y una cantidad máxima de paquetes descartados en la transmisión desde el origen hasta los múltiples destinos del grupo (por ejemplo video stream o videoconferencia) es necesaria la utilización de protocolos de señalización para la transmisión de paquetes IP sobre MPLS.

En el caso de la transmisión sin QoS, para establecer los LSPs (Label Switch Path), se puede utilizar cualquiera de los protocolos de enrutamiento multidifusión IP más una señalización sin garantías de QoS, como RSVP o LDP (Label

Distribution Protocol). Para la transmisión con QoS, CR-LDP y RSVP-TE son usados como protocolos de señalización.

Para la aplicación de protocolo RSVP-TE a la transmisión de información multidifusión IP se definen nuevos objetos a los mensajes de señalización para establecer el árbol de etiquetas a cada uno de los destinos del grupo de multidifusión. A los mensajes PATH (realiza la petición de etiquetas), RESV (realiza la devolución del valor de la etiqueta) y HELLO (para reconocimiento de vecinos) se le añaden unos nuevos objetos. Se está proponiendo habilitar funciones de multidifusión para ser independientes de los tradicionales protocolos de enrutamiento multidifusión tales como: DVMRP, MOSPF, PIM, etc.

ANÁLISIS DE TRÁFICO EN LA RED

El análisis de tráfico se realizó en dos etapas. La primera se efectuó cuando todavía no se implementaba MPLS en la red de datos. Para salida de Internet de la empresa existían dos caminos uno por Colombia (salida de tráfico de Quito) y otro por Perú (salida de tráfico de Guayaquil).

El ancho de banda de cada uno de estos enlaces se muestra en la siguiente tabla:

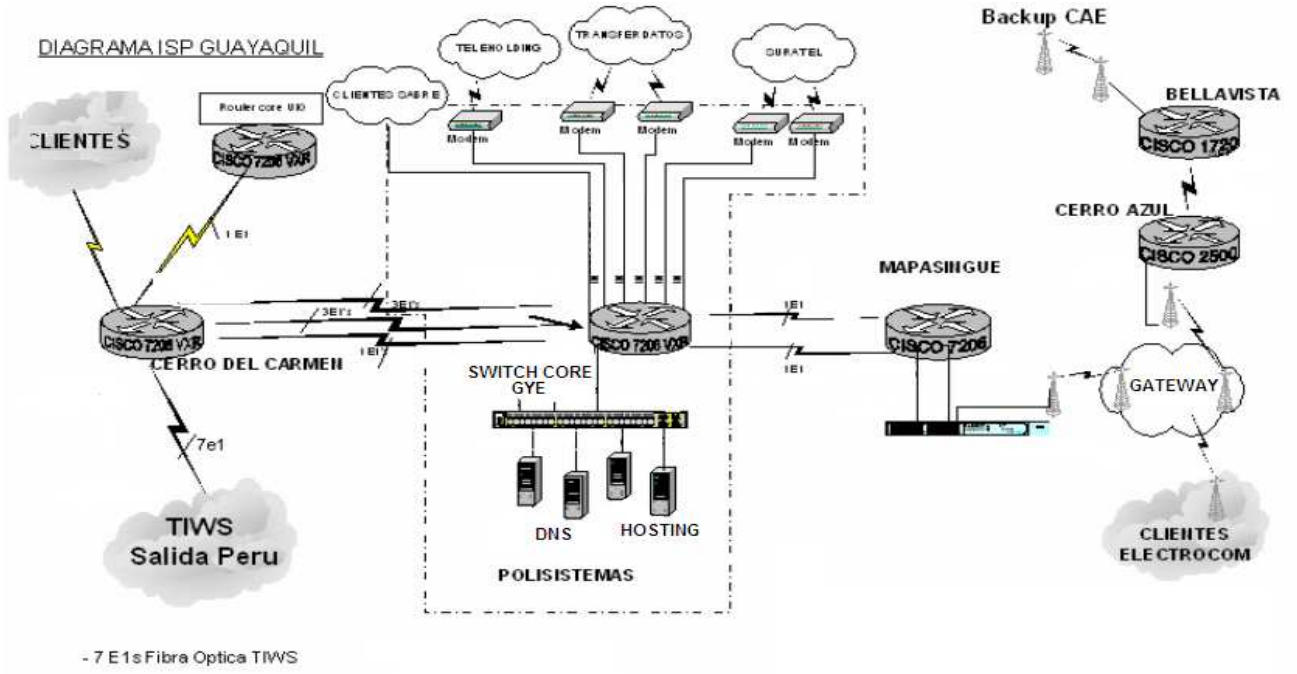
CUADRO 3.1. DATOS DE ENLACE

UIO	GYE
46Mbps	7E1

Fuente: Empresa

Elaborado por: Las autoras

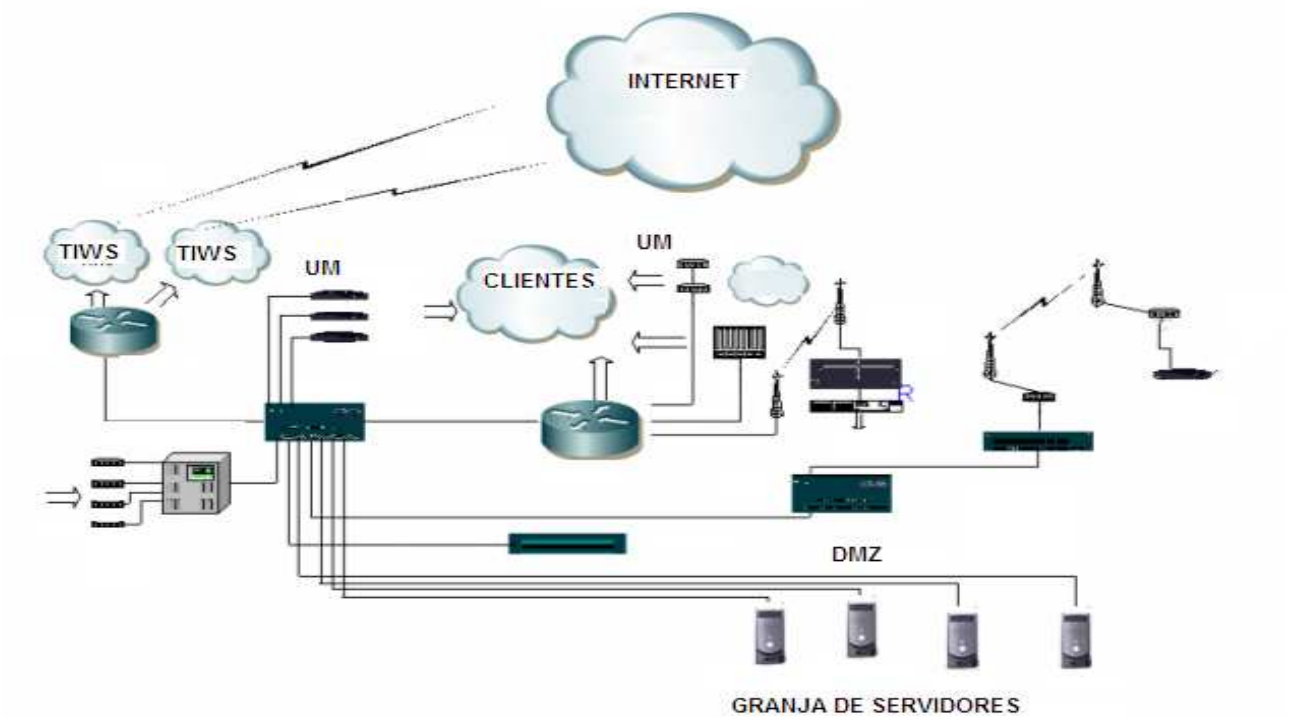
FIGURA 3.1. DIAGRAMA DE SALIDA INTERNET GYE SIN MPLS



Fuente: Empresa

Elaborado por: Las autoras

FIGURA 3.2. DIAGRAMA DE SALIDA INTERNET UIO SIN MPLS



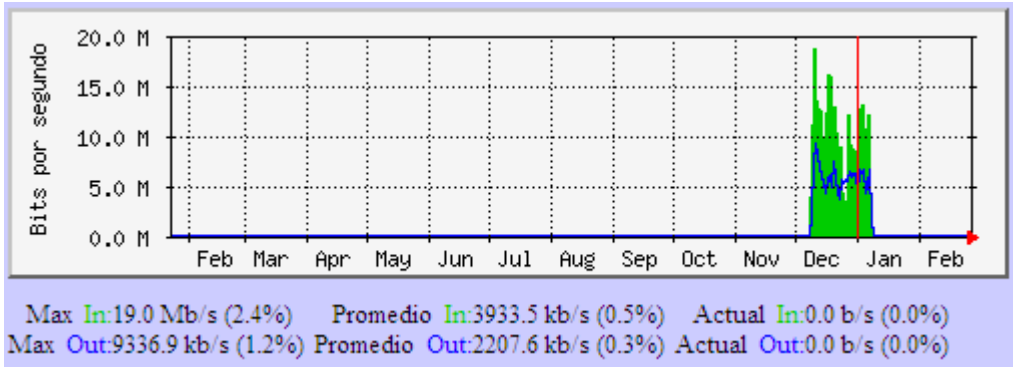
Fuente: Empresa

Elaborado por: Las autoras

Para la salida a Internet, no existían redundancias y se tenía una sobreventa de la capacidad real de la red. Por este motivo, cuando uno de los clientes generaba un tráfico excesivo se requería reenrutar al resto para que no presenten en sus enlaces tiempos promedios elevados y degradación del canal.

Es por esto que se vió la necesidad de implementar MPLS en la red de datos e incrementar el AB de la salida internacional. Actualmente, con el diseño de red implementado en el capítulo IV el canal no registra saturación como se observa en las siguientes gráficas.

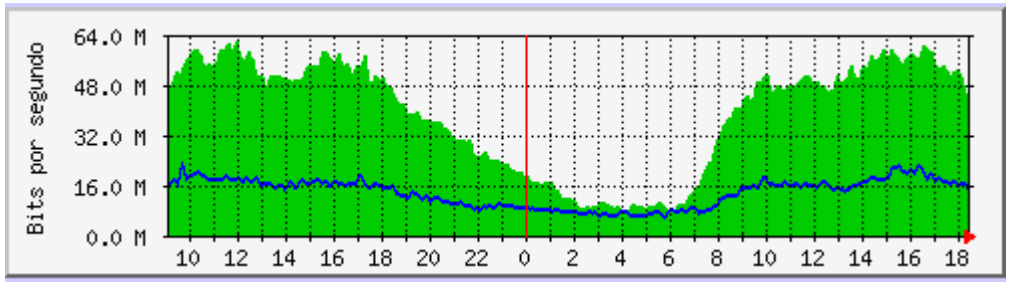
FIGURA 3.3. TRÁFICO ACTUAL EN LA SALIDA INTERNET GYE



Fuente: MRTG (THE MULTI ROUTER TRAFFIC GRAPHER)

Elaborado por: Las autoras

FIGURA 3.4. TRÁFICO ACTUAL EN LA SALIDA INTERNET UIO



Fuente: MRTG (THE MULTI ROUTER TRAFFIC GRAPHER)

Elaborado por: Las autoras

Como se puede ver en la red implementada en la actualidad con MPLS, el tráfico ni siquiera alcanza el ancho de banda contratado con el proveedor internacional, así como tampoco se observa saturación en ninguno de los canales de Internet.

ESTADÍSTICAS (HERRAMIENTAS DE PLANIFICACIÓN PARA TOMA DE DECISIONES FUTURAS Y OPTIMIZACIÓN)

MPLS es una herramienta efectiva para esta aplicación en un backbone, ya que permite al administrador establecer rutas explícitas, obtener datos estadísticos de uso LSP usadas para planificación de la red y análisis de cuellos de botella y carga de enlaces, y por último, hacer enrutamiento restringido, de modo que el administrador puede determinar rutas para servicios especiales. Se utiliza el protocolo IT-RSVP que permite reservar ancho de banda.

CAPÍTULO IV

“DISEÑO DE UNA RED MPLS Y LA CONFIGURACIÓN DE LOS EQUIPOS”

CAPÍTULO CUATRO

4.1. DISEÑO DE UNA RED MPLS Y LA CONFIGURACIÓN DE LOS EQUIPOS

Se realizará el diseño de la red con MPLS presentando esquemas, diagramas y configuración de todos los equipos de la red.

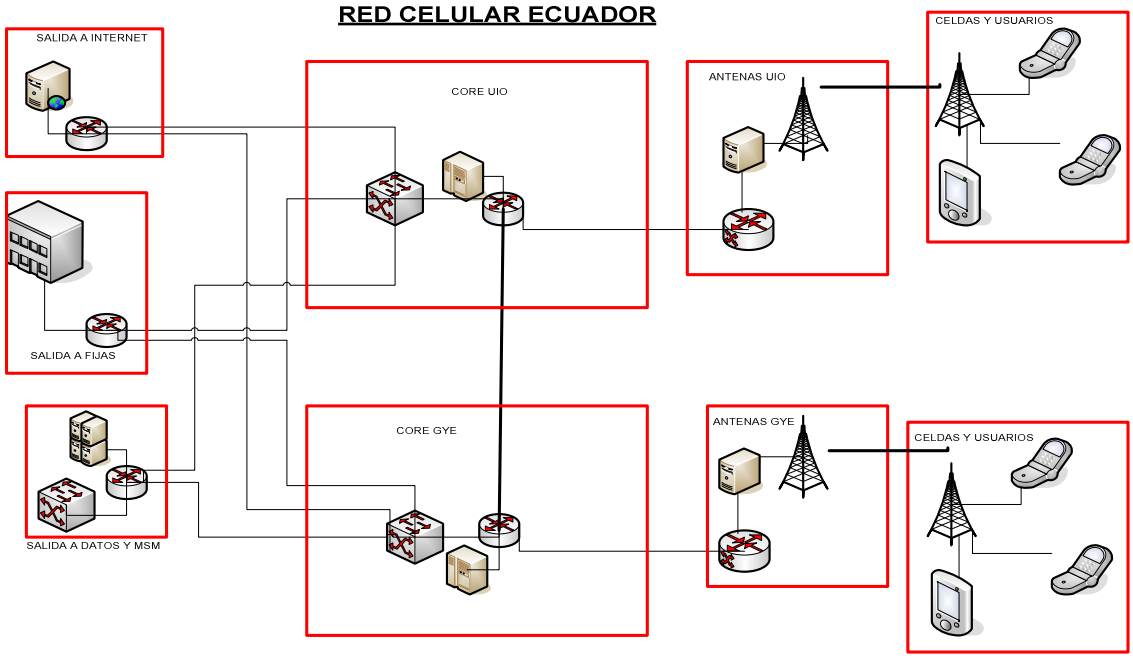
Para iniciar presentaremos la red actual celular en la que se pretende implantar MPLS.

La red consta de:

- Un core principal uio
- Un core secundario antenas uio
- Un core principal gye
- Un core secundario antenas gye
- Un core de datos y mensajería
- Un core de acceso a Internet
- Un core de facturación uio
- Un core de acceso a red telefónica local
- Dos enlaces directos uio-gye
- Enlaces a Colombia y Perú desde uio y gye respectivamente
- Routers de enlace entre cores

El CORE, puede ser una red interna o un servidor y se asume que el trabajo interno es óptimo.

FIGURA 4.1. DIAGRAMA BÁSICO DE RED

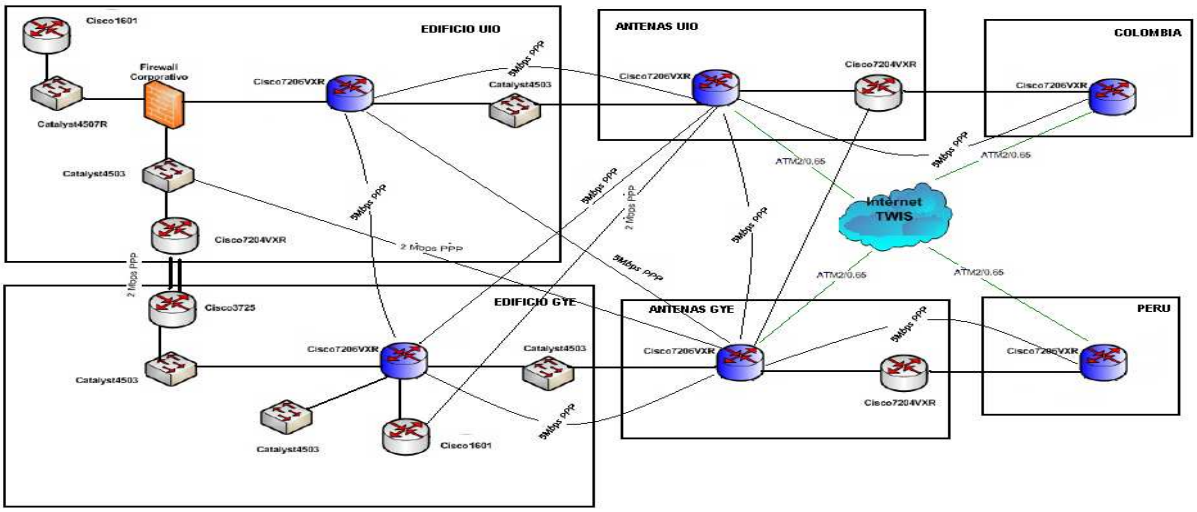


Fuente: Red aproximada a los datos reales de la empresa

Elaborado por: Las autoras

Cada enlace en este diagrama reducido de la figura 4.1., difiere de los otros en tecnología, capacidad, distancia, redundancia física, el de la figura 4.2., muestra tipos y capacidades de enlaces típicos así:

FIGURA 4.2. DIAGRAMA CON ENLACES DE RED



Fuente: Red aproximada a los datos reales de la empresa

Elaborado por: Las autoras

Ahora se debe aclarar como se va a garantizar el servicio, con uso de técnicas activas simultáneas de dos tipos:

1. Mecanismo de recuperación local de paquetes con requerimiento de garantía de servicio en entorno punto a punto en casos puntuales donde la congestión provoque la pérdida de paquetes privilegiados.

2. Recuperación local de LSP, permite establecer caminos alternativos en caso de caída accidental, con las mismas características para reconducir el tráfico privilegiado.

En el primer caso se usará algoritmo RABAN (Algoritmo de Encaminamiento en Redes Activas, buscando la mejor opción que tenga menor retardo, saturación de red y mayoría de nodos activos a atravezar) para la información con GoS y los demás con algoritmo Floyd estándard, donde el retardo es el que define el peso del enlace. GPSRP (Protocolo de Almacenamiento y Retransmisión de Paquetes Privilegiados), usa una memoria DMGP (está formada por la clave primaria que tiene todo paquete IP privilegiado más un identificador único que éste coloca) para posible retransmisión local si hay pérdida, con capacidad de confirmar o negar la retransmisión.

En el segundo caso, RLPRP (Protocolo de Recuperación Flexible de Caminos Locales), permite establecer caminos de seguridad para que en caso de caída de un LSP principal y mientras ésta persista se use el camino backup instantáneamente.

TLDP es un modo reducido pero funcional del LDP, que permite conexión, confirmación y desconexión tanto de un LSP como de su BACKUP.

EPCD es el algoritmo de captura y desechado anticipado de paquetes, controla los buffer e informa a GPSRP si debe retransmitir un paquete privilegiado por pérdida.

Para iniciar el diseño se comienza por definir y dar nombre a los equipos.

TABLA 4.1. ENRUTADORES P Y PE EN EL BACKBONE MPLS

Nombre enrutador	Modelo	Localización	Nodo
core-uio	7206VXR	Quito	Matriz
antenas-uio	7204VXR	Pichincha	Matriz
core-PERU	7206VXR	Quito	Carretas
antenas-gye	7206VXR	Cerro Carmen	Guayaquil
core-gye	7206VXR	Guayaquil	Guayaquil

Fuente: Nomenclatura propuesta

Elaborado por: Las autoras

Capacidades necesarias para los enlaces que se requieren entre los nodos para formar el backbone MPLS.

TABLA 4.2. CAPACIDAD DE ENLACES REQUERIDA

Enlace	Equipo A	Equipo B	Capacidad
Core-uio - core-PERU	Core-uio	core-PERU	100M
Core-uio - Cerro Carmen	antenas-uio	antenas-gye	1 E1
Guayaquil- Cerro Carmen	core-gye	antenas-gye	100M
Guayaquil- core-PERU	core-gye	core-PERU	3 E1´s
Guayaquil- Core-uio	core-gye	Core-uio	3 E1´s

Fuente: Empresa

Elaborado por: Las autoras

TABLA 4.3. DIAGRAMA CON ABREVIATURAS DE EQUIPOS GENERAL

Tipo de equipo			Función		
Enrutador	Switch	Firewall	core	acceso	Gestión
Rt	sw	Fw	p	pe	mg

Fuente: Abreviatura propuesta por las autoras

Elaborado por: Las autoras

TABLA 4.4. DIAGRAMA CON ABREVIATURAS DE LUGARES

Localización						
País	**Ciudad**		**Nodo**			
Ecuador	Quito	Guayaquil	Core-uio antenas	core-PERU	core-gye	antenas gye
Ec	uio	gye	Mtz	crt	Pst	crm

Fuente: Abreviatura propuesta

Elaborado por: Las autoras

TABLA 4.5. DIAGRAMA CON ABREVIATURAS DE EQUIPOS ESPECÍFICOS

Equipo	Nombre
Router PE core-gye	rtpe01_pst
Router PE antenas-gye	rtpe01_crm
Router PE core-uio	rtpe01_mtz
Router PE core-PERU	rtpe01_crt
Switch acceso core-gye	swpe01_pst
Switch acceso antenas-gye	swpe01_crm
Switch acceso core-uio	Swpe01_mtz
Switch acceso core-PERU	Swpe01_crt

Fuente: Abreviatura propuesta

Elaborado por: Las autoras

TABLA 4.6. DIAGRAMA CON ABREVIATURAS DE CONEXIONES

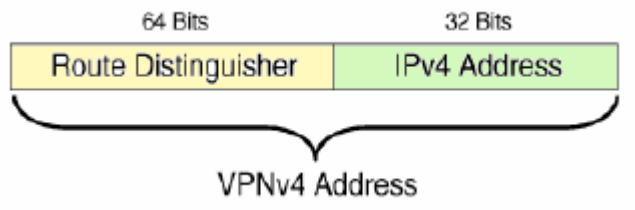
Tipo de conexión	Nombre
P – ISP	isp
Conexiones P – P	core
Conexiones PE - P	core
Conexiones PE – PE (enrutador – enrutador; enrutador – switch)	core
Enlace para demos	dem
Enlace de clientes VPN	vp3
Enlace de clientes internet	int

Fuente: Abreviatura propuesta

Elaborado por: Las autoras

RD (ROUTE DISTINGUISHER) es un campo de 8 bytes, el que permite distinguir la ruta dentro del backbone IP y debe ser único en la red.

FIGURA 4.3. DIAGRAMA DEPAQUETE VPNv4 ADDRESS



Fuente: www.cisco.com

Elaborado por: Las autoras

Existen varias formas de fijar el RD, 0x00 (2 octetos campo tipo, 2 octetos para el sistema autónomo y 4 octetos para el número único asignado por la VPN), 0x01 (2 octetos campo tipo, la dirección IP del cliente y 2 octetos para el número único asignado por la VPN) y 0x02 (2 octetos campo tipo, 4 octetos para el sistema autónomo y 2 octetos para el número único asignado por la VPN); la que

usaremos es la 0x00, ya que se trata de una red para una empresa celular y el número de clientes siempre va en aumento.

FIGURA 4.4. NÚMERO DE SISTEMA AUTÓNOMO

[Tipo][Sistema autonomo]: [Número asignado]

El número del sistema autónomo que se utilizará para la red MP-BGP será el 19114 que corresponde a el sistema autónomo privado de TME

Fuente: www.cisco.com

Elaborado por: Las autoras

Para el número asignado se debe ir desde 01 ya que con éste inicia la VPN, se debe dejar un rango para usos futuros.

Usaremos la siguiente tabla como guía de configuración de VRF.

```
        Ip vrf  "vpn-name"
        rd  19114:0#####
route-target export  19114:#####
route-target import  19114:#####
```

4.2. ESTUDIO DE PRIORIDADES

Se debe tomar en cuenta que el uso de prioridades es necesario para facilitar la aplicación de servicios en tiempo real. Para esto se usa:

Un algoritmo de Round Robin con prioridades que lee en turno circular y siempre que sea posible, más paquetes de la cola con prioridad 10 y menos paquetes a medida que la prioridad de la cola decae. En condiciones donde las colas tengan tráfico suficiente, el algoritmo de selección de paquetes leerá 11 paquetes de la cola con prioridad 10, 10 de la cola con prioridad 9, 9 de la cola con prioridad 8, y así hasta leer sólo uno de la cola con prioridad 0. Tras esto, el ciclo se repite. De este modo se está priorizando claramente el tráfico de las colas con prioridad más alta que el de las colas con menor prioridad; o sea, se procesa más cantidad de

paquetes más prioritarios que de los menos prioritarios.

En cada ciclo completo de Round Robin con prioridades, con todos los tráficos existentes y con el número suficiente de cada uno de ellos, en la peor situación, el Round Robin por prioridades atenderá paquetes de la siguiente forma:

FIGURA 4.5. DIAGRAMA DE PAQUETES POR TIPO DE TRÁFICO

Tipo de tráfico	Número de paquetes
TLDP	11
GPRSP	10
RLPRP	9
GoS 3 + LSP	8
GoS 3	7
GoS 2 + LSP	6
GoS 2	5
GoS 1 + LSP	4
GoS 1	3
GoS 0 + LSP	2
Sin GoS	1
Paquetes totales	66 paquetes

Fuente: "Soporte de Garantía de Servicio (GoS) a flujos privilegiados en MPLS", Manuel Domínguez

Por lo que, independientemente del número de paquetes que haya en el búfer el nodo o del tipo de tráfico de que se trate, al menos un paquete de cada tipo se conmutará cada 66 paquetes; en general no será normal que haya todo tipo de tráfico y con cantidad suficiente de cada uno de ellos por lo que ésta cota la mayor parte del tiempo será menor.

Las prioridades definidas en los puertos son como se muestran a continuación, dependiendo del tipo de tráfico existente en el búfer.

Prioridad 10 Paquete TLDP
Prioridad 9 Paquete GPSRP
Prioridad 8 Paquete RLPRP
Prioridad 7 Paquete con nivel GoS 3 y LDP de respaldo activado
Prioridad 6 Paquete con nivel GoS 3 y LDP de respaldo desactivado
Prioridad 5 Paquete con nivel GoS 2 y LDP de respaldo activado
Prioridad 4 Paquete con nivel GoS 2 y LDP de respaldo desactivado
Prioridad 3 Paquete con nivel GoS 1 y LDP de respaldo activado
Prioridad 2 Paquete con nivel GoS 1 y LDP de respaldo desactivado
Prioridad 1 Paquete sin GoS pero LDP de respaldo activado
Prioridad 0 Paquete tradicional

4.3. CONFIGURACIONES DE EQUIPOS, PROTOCOLO DE ENRUTAMIENTO IGP

CiscoConfig reconoce los siguientes comandos:

• InitialClass

• InitialPolicy

• LSPRoute

• PolicyModification

• Filter

• LSPPolicy

• LSPStatusChange

• LSPModification

• LSPDeletion

Para todos ellos tiene un método privado que lleva a cabo las instrucciones correspondientes a cada comando. En general, se cumple la siguiente secuencia:

1. Configuración exigida por el comando mediante una instancia a CLIRouter (interfaz que permite el uso de la clase CLICisco). Es este objeto, el que llevará a cabo la escritura de la línea de comandos mediante el protocolo CLI. Para establecer la comunicación con este protocolo, se debe establecer una sesión telnet contra el *router.* De esta sesión, el NEM obtiene respuestas que paseará según sea conveniente, ya sea para confirmar que se está escribiendo sobre el router sin encontrar errores de operación, o para recuperar algún tipo de dato (ej. *show mpls traffic-eng topology path <LSPname>*).

2. Comprobación de los cambios realizados sobre el running configuration del router mediante protocolo SNMP (siempre que sea posible).

3. Si se confirman los cambios mediante el paso 2, se procede a grabarlos en la start-up configuration del router. Si no, se instancia un objeto Error, indicando el tipo de mensaje que no se ha podido ejecutar, y se deja en la deque de envíos del handlerResponse.

El resto de características se habilitan con el update para MPLS.

Ip: 10.0.0.1

Rtpe01_mtz

```
Interface loopback 10
 description rtpe01_mtz
 ip address
 no ip redirects
 no ip unreachables
 no ip directed-broadcast

interface serial ??
 description
 bandwidth 1984
 ip address 255.255.255.252
 encapsulation ppp


router ospf 1
 router-id
 network    0.0.0.31 area 0
 network    0.0.0.3 area 0
```

Ip: 10.0.0.2

Igual a la de Rtpe01_mtz

Ip: 10.0.0.3

Rtpe01_pst

```
Interface loopback 10
 description rtpe01_pst
 ip address
 no ip redirects
 no ip unreachables
 no ip directed-broadcast

interface GigabitEthernet 0/3
 description
 bandwidth 100000
 ip address
 duplex full
 speed 100
 media-type rj45
 no negotiation auto

router ospf 1
 router-id
 network    0.0.0.31 area 0
 network    0.0.0.3 area 0
```

Ip: 10.0.0.4

rtpe01_crm

```
Interface loopback 10
 description rtpe01_crm
 ip address
 no ip redirects
 no ip unreachables
 no ip directed-broadcast

interface GigabitEthernet 0/3
 description
 bandwidth 100000
 ip address  ??
 duplex full
 speed 100
 media-type rj45
 no negotiation auto

router ospf 1
 router-id
 network     0.0.0.31 area 0
 network     0.0.0.3 area 0
```

Ip: 10.0.0.5

rtpe01_pst

```
Interface loopback 10
 description rtpe01_pst
 ip address
 no ip redirects
 no ip unreachables
 no ip directed-broadcast

interface serial 1/0
 description
 bandwidth 1984
 ip address ??    255.255.255.252
 encapsulation ppp

interface serial 1/1
 description
 bandwidth 1984
 ip address ??    255.255.255.252
 encapsulation ppp

interface serial 1/2
 description
 bandwidth 1984
 ip address ??    255.255.255.252
 encapsulation ppp

router ospf 1
 router-id
 network     0.0.0.31 area 0
 network     0.0.0.3 area 0
 network     0.0.0.3 area 0
 network     0.0.0.3 area 0
```

Ip: 10.0.0.6

rtpe01_crt

```
Interface loopback 10
 description rtpe01_crt
 ip address
 no ip redirects
 no ip unreachables
 no ip directed-broadcast

interface serial 3/0
 description
 bandwidth 1984
 ip address ??   255.255.255.252
 encapsulation ppp

interface serial 3/1
 description
 bandwidth 1984
 ip address ??   255.255.255.252
 encapsulation ppp

interface serial 4/0
 description
 bandwidth 1984
 ip address ??   255.255.255.252
 encapsulation ppp

router ospf 1
 router-id
 network     0.0.0.31 area 0
 network     0.0.0.3 area 0
 network     0.0.0.3 area 0
 network     0.0.0.3 area 0
```

Ip: 10.0.0.7

rtpe01_pst

```
Interface loopback 10
 description rtpe01_pst
 ip address
 no ip redirects
 no ip unreachables
 no ip directed-broadcast

interface serial 1/3
 description
 bandwidth 1984
 ip address ??   255.255.255.252
 encapsulation ppp

interface serial 5/3
 description
 bandwidth 1984
 ip address ??   255.255.255.252
 encapsulation ppp


router ospf 1
 router-id
 network     0.0.0.31 area 0
 network     0.0.0.3 area 0
 network     0.0.0.3 area 0
```

Ip: 10.0.0.8

```
rtpe02_mtz      Interface loopback 10
                 description rtpe01_mtz
                 ip address
                 no ip redirects
                 no ip unreachables
                 no ip directed-broadcast

                interface serial 2/4
                 description
                 bandwidth 1984
                 ip address ??   255.255.255.252
                 encapsulation ppp

                interface serial 2/5
                 description
                 bandwidth 1984
                 ip address ??   255.255.255.252
                 encapsulation ppp


                router ospf 1
                 router-id
                 network    0.0.0.31 area 0
                 network    0.0.0.3 area 0
                 network    0.0.0.3 area 0
```

4.4. CONFIGURACIÓN DE BGP

La red de TME generalmente usada, usa el sistema autónomo 19114, la forma como están hechas las vecindades entre los equipos no es óptima para la operación del backbone MPLS.

Se requiere hacer los siguientes cambios en la configuración default de BGP para corregir la configuración de BGP:

- Para la modificación de BGP se requiere que esté configurado el IGP entre los equipos que formarán el backbone de MPLS.

- Las vecindades se establecerán utilizando una dirección de loopback, en éste caso se utilizará la nueva dirección configurada en la interfaz loopback 10.

- Se utilizarán dos enrutadores que harán la función de route-reflector en la red para reducir el número de vecindades en la red. Estos serán los enrutadores rtpe01crt y rtpe01crm por ser los equipos que están conectados a cada ISP en Quito y Guayaquil.

- Pasar la conexión del enrutador gw-sl-uio del segmento del core 200.24.208.0 a la interfaz del enrutador rtpe01_mtz G0/3. Establecer una confederación entre el enrutador gw-sl-uio – y el rtpe01_mtz. La configuración de BGP se manejará en el siguiente orden:

Vecindad enrutador rtpe01_pst – a route reflectors.

- Borrar configuración de vecindades enrutador rtpe01_pst – rtpe01_crm. Se borrarán cinco vecindades.
- Configuración Vecindad enrutador rtpe01_pst – rtpe01_crm
- Configuración Vecindad enrutador rtpe01_pst – rtpe02_mtz
- Ajustes tabla de BGP ruta default

Vecindad enrutador rtpe01_mtz a route reflectors.

- Pendiente definición segmento core Matriz
- Borrar vecindad enrutador rtpe01_mtz - rtpe02_mtz
- Borrar vecindad enrutador rtpe01_mtz - rtpe02_pst
- Configuración Vecindad enrutador rtpe01_mtz – rtpe01_crm
- Configuración Vecindad enrutador rtpe01_pst – rtpe02_mtz
- Ajustes tabla de BGP ruta default

Vecindad enrutador rtpe01_crm a route reflectors

- Borrar vecindad enrutador rtpe01_crm - rtpe02_pst
- Borrar vecindad enrutador rtpe01_crm - rtpe02_mtz
- Configuración Vecindad enrutador rtpe01_crm – rtpe01_crm

- Configuración Vecindad enrutador rtpe01_crm – rtpe02_mtz
- Ajustes tabla de BGP ruta default

Vecindad enrutador rtpe01_crt a route reflectors

- Configuración Vecindad enrutador rtpe01_crm – rtpe01_crm
- Configuración Vecindad enrutador rtpe01_crm – rtpe02_mtz
- Ajustes tabla de BGP ruta default

4.5. CONTROL DE CONGESTIÓN EN LOS ACCESOS AL BACKBONE

Para las tecnologías de acceso multiacceso o que utilizan interfaces ethernet como lo son: módems de fibra óptica; radios multiacceso; convertidores E1 a ethernet, gracias a que el CPE del cliente puede transmitir a la velocidad del reloj de la interfaz de acceso, que en la mayoría de casos es mayor al rate contratado se requiere configurar una política que regule el tráfico de salida tanto en la subinterfaz del PE como en la interfase del CPE.

Configuración básica para controlar el tráfico convenido en los enlaces multiacceso:
policy-map rate_”valor rate convenido”
class class-default
shape average [valor Rate contratado]
interface GigabitEthernet0/0.2.#
encapsulation dot1Q “# Vlan#
description vpn:cliente:abc:swpe01mtz:G0/0.#
ip vrf “nombre vrf “
ip address [red] [mascara]
no ip redirect
service-policy output rate_”valor rate convenido”
duplex full
speed [valor]
media-type rj45

CAPÍTULO V

“QoS Y ESTABLECIMIENTO DE CLASES DE SERVICIO”

CAPÍTULO V

5.1. QoS EN LA RED

5.1.1. CALIDAD DE SERVICIO (QoS)

Es un mecanismo para controlar la fiabilidad y la facilidad de uso de la red de telecomunicaciones. Las compañías celulares y en general de comunicaciones deben ofrecer a sus clientes un servicio de calidad.

Una parte del QoS es el grado de servicio o GOS. QoS es parte de la categoría de servicio o SMO.

5.1.2. FACTORES QUE AFECTAN QoS

Hay muchos factores que afectan la calidad de servicio de una red móvil. Es correcto buscar principalmente QoS del punto de vista del cliente, es decir, como QoS es juzgada por el usuario. Hay una métrica estándar de QoS para el usuario que puede ser medida con la tasa de QoS.

Estos parámetros son: la cobertura, la accesibilidad (incluye SMO), y la calidad de audio.

En la cobertura de la fuerza de la señal se mide a través de equipos de prueba y esto puede utilizarse para estimar el tamaño de la celda.

Accesibilidad es acerca de la determinación de la capacidad de la red para manejar con éxito las llamadas de móvil a fijo y de las redes móviles a las redes móviles.

La calidad de audio de vigilancia considera una exitosa convocatoria de un período de tiempo por la claridad del canal de comunicación.

Todos estos indicadores son utilizados por la industria de las telecomunicaciones para calificar la calidad de los servicios de una red.

5.1.3. MEDICIÓN DE QoS

QoS en la industria se mide también desde la perspectiva de un experto (por ejemplo, ingeniería de tráfico). Esto involucra la evaluación de la red para ver si ofrece la calidad que el planificador de la red ha colocado como meta.

Ciertos instrumentos y métodos (analizadores de protocolo, conducir pruebas y mediciones de Operación y Mantenimiento), se utilizan para la medición de este QoS:

Analizadores de protocolo están conectados a BTSs, BSCs, y MSCs por un período de tiempo para comprobar si hay problemas en la red celular. Cuando un problema se descubre el personal puede registrar y analizar.

Pruebas de Drivers permitirá a la red móvil ser probado a través de la utilización de un equipo de personas que toman el papel de los usuarios y de adoptar las medidas de QoS examinado anteriormente a la tasa de QoS de la red. Esta prueba no es aplicable a toda la red, por lo que siempre es una muestra estadística.

En los Centros de Operación y Mantenimiento (OMCs), contadores se utilizan en el sistema para los diversos acontecimientos que proporcionan el operador de la red con información sobre el estado y la calidad de la red.

Por último, las quejas de los clientes son una fuente vital de información sobre QoS, y no deben ser ignorados.

5.2. GoS CELULAR

En general, GoS (Grado de Servicio) se mide mirando el tráfico transportado, el tráfico ofrecido y el cálculo de la pérdida de tráfico y bloqueado. La proporción de llamadas perdidas es la medida del SMO. Por grupos de circuito celulares SMO es un aceptable 0,02. Esto significa que dos usuarios del grupo de circuito de un centenar se enfrentarán al rechazo de una llamada durante la hora pico.

El grado de nivel de servicio es aceptable, por consiguiente, el nivel de tráfico que la red puede perder, SMO, se calcula a partir de la fórmula Erlang-B, en función del número de canales necesarios para la intensidad de tráfico ofrecido.

5.2.1. CELULAR CALIDAD DE AUDIO

DSP[33]s La calidad de audio de una red celular depende, entre otros factores, el sistema de modulación (por ejemplo, FSK, QPSK) en uso, la adecuación a las características de la canal y el procesamiento de la señal recibida en el receptor utilizando DSPs.

5.2.2. ASIGNACIÓN DE GoS

Así, luego de varias reflexiones consideramos que sería suficiente con establecer cuatro grados o niveles de garantía de servicio distintos para este análisis.

Posiblemente el establecimiento de más niveles en condiciones de tráfico real de Internet, fuese muy difícil de mantener y cumplir. Por otro lado, de alguna forma se marcan los paquetes que deseen contar con algún método para cuando un enlace falle.

[33] Un **procesador digital de señal (DSP)** es un microprocesador especializados diseñados específicamente para el procesamiento de señales digitales, en general en tiempo real de la computación.

Los cuatro niveles de GoS se pueden codificar con dos bits y la existencia de LSP de respaldo y un tercero, por lo que la siguiente tabla se observa mejor cómo se utilizarían tres bits para representar todas las opciones.

FIGURA 5.1. NIVELES DE GoS

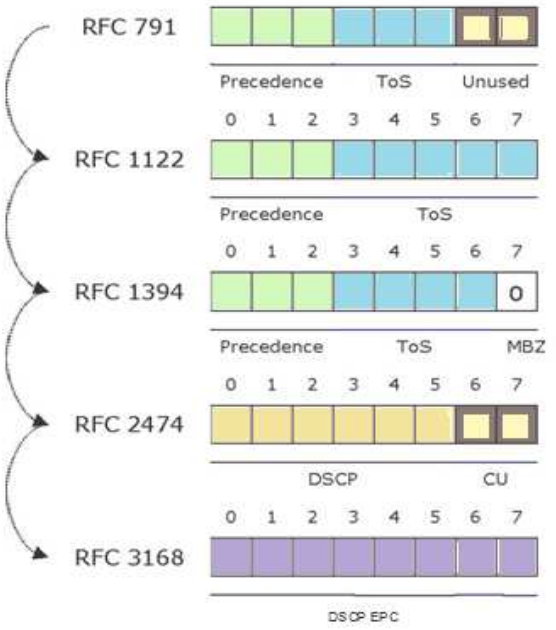
LSP	GoS_1	GoS_0	Significado
0	0	0	Paquete no marcado con GoS. A todos los efectos, es un paquete tradicional.
0	0	1	Paquete marcado con nivel de GoS 1 y sin solicitud de LSP de respaldo.
0	1	0	Paquete marcado con nivel de GoS 2 y sin solicitud de LSP de respaldo.
0	1	1	Paquete marcado con nivel de GoS 3 y sin solicitud de LSP de respaldo.
1	0	0	Paquete no marcado con GoS pero con solicitud de LSP
1	0	1	Paquete marcado con nivel de GoS 1 y con solicitud de LSP de respaldo.
1	1	0	Paquete marcado con nivel de GoS 2 y con solicitud de LSP de respaldo.
1	1	1	Paquete marcado con nivel de GoS 3 y con solicitud de LSP de respaldo.

Fuente: "Soporte de Garantía de Servicio (GoS) a flujos privilegiados en MPLS", Manuel Domínguez

El campo EXP, de 3 bits, no se usa en esta etiqueta especial; por lo que se lo puede usar para almacenar a nivel MPLS el grado de Garantía de Servicio requerido por un paquete y la necesidad o no de proteger el LSP por el que éste vaya a circular.

En la cabecera IP no hay tres bits libres al menos de una forma obvia, por lo que el primer lugar donde se introducen esos tres bits es en el campo ToS, que cuenta con 8 bits y cuyo uso se ha modificado en diversas ocasiones. Sin embargo, un estudio evolutivo de este campo en los distintos RFC's nos muestra lo siguiente:

FIGURA 5.2. DISTRIBUCIÓN DE BITS PARA ToS



Fuente: "Soporte de Garantía de Servicio (GoS) a flujos privilegiados en MPLS", Manuel Domínguez

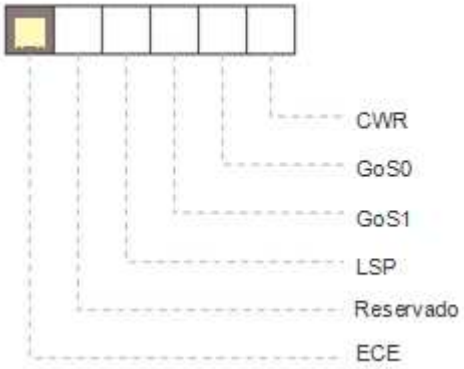
Por lo que debemos usar TCP, en su la cabecera, hay un campo de seis bits reservados desde la creación de TCP; durante mucho tiempo ese campo ha permanecido intacto, pero en los últimos años se ha comenzado a hacer uso de algunos de sus bits, concretamente de dos, para poder marcar alguna de las opciones de servicios diferenciados. En la siguiente figura se muestra la evolución de dicho campo en el transcurso del tiempo según los RFCs.

FIGURA 5.3. EVOLUCIÓN DE CAMPO TCP

Fuente: "Soporte de Garantía de Servicio (GoS) a flujos privilegiados en MPLS", Manuel Domínguez

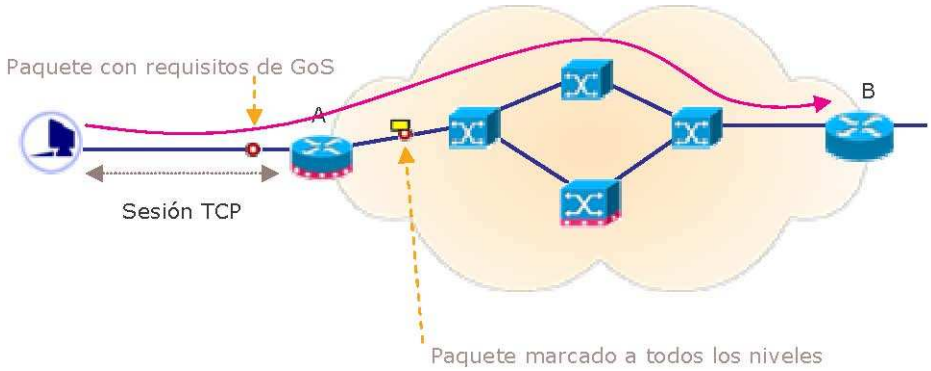
Se pueden usar así:

FIGURA 5.4. EVOLUCIÓN DE CAMPOS



Fuente: "Soporte de Garantía de Servicio (GoS) a flujos privilegiados en MPLS", Manuel Domínguez

FIGURA 5.5. CAMINO DE UN PAQUETE CON GOS



Fuente: "Soporte de Garantía de Servicio (GoS) a flujos privilegiados en MPLS", Manuel Domínguez

5.2.3. IDENTIFICACIÓN GLOBAL DE LOS PAQUETES

Así, ya de forma definitiva, todo el campo de opciones de IPv4 quedaría formateado para soportar GoS, tomando encuenta que es proceso de simulación, quedando como se muestra en la siguiente figura:

Figura 5.6. GoS EN IPv4

Fuente: "Soporte de Garantía de Servicio (GoS) a flujos privilegiados en MPLS", Manuel Domínguez

Los tres bits de GoS y de LSP de respaldo, que requiere la simulación se ordenan de la siguiente forma dentro del campo GoS+LSP de 8 bits:

FIGURA 5.7. Campo GoS+LSP

Fuente: "Soporte de Garantía de Servicio (GoS) a flujos privilegiados en MPLS", Manuel Domínguez

Así que como se puede observar, aún tenemos libres los 5 primeros bits del primer octeto del campo opciones.

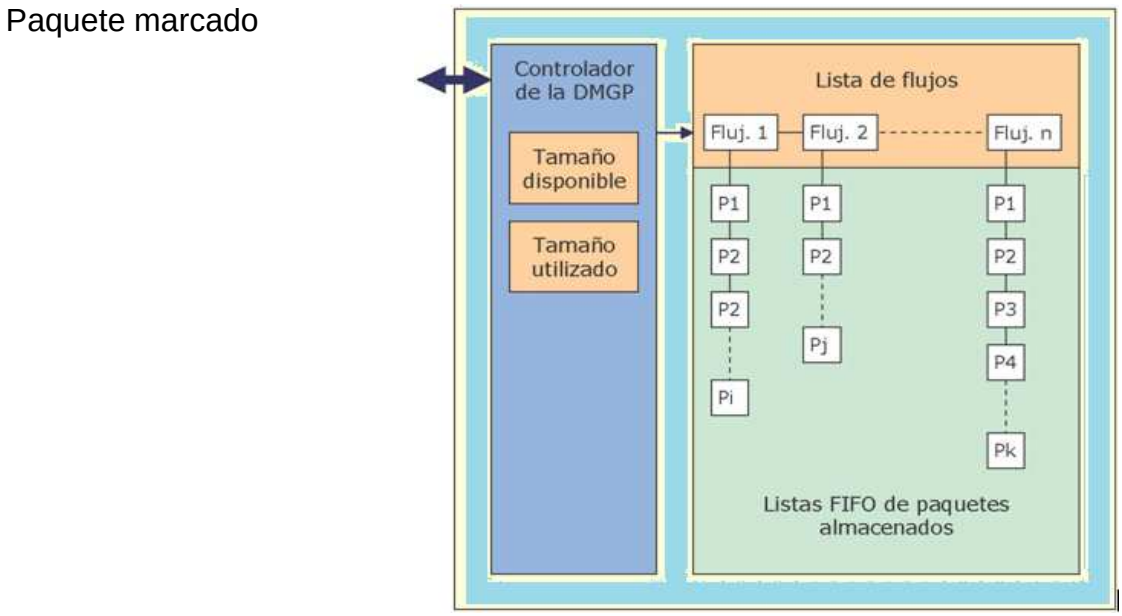
5.2.4. MEMORIAS TEMPORALES (DMGP)

Es importante que en este punto se ha conseguido marcar los requerimientos de GoS y de LSP de respaldo en los paquetes que sea necesario a todos los niveles. Ya que se puede identificar a cada paquete globalmente dentro del dominio MPLS y asi se puede solicitar su retransmisión en caso de que haya un descarte. Sin embargo para ello se almacena dentro de los nodos activos los paquetes

marcados con GoS que lo vayan atravezando; así se podrá buscar un paquete cuando el nodo activo reciba de otro nodo activo una solicitud de retransmisión.

La respuesta es el uso de memoria DMGP (Dynamic Memory for GoS PDU) o Memoria Dinámica para PDU‘s con GoS, que tiene la siguiente estructura interna.

FIGURA 5.8. ESTRUCTURA INTERNA DE MEMORIA DMGP



Fuente: “Soporte de Garantía de Servicio (GoS) a flujos privilegiados en MPLS”, Manuel Domínguez

“En ella, los paquetes marcados con GoS y almacenados. Un paquete se envía a la DMGP. Su controlador comprueba si existe una entrada de flujo para ese paquete; si no es así, comprueba si tiene reservas de memoria para satisfacer la demanda requerida por el paquete en base a su grado de GoS. Si tiene, le

reserva un tamaño fijo a ese flujo, crea la entrada e inserta el paquete en DMGP, asociándolo a dicha entrada de flujo que se acaba de crear"[34].

"Los siguientes paquetes del mismo flujo no necesitarán que se les cree una entrada de flujo. Se insertarán en la memoria DMGP si queda espacio necesario para él del total que se reservó a ese flujo. Si no queda, se comienzan a eliminar paquetes de la cola de paquetes de ese flujo, en orden FIFO, hasta que quede espacio suficiente para que pueda ser insertado el nuevo paquete"[35].

Esto nos lleva a que mientras más pase el tiempo y si no se ha recibido petición de retransmisión, por no poseer una memoria infinita (ningún dispositivo la tiene), los paquetes ceden lugar a los más nuevos, se puede reservar espacio por flujo de datos.

Si un paquete nuevo llega a la DMGP, pertenece a un flujo que no tiene entrada en la DMGP y se comprueba que la DMGP ha reservado ya todo el espacio del que disponía, ese flujo no reserva memoria; el nodo actual no podrá retransmitir paquetes de ese flujo, por lo que en caso de perdida no se podra retransmitir directamente desde ese nodo. Este caso se puede dar en hora pico o en un momento atípico.

Aún así, este nodo todavía podrá ofrecer el resto de servicios a ese tráfico que tiene requisitos de GoS, por ejemplo solicitar su retransmisión a otros nodos.

El tamaño reservado para cada flujo es constante, pero no existe razón para ser el mismo para cada DMGP de cada nodo. El tamaño se asigna por porcentajes del total de la DMGP y siempre ligado al nivel de GoS incorporado en el paquete, como se muestra a continuación.

---

[34] **Explicación tomada de** Manuel Domínguez Dorado creador del "OpenSimMPLS" http://gitaca.unex.es/opensimmpls

[35] **Explicación tomada de** Manuel Domínguez Dorado creador del "OpenSimMPLS" http://gitaca.unex.es/opensimmpls

FIGURA 5.9. PORCENTAJES ASIGNADOS EN LA DMGP

		Algunos ejemplos		
Grado de GoS	% DMGP asignado	DMGP = 1 KB	DMGP = 100 KB.	DMGP = 1000 KB.
1	4	41 Bytes por flujo.	4,1 KB. Por flujo	41 KB. Por flujo
2	8	82 Bytes por flujo.	8,2 KB. Por flujo	82 KB. Por flujo
3	12	123 Bytes por flujo.	12.3 KB. Por flujo	123 KB. Por flujo

Fuente: "Soporte de Garantía de Servicio (GoS) a flujos privilegiados en MPLS", Manuel Domínguez

En cualquier caso, se parte de que el número de flujos con requerimientos de GoS en el dominio MPLS debería ser poco usuales pese al crecimiento de nuevos servicios de valor agregado, en comparación con el número de flujos sin estos requerimientos; Además, para que hubiese problemas los flujos tendrían que coincidir en el tiempo porque cuando un flujo termina, la DMGP elimina su entrada y vuelve a quedar espacio sin reservar, aprovechable para otro flujo, no tiene porque guardardar historicos si la transmisión se completo correctamente.

El tamaño de la memoria DMGP depende en gran medida del tráfico que circule por la red, de los requerimientos de los usuarios finales, del tipo de aplicación, tipo de servicio, etc.

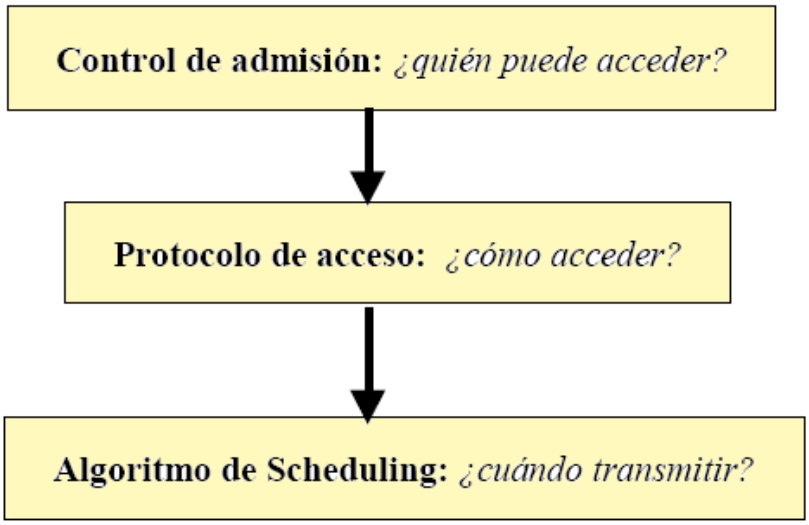
En última instancia debería ser el ingeniero de red encargado del mantenimiento de la troncal MPLS el que hiciese un estudio para modelar de forma adecuada el tráfico existente en la red y, conforme a ello, establecer el tamaño apropiado de

las memorias DMGP, igual que se puede modificar las unidades de asignación (clusters) en los sistemas de ficheros de los sistemas operativos para mejorar las prestaciones en base al tamaño medio de los ficheros del sistema, el uso del host, entre otros.

5.3. MECANISMOS PARA MANTENER QOS

Para mantener QoS se especifica la base:

FIGURA 5.10. MECANISMOS QoS



Fuente: www.cisco.com

Elaborado por: Las autoras

5.3.1. MECANISMOS DE SCHEDULING

Algoritmo de scheduling: Sirve para gestionar recursos una vez que se ha obtenido el acceso con un protocolo especificado, permitiendo efectuar una transmisión adecuada en función de prioridades.

Con esto se consigue que una transmisión no se efectúe solamente por los accesos o permisos que hayan sido adquiridos, sino que cada Terminal confirme los permisos adquiridos.

Para ello debe aplicar el algoritmo de scheduling, que determina quién de entre el conjunto de usuarios del sistema debe o no transmitir y cuándo debe hacerlo; la cantidad de información que es directamente proporcional a la ganancia.

Con ello se regular la interferencia presente en el sistema como el CDMA y así poder mantener la tasa de error bajo los límites establecidos para cada servicio.

Se puede dividir este proceso en dos:

- Priorización: De acuerdo a la prioridad se ordenan todas las peticiones de acuerdo al timeout.

- **Aceptación de peticiones**: De acuerdo a los criterios de Eb/No mínima se ordenan las peticiones, calculando la contribución con la ganancia de procesado.

Si la interferencia es inferior a la máxima permitida por cada una de las peticiones ya aceptadas, se aceptará la petición considerada, de lo contrario, se rechazará.

5.3.2. MECANISMOS DE ENCOLAMIENTO

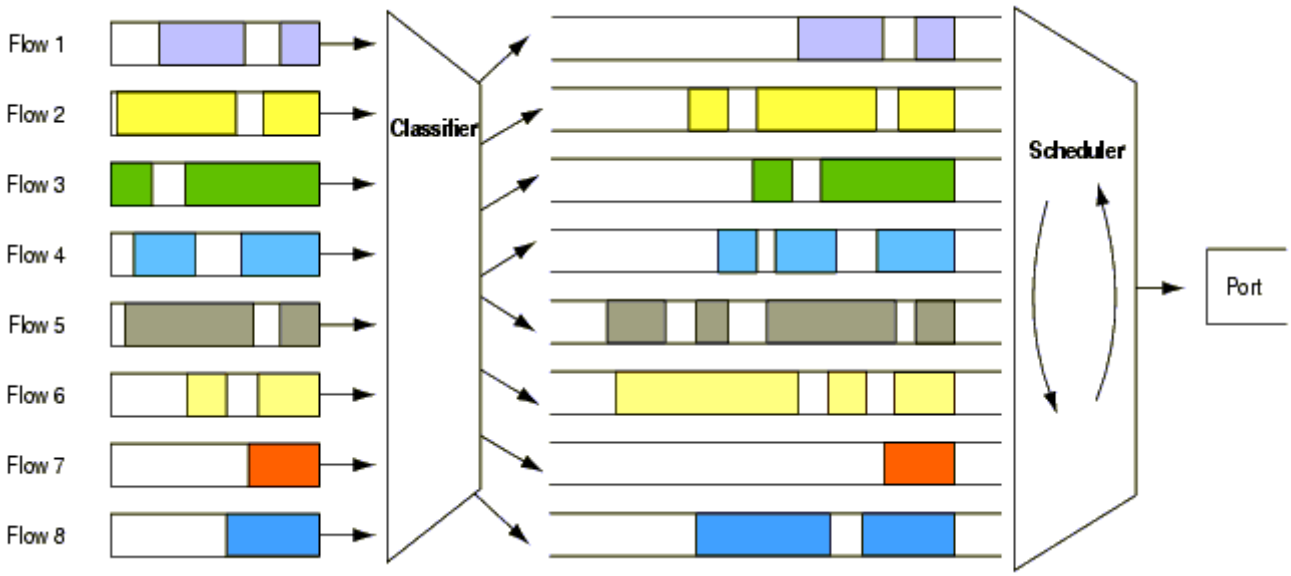
First In First Out (FIFO) es el método clásico empleado en los routers, debido a su simplicidad de implementación. En este algoritmo, el router cuenta con una sola cola en la que encola los paquetes que debe retransmitir en el mismo orden en el que son recibidos. Si recibe paquetes mas lentamente de lo que le toma retransmitirlos, la cola se mantiene desocupada. Si los recibe más rápido, los encola en el orden de llegada mientras transmite los que se encuentran antes en la cola.

Si la cola se llena y recibe paquetes adicionales, los descarta hasta que sirva el primer paquete de la cola y cuente, por tanto, con espacio para almacenar un nuevo paquete entrante.

FIFO no cuenta con ningún mecanismo orientado a asignar de manera justa la capacidad del canal en el enlace de salida. De esta forma, los flujos que envíen paquetes a una mayor velocidad van a ocupar una porción mayor de la cola del router, y por consiguiente, obtendrían una mayor cantidad de la capacidad del enlace, dado que sus paquetes serían los que el router más retransmitiría.

Fair Queuing se utiliza en enrutadores, conmutadores o multiplexores que reenviar paquetes de un buffer. El buffer funciona como un sistema de espera, donde los paquetes de datos se almacenan temporalmente hasta que sean enviados. El espacio de amortiguación se divide en muchas colas, cada una de las cuales se utiliza para mantener los paquetes de un flujo, que se define, por ejemplo, la fuente o el destino de las direcciones IP.

FIGURA 5.11. FAIR QUEUING



Fuente: www.cisco.com

Elaborado por: Las autoras

Stochastic Fair Queuing Es menos preciso que la anterior, sino que también requiere menos tiempo que los cálculos están casi perfectamente justo. SFQ se llama "estocástico", ya que realmente no asignar una cola para cada período de sesiones, tiene un algoritmo que divide el tráfico en un número limitado de colas usando un algoritmo de hashing.

RED (Random Early Detection) es un algoritmo que se utiliza para evitar la congestión. Su trabajo es evitar la congestión en la red, asegurándose de que la cola no se llene. Para ello calcula constantemente la longitud media (el tamaño) de la cola y la compara con dos umbrales o límites, un umbral mínimo y otro máximo. Si el tamaño medio de la cola se encuentra por debajo del umbral mínimo, entonces no se bloqueará ningún paquete. Si el tamaño medio se encuentra por encima del umbral máximo, entonces todos los paquetes nuevos que lleguen serán bloqueados. Si el tamaño medio se encuentra entre los valores de los dos umbrales, entonces se bloquearán los paquetes de acuerdo con un cálculo de probabilidad obtenido a raíz del tamaño medio de la cola. En otras palabras, según se va aproximando el tamaño medio de la cola al umbral máximo, se va bloqueando un número cada vez mayor de paquetes. Cuando bloquea los paquetes, RED escoge de qué conexiones bloqueará los paquetes de una forma aleatoria. Las conexiones que usen mayores cantidades de ancho de banda serán las que tengan una probabilidad más alta de que se bloqueen sus paquetes.

RED sólo se debería usar cuando el protocolo de transporte fuera capaz de responder a los indicadores de congestión de la red. En la mayoría de casos esto significa que RED se debería usar para poner en cola el tráfico TCP, y no el tráfico UDP o ICMP.

Flow Random Early Drop (FRED). FRED es RED con las siguientes modificaciones:

FRED no descarta los paquetes al estilo de RED cuando llegan si el número de paquetes en la cola de este flujo es menor que un parámetro dado cmin. Esto

previene que los flujos frágiles sufran ante la presencia de flujos robustos o no adaptivos en tiempos de congestión.

El parámetro cmin es recalculado como el número medio de paquetes por flujo en la cola del router; esto previene un descarte injusto sistemático de paquetes cuando los flujos cuentan con más de cmin paquetes en la cola.

FRED impone una cota cmax al numero de paquetes que cada flujo puede tener en la cola. Si un flujo intenta encolar más de cmax paquetes, son descartados. Adicionalmente, al flujo se le sanciona descartando todos sus paquetes entrantes hasta que el numero de paquetes que tenga en la cola sea menor que el número medio de paquetes por flujo en la cola. Esto previene que los flujos no adaptivos tomen control de toda la cola y nieguen el servicio a los adaptivos.

El valor de savg es calculado no solo al ingreso de paquetes sino también cuando son transmitidos, evitando valores artificialmente altos de este parámetro y previniendo descartes de paquetes cuando no son necesarios.

CAPÍTULO VI

“PROPUESTA DE MONITOREO DE TRÁFICO CON LA HERRAMIENTA COMPUTACIONAL OpenSimMPLS”

CAPÍTULO VI

6.1. DESCRIPCIÓN DE OpenSimMPLS

Es un simulador para dar soporte de Garantía de Servicio (GoS) a flujos privilegiados de información para ratificarse como una propuesta sólida y válida, con compatibilidad de sistemas.

Fuente: http://gitaca.unex.es/opensimmpls , "OpenSimMPLS"

Simulador creado por: Manuel Domínguez Dorado

6.2. CARACTERÍSTICAS Y FUNCIONAMIENTO BÁSICO (DATOS TOMADOS DEL MANUAL DE USO DEL PROGRAMA[36])

6.2.1. EL ENTORNO DE TRABAJO

El entorno de trabajo del simulador está diseñado con un mínimo de opciones a elegir, para que sea sencillo. Posee un área de trabajo, el menú principal y las

[36] EL MANUAL DE USUARIO DEL PROGRAMA CONSTA COMO UN ANEXO

ventanas de los escenarios abiertos.

6.2.2. ÁREA DE TRABAJO

Es el rectángulo mayor de la ventana principal que aparece al arrancar el simulador. Inicialmente no contiene nada, pero posteriormente aparecerán en ella las ventanas de escenarios que haya abiertas.

FIGURA 6.1. VENTANA DE TRABAJO DEL OpenSimMPLS

Fuente: http://gitaca.unex.es/opensimmpls , "OpenSimMPLS"

Simulador creado por: Manuel Domínguez Dorado

6.2.3. MENÚ PRINCIPAL

El menú principal es el menú de la ventana del área de trabajo. En él se encuentran todas las opciones que tienen que ver con el funcionamiento general de la aplicación.

Las opciones del menú principal están agrupadas en tres categorías:

- Escenario: que contiene las acciones que tienen que ver con los escenarios, tales como abrir, cerrar, guardar, crear un escenario nuevo, etcétera.

- Vista: que contiene las acciones que tienen que ver con la forma en que se mostrarán los distintos escenarios abiertos en el área de trabajo, como por ejemplo, minimizar, cascada, mosaico, etcétera.

- Ayuda: donde están las acciones que permiten al usuario obtener información adicional; por ejemplo, contenidos de ayuda, contactar con los autores,

etcétera.

6.2.4. VENTANA DE ESCENARIOS

Las ventanas de escenarios son aquellas que se abren sobre el área de trabajo. Puede haber diversas y cada una de ellas contiene todo lo necesario para realizar la simulación completa de un escenario propuesto.

FIGURA 6.2. VENTANA DE ESCENARIOS

Fuente: http://gitaca.unex.es/opensimmpls , "OpenSimMPLS"

Simulador creado por: Manuel Domínguez Dorado

CREAR UN NUEVO ESCENARIO

Cuando se quiera crear un nuevo escenario de simulación, hay que acudir al menú principal, concretamente a la opción escenario.

FIGURA 6.3. CREACIÓN DE UN NUEVO ESCENARIO

Fuente: http://gitaca.unex.es/opensimmpls , "OpenSimMPLS"

Simulador creado por: Manuel Domínguez Dorado

Como se ve en la figura, se debe seleccionar la opción “Nuevo“, que permitirá crear un nuevo escenario. Junto a la opción “Nuevo“, aparece una combinación de teclas, un atajado de teclado “Ctrl+N“. Esto significa que sin necesidad de desplegar nada en el menú principal, y desde cualquier lugar de la ventana principal, se puede crear un escenario pulsando simultáneamente las teclas “Control“ y “N“ del teclado.

Una vez seleccionada la opción, de cualquiera de los modos comentados, se abrirá una nueva ventana de escenario en el área de trabajo que se añadirá a las que ya pudiesen existir.

6.2.5. MODO DE TRABAJO DE OPENSIMMPLS

Posee tres modos distintos con cada escenario:

- Modo diseño: donde se podrán hacer todas las labores de diseño de topologías y configuración de los elementos de la red que queremos simular.
- Modo simulación: donde se podrá realizar la simulación en tiempo real del funcionamiento de la red diseñada.
- Modo análisis: donde se podrán ver gráficas analíticas sobre lo que ocurre en la simulación.

Y para que resulte sencillo llevar a cabo las tareas comentadas, las ventanas de escenario se dividen precisamente en estas tres áreas, mediante pestañas de separación.

FIGURA 6.4. MODO DISEÑO

Fuente: http://gitaca.unex.es/opensimmpls , “OpenSimMPLS”

Simulador creado por: Manuel Domínguez Dorado

Además, la ventana de escenario incorpora una cuarta pestaña donde se deben

configurar aspectos generales de la simulación.

6.3. REQUISITOS PARA LA IMPLEMENTACIÓN DE OpenSimMPLS

OpenSimMPLS 1.0, en su versión Standard única y autónoma se distribuye como una aplicación JAR.

Su instalación por tanto no requiere de ningún paso significativo, y simplemente hay que invocar su ejecución mediante la Máquina Virtual Java, correctamente instalada en el PC según su sistema operativo.

Lo más complejo, por tanto, será copiar el fichero OpenSimMPLS.jar en la carpeta que se desee y la instalación habrá concluido.

FIGURA 6.5. EJEMPLO EN WINDOWS, COPIANDO OPEN SIMMPLS DESDE EL CD

```
md c:\openSimMPLS 1.0
copy d:\openSimMPLS c:\openSimMPLS 1.0\openSimMPLS
```

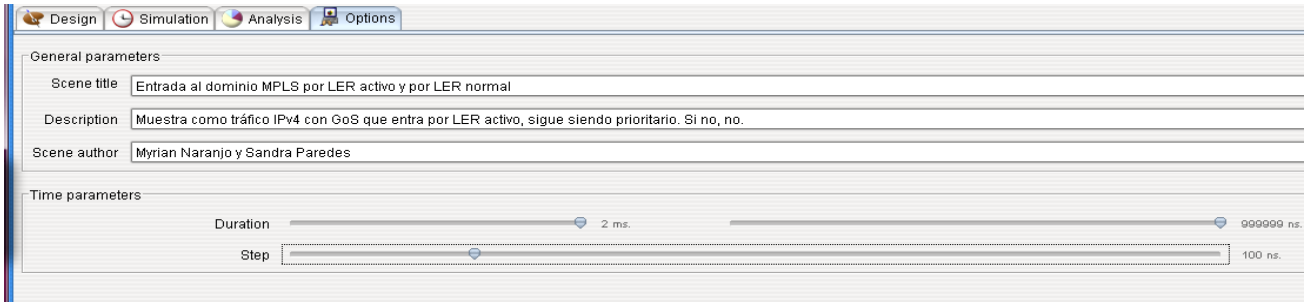
Fuente: http://gitaca.unex.es/opensimmpls , "OpenSimMPLS"

Simulador creado por: Manuel Domínguez Dorado

6.4. PROPUESTA DE MONITOREO DE TRÁFICO EN LÍNEA

Opciones:

FIGURA 6.6. OPCIONES DE MONITOREO CON OpenSimMPLS

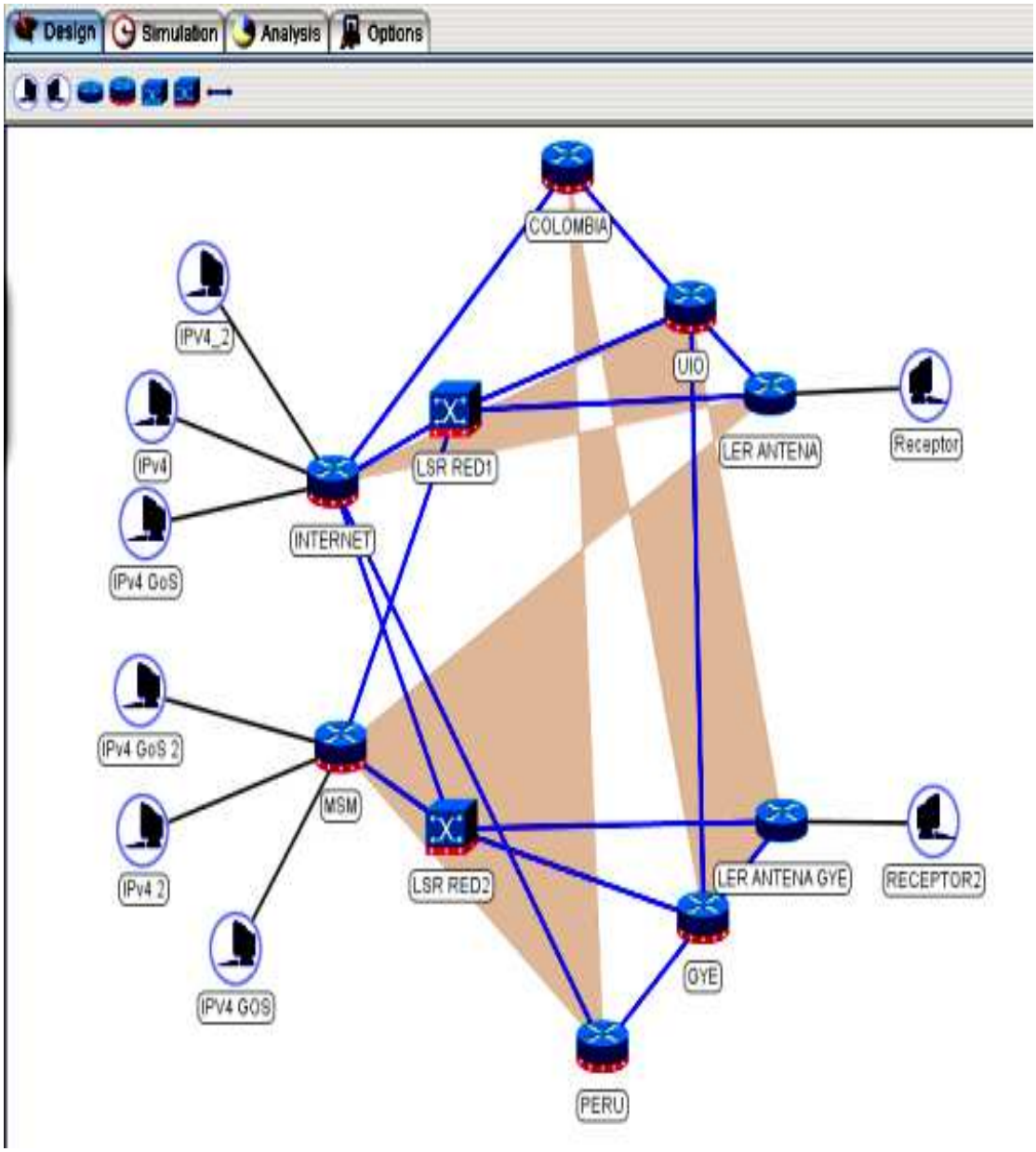


Fuente: http://gitaca.unex.es/opensimmpls , "OpenSimMPLS"

Simulador creado por: Manuel Domínguez Dorado

Red:

FIGURA 6.7. TOPOLOGÍA DE RED SIMULADA CON OpenSimMPLS

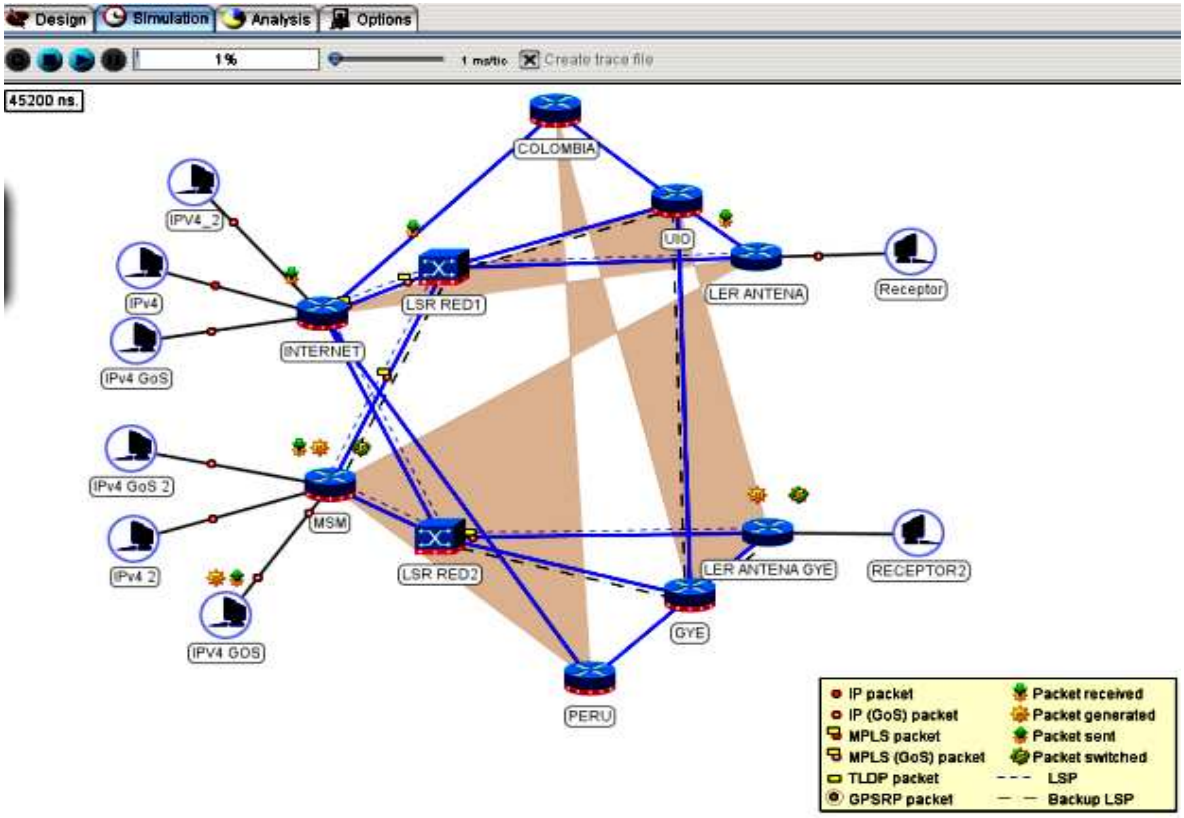


Fuente: http://gitaca.unex.es/opensimmpls , "OpenSimMPLS"

Simulador creado por: Manuel Domínguez Dorado

Red Elaborado por: Las autoras

FIGURA 6.8. SIMULACIÓN SIN CONGESTIÓN NI CAÍDA DE ENLACES



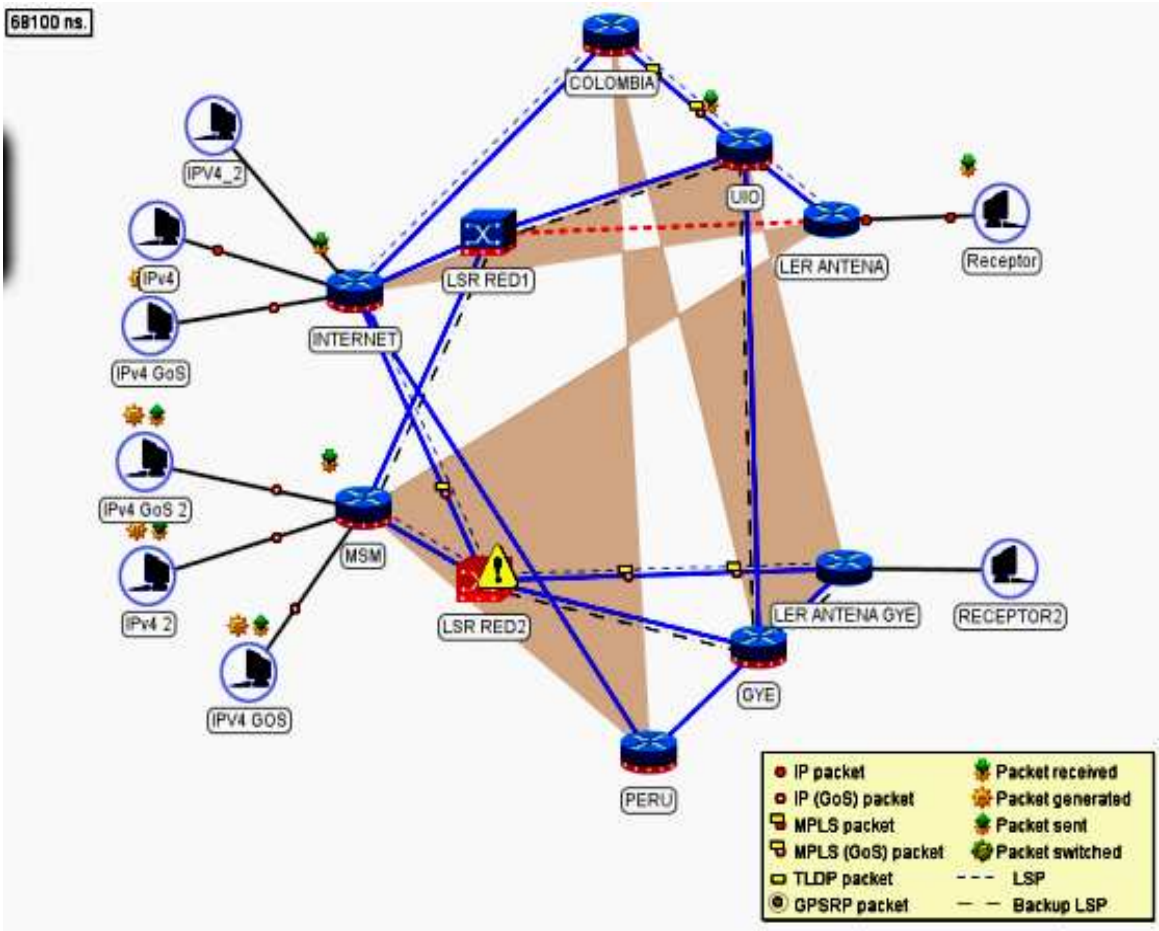
Fuente: http://gitaca.unex.es/opensimmpls , "OpenSimMPLS"

Simulador creado por: Manuel Domínguez Dorado

Red Elaborado por: Las autoras

En la siguiente figura se ve como una congestión en el router es balanceada con los otros y como la caída de un enlace es compensada con uso de enlaces redundantes automáticamente.

FIGURA 6.9. SIMULACIÓN CON ENLACES REDUNDANTES



Fuente: http://gitaca.unex.es/opensimmpls , "OpenSimMPLS"

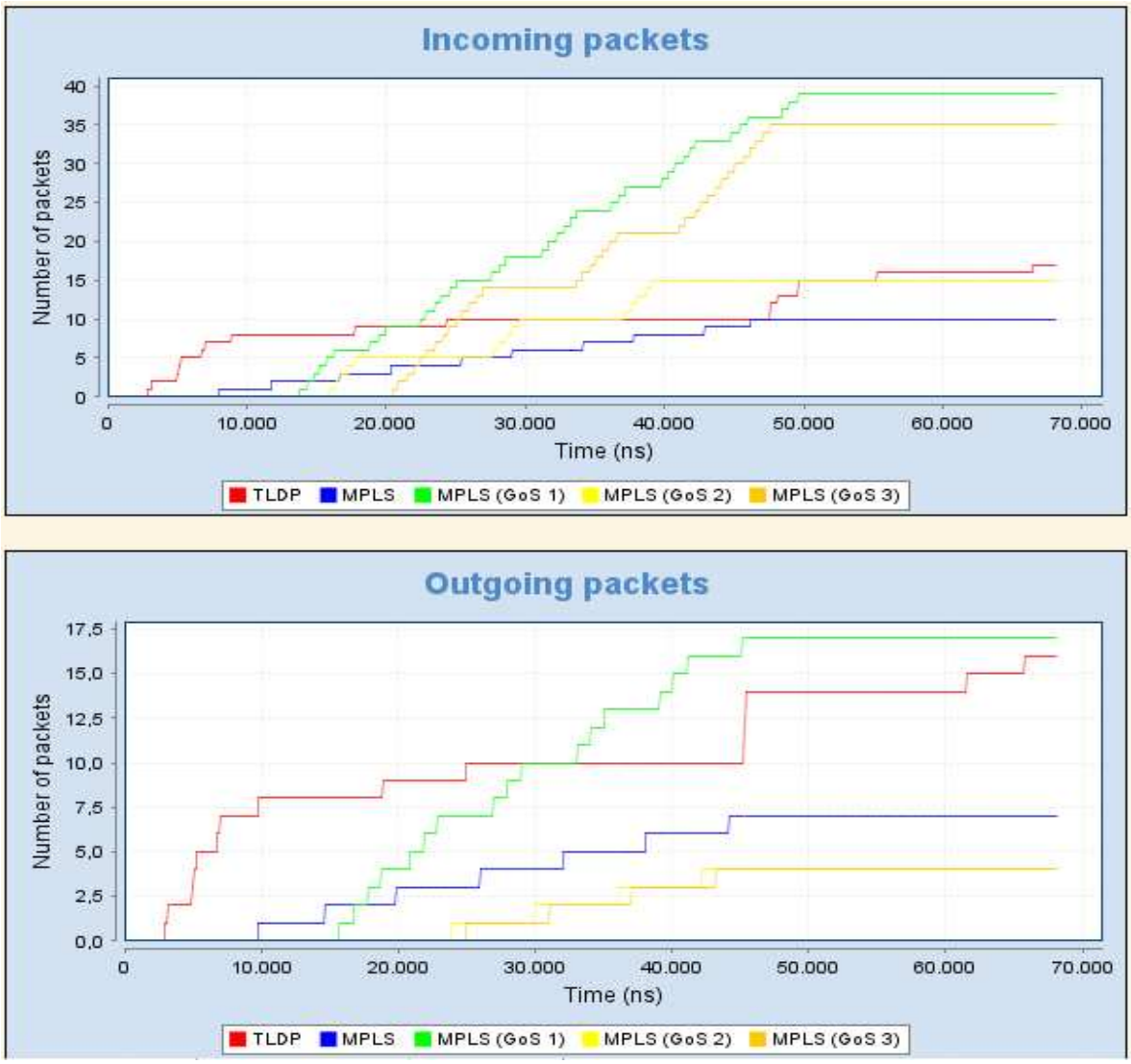
Simulador creado por: Manuel Domínguez Dorado

Red Elaborado por: Las autoras

Resultados:

A continuación se presentan los resultados obtenidos para cada nodo.

FIGURA 6.10. LSR RED 1 PAQUETES DE ENTRADA Y SALIDA

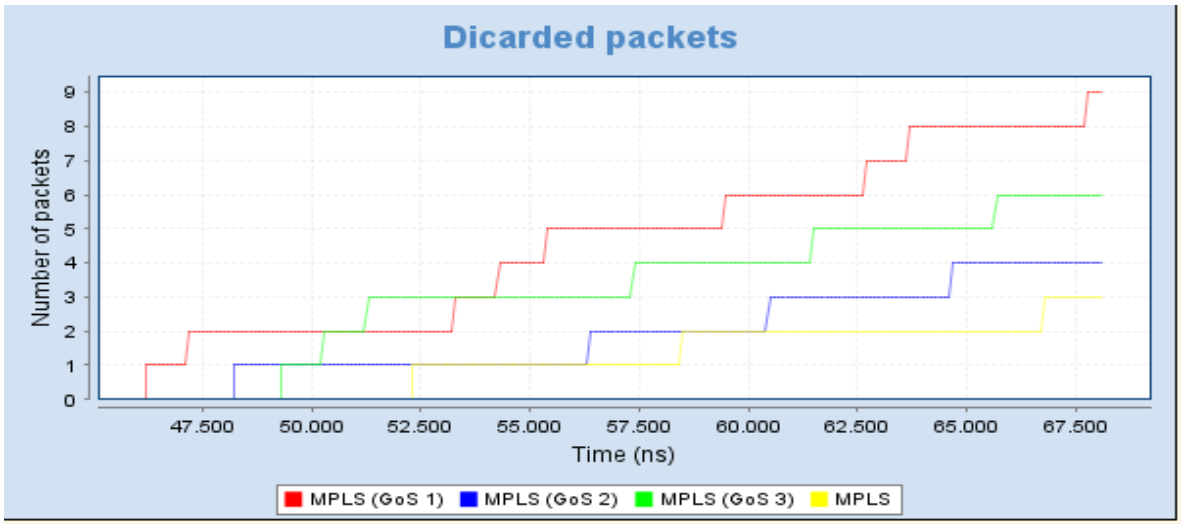


Fuente: http://gitaca.unex.es/opensimmpls , "OpenSimMPLS"

Simulador creado por: Manuel Domínguez Dorado

Red Elaborado por: Las autoras

FIGURA 6.11. LSR RED 1 PAQUETES DESCARTADOS

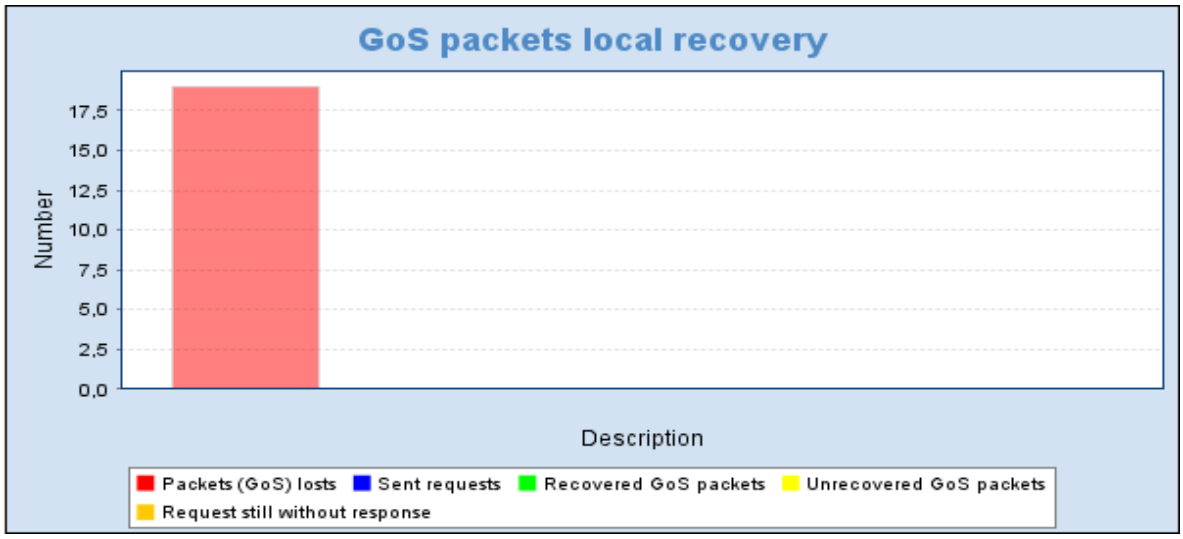


Fuente: http://gitaca.unex.es/opensimmpls , "OpenSimMPLS"

Simulador creado por: Manuel Domínguez Dorado

Red Elaborado por: Las autoras

FIGURA 6.12. LSR RED 1 GoS (GRADO DE SERVICIO)

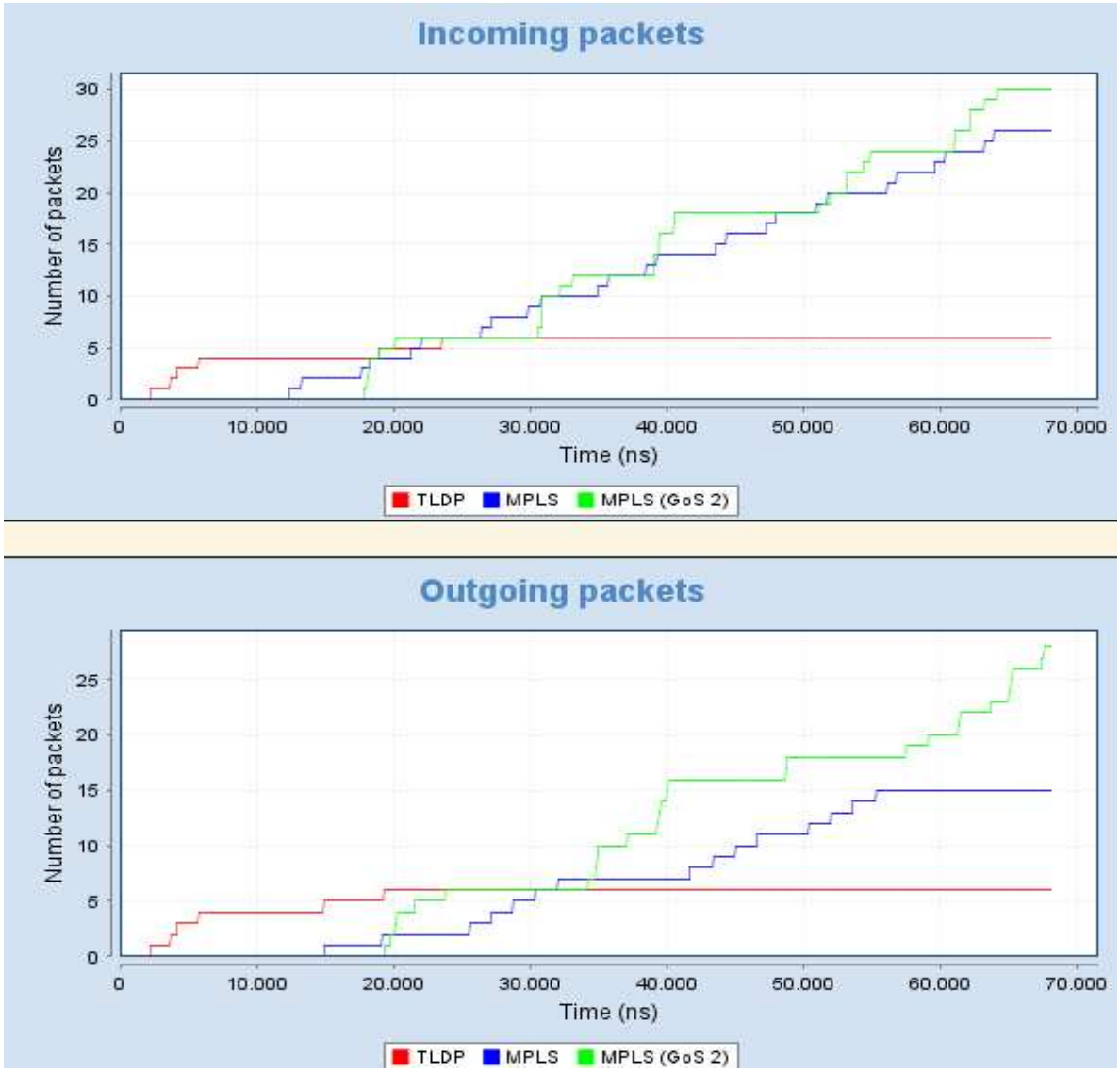


Fuente: http://gitaca.unex.es/opensimmpls , "OpenSimMPLS"

Simulador creado por: Manuel Domínguez Dorado

Red Elaborado por: Las autoras

FIGURA 6.13. LSR RED 2 PAQUETES DE ENTRADA Y SALIDA

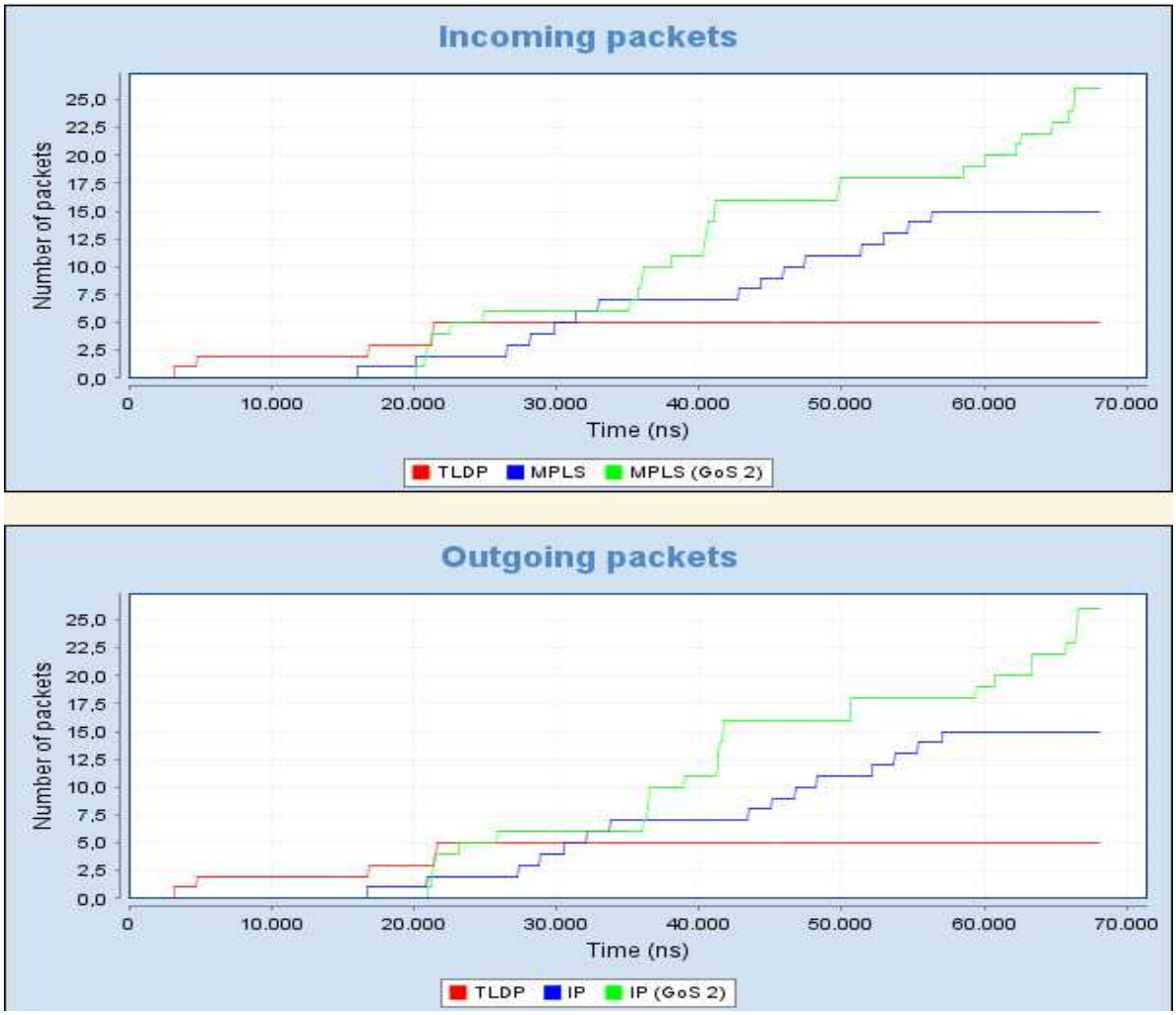


Fuente: http://gitaca.unex.es/opensimmpls , "OpenSimMPLS"

Simulador creado por: Manuel Domínguez Dorado

Red Elaborado por: Las autoras

FIGURA 6.14. LER ANTENA GYE

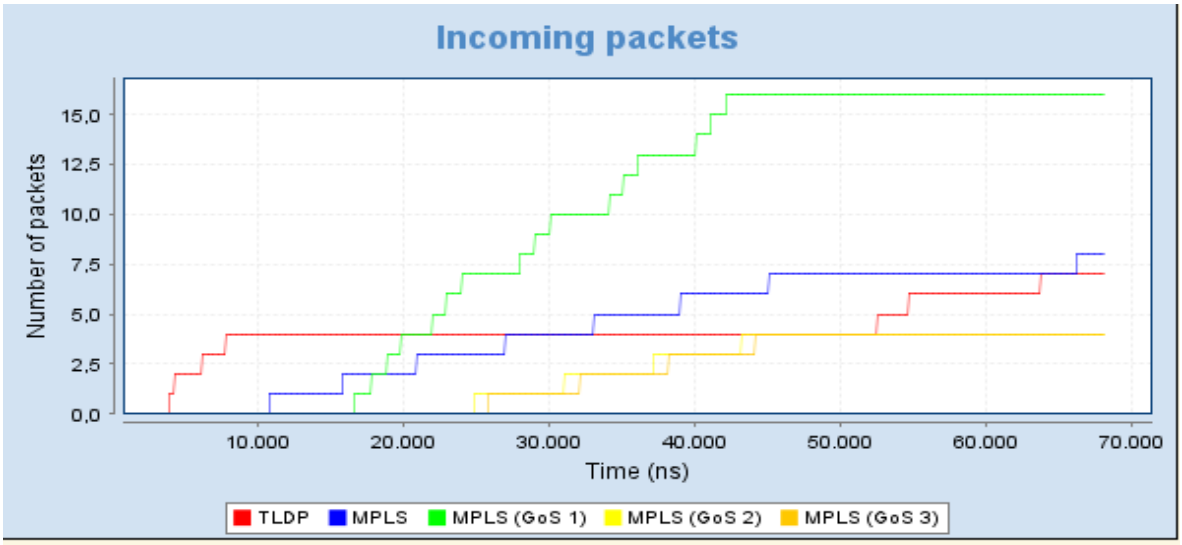


Fuente: http://gitaca.unex.es/opensimmpls , "OpenSimMPLS"

Simulador creado por: Manuel Domínguez Dorado

Red Elaborado por: Las autoras

FIGURA 6.15. LER ANTENA. PAQUETES DE ENTRADA

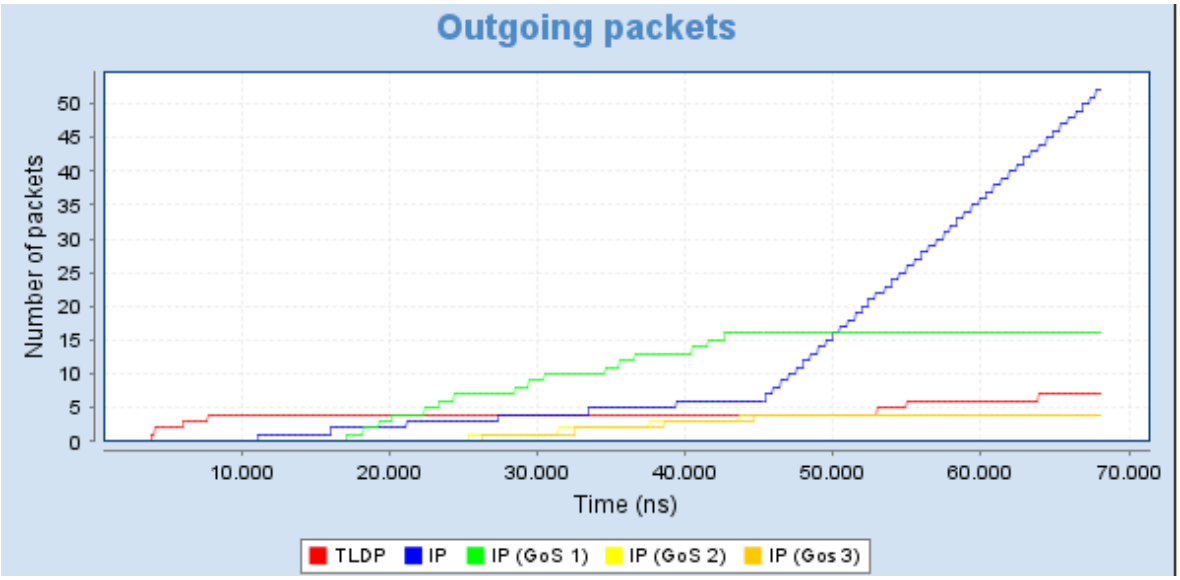


Fuente: http://gitaca.unex.es/opensimmpls , "OpenSimMPLS"

Simulador creado por: Manuel Domínguez Dorado

Red Elaborado por: Las autoras

FIGURA 6.16. LER ANTENA. PAQUETES DE SALIDA



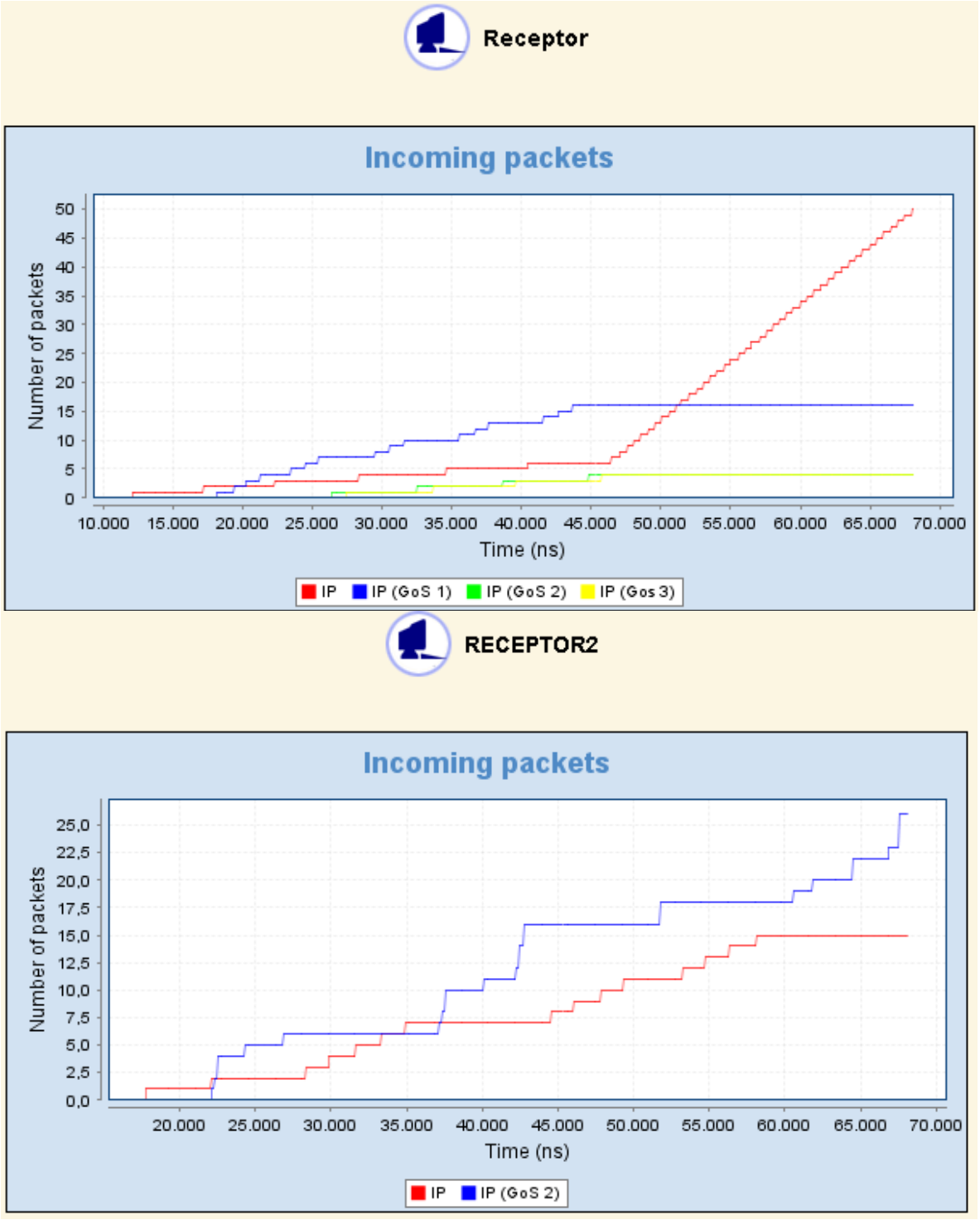
Fuente: http://gitaca.unex.es/opensimmpls , "OpenSimMPLS"

Simulador creado por: Manuel Domínguez Dorado

Red Elaborado por: Las autoras

En la figura se observa que al congestionarse el otro nodo transmitió más paquetes, es decir que cuando existe congestión en un camino el tráfico que satura este enlace se reenruta por el otro incrementando el flujo de datos.

FIGURA 6.17. PAQUETES DE ENTRADA EN EL RECEPTOR



Fuente: http://gitaca.unex.es/opensimmpls , "OpenSimMPLS"

Simulador creado por: Manuel Domínguez Dorado

Red Elaborado por: Las autoras

CAPÍTULO VII

“SIMULAR CON NETWORK SIMULATOR (NS), EL FUNCIONAMIENTO DE LA RED DISEÑADA”

7.1. DEFINICIÓN DE NS

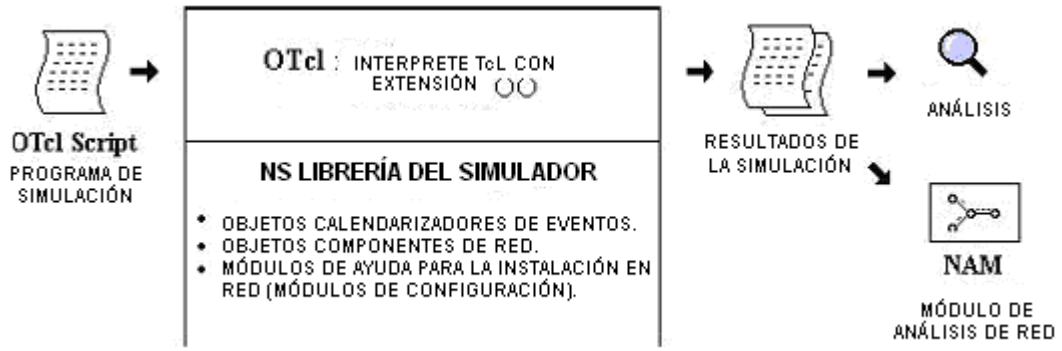
Network Simulator es un simulador discreto de eventos creado por la Universidad de Berkeley para modelar redes de tipo IP. En la simulación se toma en cuenta lo que es la estructura (topología) de la red y el tráfico de paquetes que posee la misma, con el _n de crear una especie de diagnóstico que nos muestre el comportamiento que se obtiene al tener una red con ciertas características.

Actualmente, el proyecto NS es parte de VINT proyecto que desarrolla herramientas para visualizar los resultados de una simulación (por ejemplo, una interfaz gráfica).

Trae implementaciones de protocolos tales como TCP y UDP, que es posible hacerlos comportar como un tráfico FTP, Telnet, Web, CBR y VBR. Maneja diversos mecanismos de colas que se generan en los routers, tales como DropTail, RED, CQB, algoritmo de Dijkstra, etc.

El funcionamiento de Network Simulator se explicaría poco a poco mostrando las partes más generales a las más particulares. Para comenzar se mostraría una vista bastante simplificada de lo que es NS.

FIGURA 7.1. ESQUEMA DE NETWORK SIMULATOR (NS)



Fuente: www.cisco.com

Elaborado por: Las autoras

Se debe recalcar que el análisis realizado con NS se encuentra a nivel de capa cuatro, por lo que se presentará los diagramas de flujo correspondientes al código analizado con la Empresa auspiciante.

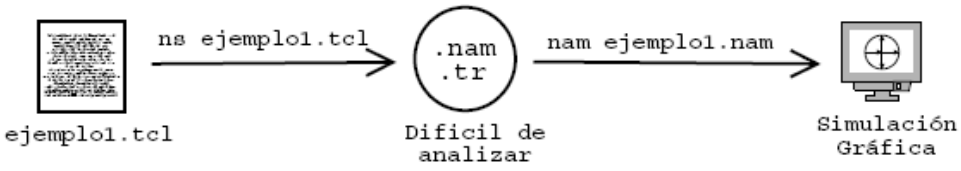
7.2. FUNCIONAMIENTO DE NS

Como se puede observar, se comienza con un script en OTcl que viene a ser lo que el usuario codifica para simular. Es el único INPUT que dá el usuario al programa. El resto es el procesamiento interno de NS.

La simulación queda en un archivo que puede ser bastante incómodo de leer o analizar para el usuario, sin embargo, usando una aplicación especial se puede mostrar mediante una interfaz gráfica.

El script es un archivo escrito en Tcl orientado a objetos, es decir, OTcl, que tiene diversos componentes internos que se muestran en el cuadro del medio de la figura 7.2. En estos componentes se configura la topología de la red, calendariza los eventos, carga las funciones necesarias para la simulación, planifica cuando iniciar o terminar el tráfico de un determinado paquete, entre otras cosas.

FIGURA 7.2. COMPONENTES INTERNOS



Fuente: www.cisco.com

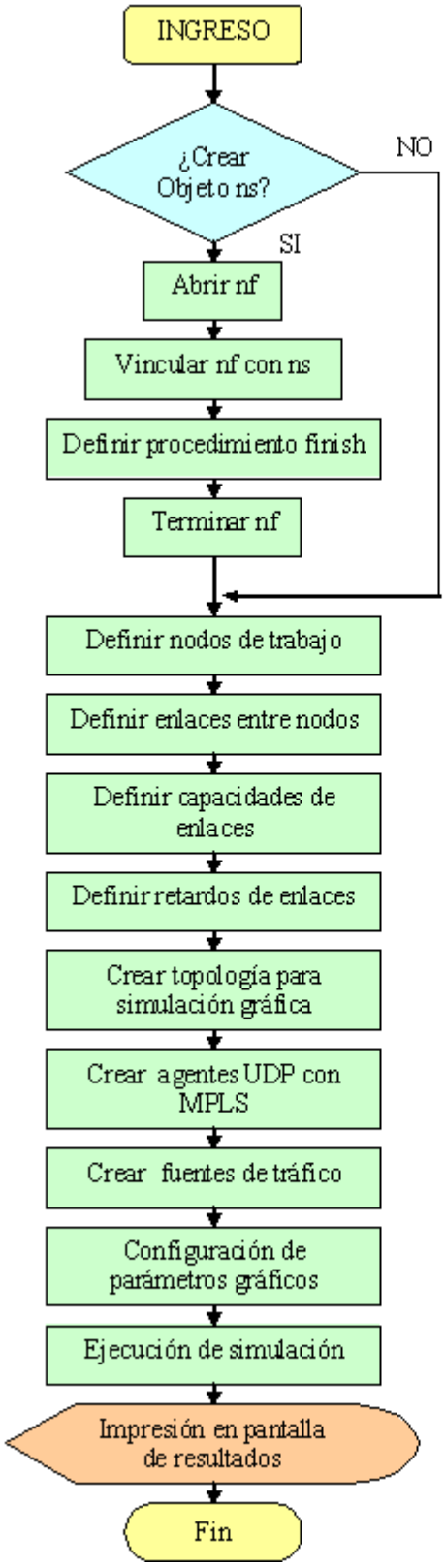
Elaborado por: Las autoras

La instalación y guía básica de usuario se encuentra en los anexos.

A continuación se muestra los diagramas de flujo del código script que se encuentra en los anexos.

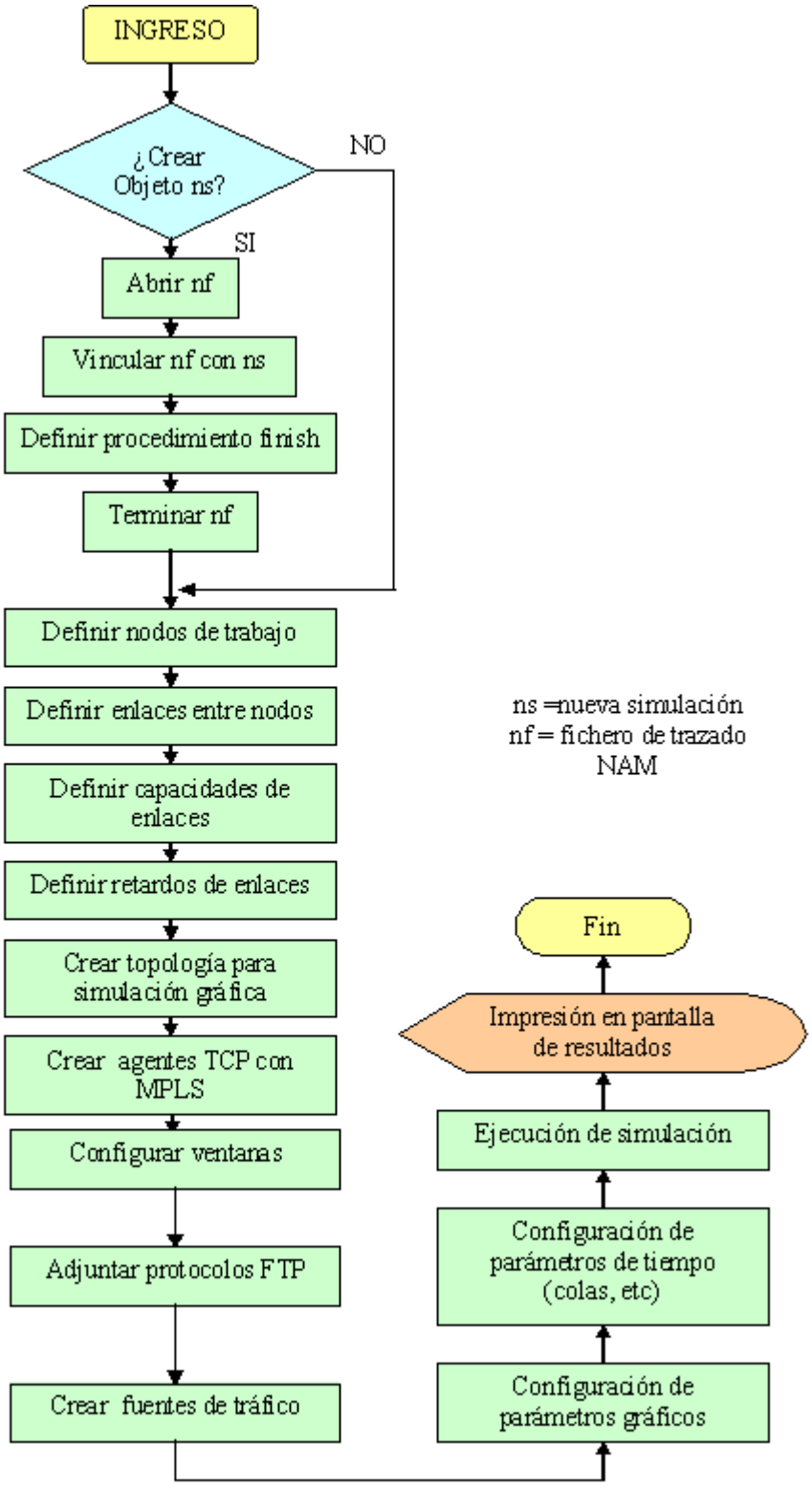
SCRIPT

SIMULACIÓN NS PARA UDP



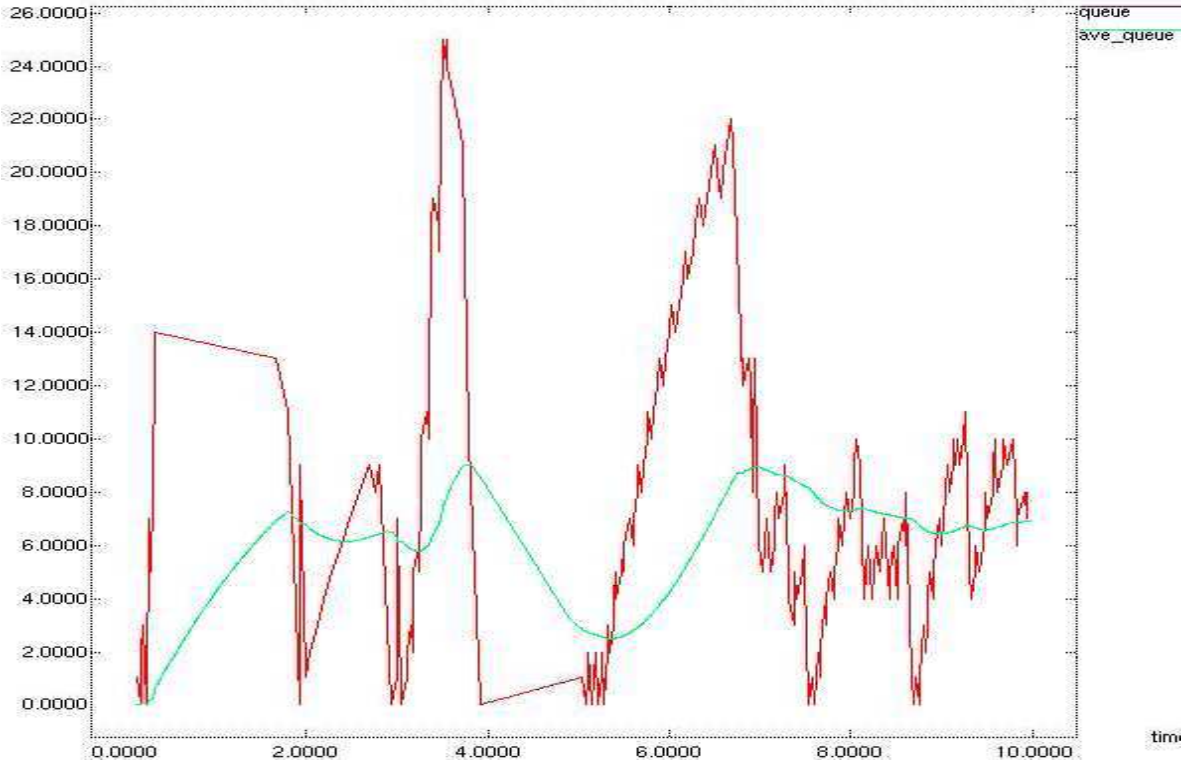
ns =nueva simulación
nf = fichero de trazado NAM

SIMULACIÓN NS PARA TCP



7.3. PRESENTACIÓN DE RESULTADOS

FIGURA 7.3. PRESENTACIÓN DE RESULTADOS



Fuente: Network Simulator

Elaborado por: Las autoras

La figura 7.3. muestra el resultado de un nodo real para tráfico TCP (verde) y UDP (rojo), estos son paquetes de la red MPLS entre dos LSRs.

CAPÍTULO VIII

“CONCLUSIONES
Y
RECOMENDACIONES”

CAPÍTULO VIII

8.1. CONCLUSIONES Y RECOMENDACIONES

- Se requiere de routers con capacidad MPLS o que soporten actualizaciones en su versión de sistema operativo. Los routers utilizados para este diseño son CISCO (serie 12000) sobre los que se puede configurar QoS, TE, y VPNs.

- Las complejidades de una red celular deben estructurarse por locaciones y luego por servicios, ya que esto facilita el estudio de tráfico.

- Se debe analizar el tráfico sin olvidar que tenemos voz y datos sobre los que se debe aplicar QoS y seguridad. Debemos tomar en cuenta que en una red celular se prioriza el tráfico de voz al de datos en caso de congestión.

- MPLS es una alternativa, pero requiere que de ser implementada en una red existente, se tomen precauciones para no interrumpir el servicio, por el cambio en la configuración de routers.

- Es de gran ayuda el uso de simuladores pero siempre se debe sopesar los posibles imprevistos al realizar cambios en la configuración de router y más si requieren updates, ya que una no correcta aplicación de los mismos, conlleva a retrasos en la implementación y posible molestia a los usuarios.

- Se deben establecer enlaces de redundancia en la red, de manera que si se pierde conectividad el tráfico sea reenrutado a un enlace operativo.

- Se concluye que al utilizar valores aproximados a los de la red real para la simulación, se obtuvieron resultados efectivos que representan tanto el tráfico como los problemas reales que se presentan en la operación de la red.

- Se recomienda el uso de simuladores, cuya programación tenga integrada el soporte de configuración para MPLS.

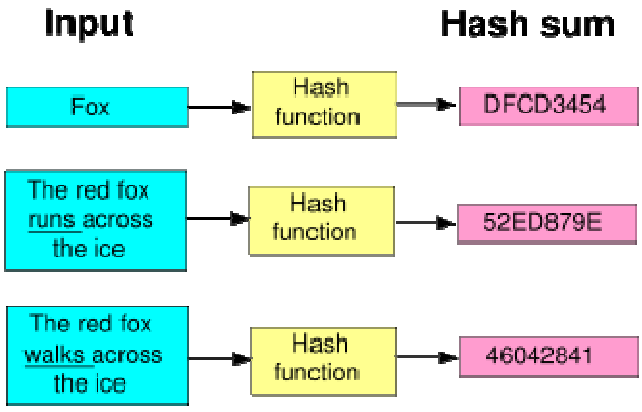
- Se recomienda realizar simulaciones de Ingeniería de Tráfico y Calidad de Servicio, antes de implementar MPLS en la red para diferenciar los problemas y limitaciones de tecnologías tradicionales, así como también las ventajas proporcionadas en un backbone con MPLS.

- Se recomienda el análisis y estudio de protocolos de estado de enlade (OSPF o IS-IS) como IGP, ya que son necesarios para la implementación de Ingeniería de Tráfico.

TERMINOLOGÍA

- Multi-Prototocol Label Switching (MPLS). MPLS se considera fundamental en la construcción de los nuevos cimientos para la Internet del siglo XXI.
- Best effort: Servicios ATM sin calidad garantizada.
- IETF: Fuerza de Trabajo de Ingeniería de Internet.
- Tunneling: Arquitectura diseñada para suministrar los servicios necesarios para implementar cualquier esquema de encapsulación punto a punto estándar.
- T1: Servicio de portadora WAN digital que transmite datos formateados a 1.544 Mbps a través de una red de conmutación telefónica, comúnmente utilizada en los EE.UU. E1: Esquema de transmisión digital de WAN utilizado en Europa, que lleva datos a una velocidad de 2.048 Mbps.
- T3: Servicio de transmisión digital a una velocidad de 44.736 Mbps en los EE.UU. E3: en Europa de 34.368 Mbps.
- UBR: *Velocidad binaria sin especifica:* Clase de servicio definida por el foro ATM para las redes ATM. Permite el envío de cualquier cantidad de datos hasta un máximo especificado a través de la red, pero no ofrece garantías en términos de pérdida y retraso.
- IGP: *Protocolo de gateway interior:* Protocolo de Internet que se utiliza para intercambiar información de enrutamiento dentro de un sistema autónomo. IGRP, OSPF y RIP son ejemplos de IGP de Internet comunes.
- RSVP: *Protocolo de reserva de recursos:* Protocolo que hace posible la reserva de recursos a través de una red IP. Las aplicaciones que se ejecutan en los sistemas finales IP pueden usar RSVP para indicarle a los otros nodos la naturaleza (ancho de banda, fluctuación de fase, ráfaga máxima, etc.) de los flujos de paquetes que desean recibir.
- CIR: *Velocidad de información suscrita:* La velocidad a la que una red Frame Relay acepta transferir información bajo condiciones normales, con un promedio sobre un incremento de tiempo mínimo. Es una de las más importantes métricas de tarifa negociada.
- IPSec: *Internet Protocol Security:* IP Seguro. Estándar abierto para garantizar seguridad en comunicaciones privadas en redes IP.

- Hash: En informática, se refiere a una función o método para generar claves o llaves que representen de manera casi unívoca a un documento, registro, archivo, etc., resumir o identificar un dato a través de la probabilidad, utilizando una función hash o algoritmo hash. Un **hash** es el resultado de dicha función o algoritmo.



REFERENCIAS BIBLIOGRÁFICAS

- http://gitaca.unex.es/opensimmpls/
- MPLS, ESTUDIO PRACTICO, http://www.mplsrc.com/index.shtml
- FRAMEIP, MPLS CON CISCO,http://www.frameip.com/mpls-cisco/
- FRAMEIP, MPLS, http://www.frameip.com/mpls/
  http://www.rsa.com/rsalabs/faq/index.html
  http://www.rsasecurity.com/rsalabs/faq/index.html
- Cryptographic Algorithms
  http://www.ssh.fi/tech/crypto/algorithms.html
- A short history of crypto
  http://webhome.idirect.com/~jproc/crypto/crypto_hist.html
- AES (Advanced Encryption Standard)
  http://csrc.nist.gov/encryption/aes/
- The Block Cipher Lounge - AES
  http://www.ii.uib.no/~larsr/aes.html
- The Rijndael Block Cipher (AES)
  http://rijndael.com/
- TEA - Tiny Encryption Algorithm
  http://vader.brad.ac.uk/tea/tea.shtml
- Block Cyphers and Cryptoanalysis - Fauzan Mirza
  http://mesa-sys.com/~eternal/fame/solutions/report.pdf
- Speed Comparison of Popular Crypto Algorithms
  http://www.eskimo.com/~weidai/benchmarks.html
- The Mathematical Guts of RSA Encryption
  http://world.std.com/~franl/crypto/rsa-guts.html
- The CAST-128 Encryption Algorithm
  http://finecrypt.tripod.com/cast.html
- "Weaknesses in the Key Scheduling Algorithm of RC4" by Scott Fluhrer, Itsik
- Mantin and Adi Shamir.
  http://www.eyetap.org/~rguerra/toronto2001/rc4_ksaproc.pdf
- whatis.com - Definitions for thousands of the most current IT-related words
  http://whatis.techtarget.com

- Counterpane Labs - A Cryptographic Evaluation of IPsec
  http://www.counterpane.com/ipsec.html
- VPN Info on the World Wide Web (Tina Bird)
  http://kubarb.phsx.ukans.edu/~tbird/vpn.html
- Virtual Private Networks Frequently Asked Questions (Tina Bird)
  http://kubarb.phsx.ukans.edu/~tbird/vpn/FAQ.html
- Microsoft Point-Point Tunneling Protocol (PPTP) FAQ
  http://www.microsoft.com/ntserver/ProductInfo/faqs/PPTPfaq.asp
- VPNs: Virtually Anything? - A Core Competence Industry Report
  http://www.corecom.com/html/vpn.html
- http://www.corecom.com/external/vpn/compare.html
- RFC 2661 - Layer Two Tunneling Protocol "L2TP"
  http://www.landfield.com/rfcs/rfc2661.html
- RFC 1661 - The Point-to-Point Protocol (PPP)
  http://www.faqs.org/rfcs/rfc1661.html
- RFC 2246 - The TLS Protocol Version 1.0
  http://www.ietf.org/rfc/rfc2246.txt
- RFC 2631 - Diffie-Hellman Key Agreement Method
  http://www.ietf.org/rfc/rfc2631.txt
- RFC 2144 - The CAST-128 Encryption Algorithm
  http://www.faqs.org/rfcs/rfc2144.html
- RFC 2612 - The CAST-256 Encryption Algorithm
  http://www.faqs.org/rfcs/rfc2612.html
- RFC 1492 - An Access Control Protocol, Sometimes Called TACACS
  http://www.ietf.org/rfc/rfc1492.txt
- RFC 2138 - Remote Authentication Dial In User Service (RADIUS)
  http://www.ietf.org/rfc/rfc2138.txt
- RFC-ES - Traducción al castellano de RFCs
  http://www.rfc-es.org/
- TACACS+
  http://www.mentortech.com/learn/welcher/papers/tacacs.htm
- MPLS Resource Center
  http://www.mplsrc.com/

- RSA Laboratories' Frequently Asked Questions About Today's Cryptography
- Cisco - IPSec Overview
  http://www.cisco.com/warp/public/cc/techno/protocol/ipsecur/prodlit/ipsec_ov.htm

ANEXOS

LISTA DE TÚNELES POR EQUIPO

Caracteristicas	Fortigate 100	Vigor 2200	Cisco 2600	Firebox SSL Core	OfficeConnect 3Com
VPN					
Administración de permisos por puertos	*				*
Administración de rangos de ip con acceso	*		*	*	*
Numero de Túneles	50	30	25	10 + opc	10
Soporte Site to Site	*	*	*	*	
Soporte Site to Client	*		*	*	
Administración Web	*	*		*	
Soporte IPSEC	*	*	*	*	
Static IP	*		*	*	
Dial-UP Users	*	*	*	*	*
Dinamic DNS	*				
Modo Aggressive	*	*	*	*	*
Modo Main	*	*	*	*	*
Método de autentificación Preshared Key	*	*	*	*	*
Método de autentificación RSA Signature	*		*		
Soporte para encriptación DES	*	*	*	*	
Soporte para encriptación 3DES	*	*	*	*	
Soporte para encriptación AES 128	*				
Soporte para encriptación AES 192	*				
Soporte para encriptación AES 256	*				
soporte para autentificación SHA1	*	*	*	*	*
soporte para autentificación MD5	*	*	*	*	*
Keylife configurable	*		*	*	
Soporte Xauth Client	*				
Soporte Xauth Server	*				
Nat-traversal	*		*		
VPN redundante	*				
Autokey Keep Alive	*	*	*	*	
VPN Concentrador	*		*		
Ping Generator	*				
Listado de conexiones activas	*	*		*	
Soporte PPTP	*		*	*	*
Soporte L2TP	*			*	*
Soporte L2TP over IPSec		*	*		*
Soporte Certificados	*		*	*	
Generación de Certificados	*				
Importación de Certificados	*		*	*	

RFC 4305

Network Working Group
Request for Comments: 4305
Obsoletes: 2404, 2406
December 2005
Category: Standards Track

Cryptographic Algorithm Implementation Requirements for Encapsulating Security Payload (ESP) and Authentication Header (AH)

Status of This Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Abstract

The IPsec series of protocols makes use of various cryptographic algorithms in order to provide security services. The Encapsulating Security Payload (ESP) and the Authentication Header (AH) provide two mechanisms for protecting data being sent over an Ipsec Security Association (SA). To ensure interoperability between disparate implementations, it is necessary to specify a set of mandatory-to-implement algorithms to ensure that there is at least one algorithm that all implementations will have available. This document defines the current set of mandatory-to-implement algorithms for ESP and AH as well as specifying algorithms that should be implemented because they may be promoted to mandatory at some future time.

Table of Contents

1. Introduction

The Encapsulating Security Payload (ESP) and the Authentication Header (AH) provide two mechanisms for protecting data being sent over an IPsec Security Association (SA) [IPsec, ESP, AH]. To ensure interoperability between disparate implementations, it is necessary to specify a set of mandatory-to-implement algorithms to ensure that there is at least one algorithm that all implementations will have available. This document defines the current set of mandatory-to-implement algorithms for ESP and AH as well as specifying algorithms that should be implemented because they may be promoted to mandatory at some future time.

The nature of cryptography is that new algorithms surface continuously and existing algorithms are continuously attacked. An algorithm believed to be strong today may be demonstrated to be weak tomorrow. Given this, the choice of mandatory-to-implement algorithm should be conservative so as to minimize the likelihood of it being compromised quickly. Thought should also be given to performance considerations as many uses of IPsec will be in environments where performance is a concern.

Finally, we need to recognize that the mandatory-to-implement algorithm(s) may need to change over time to adapt to the changing world. For this reason, the selection of mandatory-to-implement algorithms is not included the main IPsec, ESP, or AH specifications. It is instead placed in this document. As the choice of algorithm changes, only this document should need to be updated.

Ideally, the mandatory-to-implement algorithm of tomorrow should already be available in most implementations of IPsec by the time it is made mandatory. To facilitate this, we will attempt to identify such algorithms (as they are known today) in this document. There is no guarantee that the algorithms we believe today may be mandatory in the future will in fact become so. All algorithms known today are subject to cryptographic attack and may be broken in the future.

2. Requirements Terminology

Keywords "MUST", "MUST NOT", "REQUIRED", "SHOULD", "SHOULD NOT" and "MAY" that appear in this document are to be interpreted as described in [RFC2119].

We define some additional terms here:

SHOULD+ This term means the same as SHOULD. However, it is likely that an algorithm marked as SHOULD+ will be promoted at some future time to be a MUST.
SHOULD- This term means the same as SHOULD. However, it is likely that an algorithm marked as SHOULD- will be deprecated to a MAY or worse in a future version of this document.
MUST- This term means the same as MUST. However, we expect that at some point in the future this algorithm will no longer be a MUST.

3. Algorithm Selection

For IPsec implementations to interoperate, they must support one or more security algorithms in common. This section specifies the security algorithm implementation requirements for standards-conformant ESP and AH implementations. The security algorithms actually used for any particular ESP or AH security association are determined by a negotiation mechanism, such as the Internet Key Exchange (IKE [RFC2409, IKEv2]) or pre-establishment.

Of course, additional standard and proprietary algorithms beyond those listed below can be implemented.

3.1. Encapsulating Security Payload

The implementation conformance requirements for security algorithms for ESP are given in the tables below. See Section 2 for definitions of the values in the "Requirement" column.

3.1.1. ESP Encryption and Authentication Algorithms

These tables list encryption and authentication algorithms for the IPsec Encapsulating Security Payload protocol.

Requirement	Encryption Algorithm (notes)
MUST	NULL (1)
MUST-	TripleDES-CBC [RFC2451]
SHOULD+	AES-CBC with 128-bit keys [RFC3602]
SHOULD	AES-CTR [RFC3686]
SHOULD NOT	DES-CBC [RFC2405] (3)

Requirement	Authentication Algorithm (notes)
MUST	HMAC-SHA1-96 [RFC2404]
MUST	NULL (1)
SHOULD+	AES-XCBC-MAC-96 [RFC3566]
MAY	HMAC-MD5-96 [RFC2403] (2)

Notes:

(1) Since ESP encryption and authentication are optional, support for the two "NULL" algorithms is required to maintain consistency with the way these services are negotiated. Note that while authentication and encryption can each be "NULL", they MUST NOT both be "NULL".
(2) Weaknesses have become apparent in MD5; however, these should not affect the use of MD5 with HMAC.
(3) DES, with its small key size and publicly demonstrated and open-design special-purpose cracking hardware, is of questionable security for general use.

3.1.2. ESP Combined Mode Algorithms

As specified in [ESP], combined mode algorithms are supported that provide both confidentiality and authentication services. Support of such algorithms will require proper structuring of ESP implementations. Under many circumstances, combined mode algorithms provide significant efficiency and throughput advantages. Although there are no suggested or required combined algorithms at this time, AES-CCM [CCM], which has been adopted as the preferred mode for security in IEEE 802.11 [802.11i], is expected to be of interest in the near future.

3.2. Authentication Header

The implementation conformance requirements for security algorithms for AH are given below. See Section 2 for definitions of the values in the "Requirement" column. As you would suspect, all of these algorithms are authentication algorithms.

Requirement	Algorithm (notes)
MUST	HMAC-SHA1-96 [RFC2404]
SHOULD+	AES-XCBC-MAC-96 [RFC3566]
MAY	HMAC-MD5-96 [RFC2403] (1)

Note:

(1) Weaknesses have become apparent in MD5; however, these should not affect the use of MD5 with HMAC.

4. Security Considerations

The security of cryptographic-based systems depends on both the strength of the cryptographic algorithms chosen and the strength of the keys used with those algorithms. The security also depends on the engineering and administration of the protocol used by the system to ensure that there are no non-cryptographic ways to bypass the security of the overall system.

This document concerns itself with the selection of cryptographic algorithms for the use of ESP and AH, specifically with the selection of mandatory-to-implement algorithms. The algorithms identified in this document as "MUST implement" or "SHOULD implement" are not known to be broken at the current time, and cryptographic research so far leads us to believe that they will likely remain secure into the foreseeable future. However, this is not necessarily forever. We would therefore expect that new revisions of this document will be issued from time to time that reflect the current best practice in this area.

5. Acknowledgement

Much of the wording herein was adapted from RFC 4307, "Cryptographic Algorithms for Use in the Internet Key Exchange Version 2", by Jeffrey I. Schiller.

6. Changes from RFC 2402 and 2406

[RFC2402] and [RFC2406] defined the IPsec Authentication Header and IPsec Encapsulating Security Payload. Each specified the implementation requirements for cryptographic algorithms for their respective protocols. They have now been replaced with [AH] and [ESP], which do not specify cryptographic algorithm implementation requirements, and this document, which specifies such requirements for both [AH] and [ESP].

The implementation requirements are compared below:

Old Req.	Old RFC(s)	New Requirement	Algorithm (notes)
MUST	2406	SHOULD NOT	DES-CBC [RFC2405] (1)
MUST	2402 2406	MAY	HMAC-MD5-96 [RFC2403]
MUST	2402 2406	MUST	HMAC-SHA1-96 [RFC2404]

Note:

(1) The IETF deprecated the use of single DES years ago and has not included it in any new standard for some time (see IESG note on the first page of [RFC2407]). But this document represents the first standards-track recognition of that deprecation by specifying that implementations SHOULD NOT provide single DES. The US Government National Institute of Standards and Technology (NIST) has formally recognized the weakness of single DES by a notice published in the 26 July 2004 US Government Federal Register (Docket No. 040602169-4169-01) proposing to withdraw it as a US Government Standard. Triple DES demains approved by both the IETF and NIST.

7. Normative References

[AH] Kent, S., "IP Authentication Header", RFC 4302, December 2005.

[ESP] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, December 2005.

[IPsec] Kent, S., "Security Architecture for the Internet Protocol", RFC 4301, December 2005.

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

[RFC2403] Madson, C. and R. Glenn, "The Use of HMAC-MD5-96 within ESP and AH", RFC 2403, november 1998.

[RFC2404] Madson, C. and R. Glenn, "The Use of HMAC-SHA-1-96 within ESP and AH", RFC 2404, November 1998.

[RFC2405] Madson, C. and N. Doraswamy, "The ESP DES-CBC Cipher Algorithm With Explicit IV", RFC 2405, November 1998.

[RFC3566] Frankel, S. and H. Herbert, "The AES-XCBC-MAC-96 Algorithm and Its Use With IPsec", RFC 3566, September 2003.

[RFC3602] Frankel, S., Glenn, R., and S. Kelly, "The AES-CBC Cipher Algorithm and Its Use with IPsec", RFC 3602, September 2003.

[RFC3686] Housley, R., "Using Advanced Encryption Standard (AES) Counter Mode With IPsec Encapsulating Security Payload (ESP)", RFC 3686, January 2004.

8. Informative References

[802.11i] LAN/MAN Specific Requirements Part 11: Wireless Medium Access Control (MAC) and physical layer (PHY) specifications: Medium Access Control (MAC) Security Enhancements, IEEE Std 802.11i, June 2004.

[JIS] Schiller, J., "Cryptographic Algorithms for Use in the Internet Key Exchange Version 2 (IKEv2)", RFC 4307, December 2005.

[CCM] Housley, R., "Using Advanced Encryption Standard (AES) Counter Mode With IPsec Encapsulating Security Payload (ESP)", RFC 3686, January 2004.

[IKEv2] Kaufman, C., Ed., "Internet Key Exchange (IKEv2) Protocol", RFC 4306, December 2005.

[RFC791] Postel, J., "Internet Protocol", STD 5, RFC 791, September 1981.

[RFC2402] Kent, S. and R. Atkinson, "IP Authentication Header", RFC 2402, November 1998.

[RFC2406] Kent, S. and R. Atkinson, "IP Encapsulating Security Payload (ESP)", RFC 2406, November 1998. RFC 4305 Cryptographic Algorithms for ESP & AH December 2005

[RFC2407] Piper, D., "The Internet IP Security Domain of Interpretation for ISAKMP", RFC 2407, November 1998.

[RFC2409] Harkins, D. and D. Carrel, "The Internet Key Exchange (IKE)", RFC 2409, November 1998.

Author's Address

Donald E. Eastlake 3rd
Motorola Laboratories
155 Beaver Street
Milford, MA 01757 USA

Phone: +1-508-786-7554 (w)
+1-508-634-2066 (h)
EMail: Donald.Eastlake@Motorola.com

Full Copyright Statement

RFC 2547

Network Working Group
Request for Comments: 2547
Category: Informational
March 1999

E. Rosen
Y. Rekhter
Cisco Systems, Inc.

BGP/MPLS VPNs

Status of this Memo

This memo provides information for the Internet community. It does not specify an Internet standard of any kind. Distribution of this memo is unlimited.

Abstract

This document describes a method by which a Service Provider with an IP backbone may provide VPNs (Virtual Private Networks) for its customers. MPLS (Multiprotocol Label Switching) is used for forwarding packets over the backbone, and BGP (Border Gateway Protocol) is used for distributing routes over the backbone. The primary goal of this method is to support the outsourcing of IP backbone services for enterprise networks. It does so in a manner which is simple for the enterprise, while still scalable and flexible for the Service Provider, and while allowing the Service Provider to add value. These techniques can also be used to provide a VPN which itself provides IP service to customers.

Table of Contents

1. Introduction

1.1. Virtual Private Networks

Consider a set of "sites" which are attached to a common network which we may call the "backbone". Let's apply some policy to create a number of subsets of that set, and let's impose the following rule: two sites may have IP interconnectivity over that backbone only if at least one of these subsets contains them both.

The subsets we have created are "Virtual Private Networks" (VPNs). Two sites have IP connectivity over the common backbone only if there is some VPN which contains them both. Two sites which have no VPN in common have no connectivity over that backbone. If all the sites in a VPN are owned by the same enterprise, the VPN is a corporate "intranet". If the various sites in a VPN are owned by different enterprises, the VPN is an "extranet". A site can be in more than one VPN; e.g., in an intranet and several extranets. We regard both intranets and extranets as VPNs. In general, when we use the term VPN we will not be eistinguishing between intranets and extranets.

We wish to consider the case in which the backbone is owned and operated by one or more Service Providers (SPs). The owners of the sites are the "customers" of the SPs. The policies that determine whether a particular collection of sites is a VPN are the policies of the customers. Some customers will want the implementation of these policies to be entirely the responsibility of the SP. Other customers may want to implement these policies themselves, or to share with the SP the responsibility for implementing these policies. In this document, we are primarily discussing mechanisms that may be used to implement these policies. The mechanisms we describe are general enough to allow these policies to be implemented either by the SP alone, or by a VPN customer together with the SP. Most of the discussion is focused on the former case, however.

The mechanisms discussed in this document allow the implementation of a wide range of policies. For example, within a given VPN, we can allow every site to have a direct route to every other site ("full mesh"), or we can restrict certain pairs of sites from having direct routes to each other ("partial mesh").

In this document, we are particularly interested in the case where the common backbone offers an IP service. We are primarily concerned with the case in which an enterprise is outsourcing its backbone to a service provider, or perhaps to a set of service providers, with which it maintains contractual relationships. We are not focused on providing VPNs over the public Internet.

In the rest of this introduction, we specify some properties which VPNs should have. The remainder of this document outlines a VPN model which has all these properties. The VPN Model of this document appears to be an instance of the framework described in [4].

1.2. Edge Devices

We suppose that at each site, there are one or more Customer Edge (CE) devices, each of which is attached via some sort of data link (e.g., PPP, ATM, ethernet, Frame Relay, GRE tunnel, etc.) to one or more provider Edge (PE) routers.

If a particular site has a single host, that host may be the CE device. If a particular site has a single subnet, that the CE device may be a switch. In general, the CE device can be expected to be a router, which we call the CE router.

We will say that a PE router is attached to a particular VPN if it is attached to a CE device which is in that VPN. Similarly, we will say that a PE router is attached to a particular site if it is attached to a CE device which is in that site.

When the CE device is a router, it is a routing peer of the PE(s) to which it is attached, but is not a routing peer of CE routers at other sites. Routers at different sites do not directly exchange routing information with each other; in fact, they do not even need to know of each other at all (except in the case where this is necessary for security purposes, see section 9). As a consequence, very large VPNs (i.e., VPNs with a very large number of sites) are easily supported, while the routing strategy for each individual site is greatly simplified.

It is important to maintain clear administrative boundaries between the SP and its customers (cf. [4]). The PE and P routers should be administered solely by the SP, and the SP's customers should not have any management access to it. The CE devices should be administered solely by the customer (unless the customer has contracted the management services out to the SP).

1.3. VPNs with Overlapping Address Spaces

We assume that any two non-intersecting VPNs (i.e., VPNs with no sites in common) may have overlapping address spaces; the same address may be reused, for different systems, in different VPNs. As long as a given endsystem has an address which is unique within the scope of the VPNs that it belongs to, the endsystem itself does not need to know anything about VPNs.

In this model, the VPN owners do not have a backbone to administer, not even a "virtual backbone". Nor do the SPs have to administer a separate backbone or "virtual backbone" for each VPN. Site-to-site routing in the backbone is optimal (within the constraints of the policies used to form the VPNs), and is not constrained in any way by an artificial "virtual topology" of tunnels.

1.4. VPNs with Different Routes to the Same System

Although a site may be in multiple VPNs, it is not necessarily the case that the route to a given system at that site should be the same in all the VPNs. Suppose, for example, we have an intranet consisting of sites A, B, and C, and an extranet consisting of A, B, C, and the "foreign" site D. Suppose that at site A there is a server, and we want clients from B, C, or D to be able to use that server. Suppose also that at site B there is a firewall. We want all the traffic from site D to the server to pass through the firewall, so that traffic from the extranet can be access controlled. However, we don't want traffic from C to pass through the firewall on the way to the server, since this is intranet traffic.

This means that it needs to be possible to set up two routes to the server. One route, used by sites B and C, takes the traffic directly to site A. The second route, used by site D, takes the traffic instead to the firewall at site B. If the firewall allows the traffic to pass, it then appears to be traffic coming from site B, and follows the route to site A.

1.5. Multiple Forwarding Tables in PEs

Each PE router needs to maintain a number of separate forwarding tables. Every site to which the PE is attached must be mapped to one of those forwarding tables. When a packet is received from a particular site, the forwarding table associated with that site is consulted in order to determine how to route the packet. The forwarding table associated with a particular site S is populated only with routes that lead to other sites which have at least one VPN in common with S. This prevents communication between sites which have no VPN in common, and it allows two VPNs with no site in common to use address spaces that overlap with each other.

1.6. SP Backbone Routers

The SP's backbone consists of the PE routers, as well as other routers (P routers) which do not attach to CE devices.

If every router in an SP's backbone had to maintain routing information for all the VPNs supported by the SP, this model would have severe scalability problems; the number of sites that could be supported would be

limited by the amount of routing information that could be held in a single router. It is important to require therefore that the routing information about a particular VPN be present ONLY in those PE routers which attach to that VPN. In particular, the P routers should not need to have ANY per-VPN routing information whatsoever.

VPNs may span multiple service providers. We assume though that when the path between PE routers crosses a boundary between SP networks, it does so via a private peering arrangement, at which there exists mutual trust between the two providers. In particular, each provider must trust the other to pass it only correct routing information, and to pass it labeled (in the sense of MPLS [9]) packets only if those packets have been labeled by trusted sources. We also assume that it is possible for label switched paths to cross the boundary between service providers.

1.7. Security

A VPN model should, even without the use of cryptographic security measures, provide a level of security equivalent to that obtainable when a level 2 backbone (e.g., Frame Relay) is used. That is, in the absence of misconfiguration or deliberate interconnection of different VPNs, it should not be possible for systems in one VPN to gain access to systems in another VPN.

It should also be possible to deploy standard security procedures.

2. Sites and CEs

From the perspective of a particular backbone network, a set of IP systems constitutes a site if those systems have mutual IP interconnectivity, and communication between them occurs without use of the backbone. In general, a site will consist of a set of systems which are in geographic proximity. However, this is not universally true; two geographic locations connected via a leased line, over which OSPF is running, will constitute a single site, because communication between the two locations does not involve the use of the backbone.

A CE device is always regarded as being in a single site (though as we shall see, a site may consist of multiple "virtual sites"). A site, however, may belong to multiple VPNs.

A PE router may attach to CE devices in any number of different sites, whether those CE devices are in the same or in different VPNs. A CE device may, for robustness, attach to multiple PE routers, of the same or of different service providers. If the CE device is a router, the PE router and the CE router will appear as router adjacencies to each other.

While the basic unit of interconnection is the site, the architecture described herein allows a finer degree of granularity in the control of interconnectivity. For example, certain systems at a site may be members of an intranet as well as members of one or more extranets, while other systems at the same site may be restricted to being members of the intranet only.

3. Per-Site Forwarding Tables in the PEs

Each PE router maintains one or more "per-site forwarding tables". Every site to which the PE router is attached is associated with one of these tables. A particular packet's IP destination address is looked up in a particular per-site forwarding table only if that packet has arrived directly from a site which is associated with that table.

How are the per-site forwarding tables populated?

As an example, let PE1, PE2, and PE3 be three PE routers, and let CE1, CE2, and CE3 be three CE routers. Suppose that PE1 learns, from CE1, the routes which are reachable at CE1's site. If PE2 and PE3 are attached respectively to CE2 and CE3, and there is some VPN V containing CE1, CE2, and CE3, then PE1 uses BGP to distribute to PE2 and PE3 the routes which it has learned from CE1. PE2 and PE3 use these routes to populate the forwarding tables which they associate respectively with the sites of CE2 and CE3. Routes from sites which are not in VPN V do not appear in these forwarding tables, which means that packets from CE2 or CE3 cannot be sent to sites which are not in VPN V.

If a site is in multiple VPNs, the forwarding table associated with that site can contain routes from the full set of VPNs of which the site is a member. A PE generally maintains only one forwarding table per site, even if it is multiply connected to that site. Also, different sites can share the same forwarding table if they are meant to use exactly the same set of routes.

Suppose a packet is received by a PE router from a particular directly attached site, but the packet's destination address does not match any entry in the forwarding table associated with that site. If the SP is not providing Internet access for that site, then the packet is discarded as undeliverable. If the SP is providing Internet access for that site, then the PE's Internet forwarding table will be consulted. This means that in general, only one forwarding table per PE need ever contain routes from the Internet, even if Internet access is provided.

To maintain proper isolation of one VPN from another, it is important that no router in the backbone accept a labeled packet from any adjacent non-backbone device unless (a) the label at the top of the label stack was actually distributed by the backbone router to the non-backbone device, and (b) the backbone router can determine that use of that label will cause the packet to leave the backbone before any labels lower in the stack will be inspected, and before the IP header will be inspected. These restrictions are necessary in order to prevent packets from entering a VPN where they do not belong.

The per-site forwarding tables in a PE are ONLY used for packets which arrive from a site which is directly attached to the PE. They are not used for routing packets which arrive from other routers that belong to the SP backbone. As a result, there may be multiple different routes to the same system, where the route followed by a given packet is determined by the site from which the packet enters the backbone. E.g., one may have one route to a given system for packets from the extranet (where the route leads to a firewall), and a different route to the same system for packets from the intranet (including packets that have already passed through the firewall).

3.1. Virtual Sites

In some cases, a particular site may be divided by the customer into several virtual sites, perhaps by the use of VLANs. Each virtual site may be a member of a different set of VPNs. The PE then needs to contain a separate forwarding table for each virtual site. For example, if a CE supports VLANs, and wants each VLAN mapped to a separate VPN, the packets sent between CE and PE could be contained in the site's VLAN encapsulation, and this could be used by the PE, along with the interface over which the packet is received, to assign the packet to a particular virtual site.

Alternatively, one could divide the interface into multiple "sub- interfaces" (particularly if the interface is Frame Relay or ATM), and assign the packet to a VPN based on the sub-interface over which it arrives. Or one could simply use a different interface for each virtual site. In any case, only one CE router is ever needed per site, even if there are multiple virtual sites. Of course, a different CE router could be used for each virtual site, if that is desired.

Note that in all these cases, the mechanisms, as well as the policy, for controlling which traffic is in which VPN are in the hand of the customer.

If it is desired to have a particular host be in multiple virtual sites, then that host must determine, for each packet, which virtual site the packet is associated with. It can do this, e.g., by sending packets from different virtual sites on different VLANs, our out different network interfaces.

These schemes do NOT require the CE to support MPLS. Section 8 contains a brief discussion of how the CE might support multiple virtual sites if it does support MPLS.

4. VPN Route Distribution via BGP

PE routers use BGP to distribute VPN routes to each other (more accurately, to cause VPN routes to be distributed to each other). A BGP speaker can only install and distribute one route to a given address prefix. Yet we allow each VPN to have its own address space, which means that the same address can be used in any number of VPNs, where in each VPN the address denotes a different system. It follows that we need to allow BGP to install and distribute multiple routes to a single IP address prefix. Further, we must ensure that

POLICY is used to determine which sites can be use which routes; given that several such routes are installed by BGP, only one such must appear in any particular per-site forwarding table.

We meet these goals by the use of a new address family, as specified below.

4.1. The VPN-IPv4 Address Family

The BGP Multiprotocol Extensions [3] allow BGP to carry routes from multiple "address families". We introduce the notion of the "VPN- IPv4 address family". A VPN-IPv4 address is a 12-byte quantity, beginning with an 8-byte "Route Distinguisher (RD)" and ending with a 4-byte IPv4 address. If two VPNs use the same IPv4 address prefix, the PEs translate these into unique VPN-IPv4 address prefixes. This ensures that if the same address is used in two different VPNs, it is possible to install two completely different routes to that address, one for each VPN.

The RD does not by itself impose any semantics; it contains no information about the origin of the route or about the set of VPNs to which the route is to be distributed. The purpose of the RD is solely to allow one to create distinct routes to a common IPv4 address prefix. Other means are used to determine where to redistribute the route (see section 4.2).

The RD can also be used to create multiple different routes to the very same system. In section 3, we gave an example where the route to a particular server had to be different for intranet traffic than for extranet traffic. This can be achieved by creating two different VPN-IPv4 routes that have the same IPv4 part, but different RDs. This allows BGP to install multiple different routes to the same system, and allows policy to be used (see section 4.2.3) to decide which packets use which route.

The RDs are structured so that every service provider can administer its own "numbering space" (i.e., can make its own assignments of RDs), without conflicting with the RD assignments made by any other service provider. An RD consists of a two-byte type field, an administrator field, and an assigned number field. The value of the type field determines the lengths of the other two fields, as well as the semantics of the administrator field. The administrator field identifies an assigned number authority, and the assigned number field contains a number which has been assigned, by the identified authority, for a particular purpose. For example, one could have an RD whose administrator field contains an Autonomous System number (ASN), and whose (4-byte) number field contains a number assigned by the SP to whom IANA has assigned that ASN. RDs are given this structure in order to ensure that an SP which provides VPN backbone service can always create a unique RD when it needs to do so. However, the structuring provides no semantics. When BGP compares two such address prefixes, it ignores the structure entirely.

If the Administrator subfield and the Assigned Number subfield of a VPN-IPv4 address are both set to all zeroes, the VPN-IPv4 address is considered to have exactly the same meaning as the corresponding globally unique IPv4 address. In particular, this VPN-IPv4 address and the corresponding globally unique IPv4 address will be considered comparable by BGP. In all other cases, a VPN-IPv4 address and its corresponding globally unique IPv4 address will be considered noncomparable by BGP.

A given per-site forwarding table will only have one VPN-IPv4 route for any given IPv4 address prefix. When a packet's destination address is matched against a VPN-IPv4 route, only the IPv4 part is actually matched.

A PE needs to be configured to associate routes which lead to particular CE with a particular RD. The PE may be configured to associate all routes leading to the same CE with the same RD, or it may be configured to associate different routes with different RDs, even if they lead to the same CE.

4.2. Controlling Route Distribution

In this section, we discuss the way in which the distribution of the VPN-IPv4 routes is controlled.

4.2.1. The Target VPN Attribute

Every per-site forwarding table is associated with one or more "Target VPN" attributes.

When a VPN-IPv4 route is created by a PE router, it is associated with one or more "Target VPN" attributes. These are carried in BGP as attributes of the route.

Any route associated with Target VPN T must be distributed to every PE router that has a forwarding table associated with Target VPN T. When such a route is received by a PE router, it is eligible to be installed in each of the PE's per-site forwarding tables that is associated with Target VPN T. (Whether it actually gets installed depends on the outcome of the BGP decision process.)

In essence, a Target VPN attribute identifies a set of sites. Associating a particular Target VPN attribute with a route allows that route to be placed in the per-site forwarding tables that are used for routing traffic which is received from the corresponding sites.

There is a set of Target VPNs that a PE router attaches to a route received from site S. And there is a set of Target VPNs that a PE router uses to determine whether a route received from another PE router could be placed in the forwarding table associated with site S. The two sets are distinct, and need not be the same.

The function performed by the Target VPN attribute is similar to that performed by the BGP Communities Attribute. However, the format of the latter is inadequate, since it allows only a two-byte numbering space. It would be fairly straightforward to extend the BGP Communities Attribute to provide a larger numbering space. It should also be possible to structure the format, similar to what we have described for RDs (see section 4.1), so that a type field defines the length of an administrator field, and the remainder of the attribute is a number from the specified administrator's numbering space. When a BGP speaker has received two routes to the same VPN-IPv4 prefix, it chooses one, according to the BGP rules for route preference.

Note that a route can only have one RD, but it can have multiple Target VPNs. In BGP, scalability is improved if one has a single route with multiple attributes, as opposed to multiple routes. One could eliminate the Target VPN attribute by creating more routes (i.e., using more RDs), but the scaling properties would be less favorable.

How does a PE determine which Target VPN attributes to associate with a given route? There are a number of different possible ways. The PE might be configured to associate all routes that lead to a particular site with a particular Target VPN. Or the PE might be configured to associate certain routes leading to a particular site with one Target VPN, and certain with another. Or the CE router, when it distributes these routes to the PE (see section 6), might specify one or more Target VPNs for each route. The latter method shifts the control of the mechanisms used to implement the VPN policies from the SP to the customer. If this method is used, it may still be desirable to have the PE eliminate any Target VPNs that, according to its own configuration, are not allowed, and/or to add in some Target VPNs that according to its own configuration are mandatory.

It might be more accurate, if less suggestive, to call this attribute the "Route Target" attribute instead of the "VPN Target" attribute. It really identifies only a set of sites which will be able to use If two sites of a VPN attach to PEs which are in the same Autonomous System, the PEs can distribute VPN-IPv4 routes to each other by means of an IBGP connection between them. Alternatively, each can have an IBGP connection to a route reflector.
If two sites of VPN are in different Autonomous Systems (e.g., because they are connected to different SPs), then a PE router will need to use IBGP to redistribute VPN-IPv4 routes either to an Autonomous System Border Router (ASBR), or to a route reflector of which an ASBR is a client. The ASBR will then need to use EBGP to redistribute those routes to an ASBR in another AS. This allows one to connect different VPN sites to different Service Providers. However, VPN-IPv4 routes should only be accepted on EBGP connections at private peering points, as part of a trusted arrangement between SPs. VPN-IPv4 routes should neither be distributed to nor accepted from the public Internet.

If there are many VPNs having sites attached to different Autonomous Systems, there does not need to be a single ASBR between those two ASes which holds all the routes for all the VPNs; there can be multiple ASBRs, each of which holds only the routes for a particular subset of the VPNs. When a PE router distributes a VPN-IPv4 route via BGP, it uses its own address as the "BGP next hop". It also assigns and distributes an MPLS label. (Essentially, PE routers distribute not VPN-IPv4 routes, but Labeled VPN-IPv4 routes. Cf. [8]) When the PE processes a received packet that has this label at the top of the stack, the PE will pop the stack, and send the packet directly to the site from to which the route leads. This will usually

mean that it just sends the packet to the CE router from which it learned the route. The label may also determine the data link encapsulation.

In most cases, the label assigned by a PE will cause the packet to be sent directly to a CE, and the PE which receives the labeled packet will not look up the packet's destination address in any forwarding table. However, it is also possible for the PE to assign a label which implicitly identifies a particular forwarding table. In this case, the PE receiving a packet that label would look up the packet's destination address in one of its forwarding tables. While this can be very useful in certain circumstances, we do not consider it further in this paper.

Note that the MPLS label that is distributed in this way is only usable if there is a label switched path between the router that installs a route and the BGP next hop of that route. We do not make any assumption about the procedure used to set up that label switched path. It may be set up on a pre-established basis, or it may be set up when a route which would need it is installed. It may be a "best effort" route, or it may be a traffic engineered route. Between a particular PE router and its BGP next hop for a particular route there may be one LSP, or there may be several, perhaps with different QoS characteristics. All that matters for the VPN architecture is that some label switched path between the router and its BGP next hop exists.

All the usual techniques for using route reflectors [2] to improve scalability, e.g., route reflector hierarchies, are available. If route reflectors are used, there is no need to have any one route reflector know all the VPN-IPv4 routes for all the VPNs supported by the backbone. One can have separate route reflectors, which do not communicate with each other, each of which supports a subset of the total set of VPNs.

If a given PE router is not attached to any of the Target VPNs of a particular route, it should not receive that route; the other PE or route reflector which is distributing routes to it should apply outbound filtering to avoid sending it unnecessary routes. Of course, if a PE router receives a route via BGP, and that PE is not attached to any of the route's target VPNs, the PE should apply inbound filtering to the route, neither installing nor redistributing it.

A router which is not attached to any VPN, i.e., a P router, never installs any VPN-IPv4 routes at all.

These distribution rules ensure that there is no one box which needs to know all the VPN-IPv4 routes that are supported over the backbone. As a result, the total number of such routes that can be supported over the backbone is not bound by the capacity of any single device, and therefore can increase virtually without bound.

4.2.3. The VPN of Origin Attribute

A VPN-IPv4 route may be optionally associated with a VPN of Origin attribute. This attribute uniquely identifies a set of sites, and identifies the corresponding route as having come from one of the sites in that set. Typical uses of this attribute might be to identify the enterprise which owns the site where the route leads, or to identify the site's intranet. However, other uses are also possible. This attribute could be encoded as an extended BGP communities attribute.

In situations in which it is necessary to identify the source of a route, it is this attribute, not the RD, which must be used. This attribute may be used when "constructing" VPNs, as described below.

It might be more accurate, if less suggestive, to call this attribute the "Route Origin" attribute instead of the "VPN of Origin" attribute. It really identifies the route only has having come from one of a particular set of sites, without prejudice as to whether that particular set of sites really constitutes a VPN.

4.2.4. Building VPNs using Target and Origin Attributes

By setting up the Target VPN and VPN of Origin attributes properly, one can construct different kinds of VPNs.

Suppose it is desired to create a Closed User Group (CUG) which contains a particular set of sites. This can be done by creating a particular Target VPN attribute value to represent the CUG. This value then needs to be associated with the per-site forwarding tables for each site in the CUG, and it needs to be associated with

every route learned from a site in the CUG. Any route which has this Target VPN attribute will need to be redistributed so that it reaches every PE router attached to one of the sites in the CUG.

Alternatively, suppose one desired, for whatever reason, to create a "hub and spoke" kind of VPN. This could be done by the use of two Target Attribute values, one meaning "Hub" and one meaning "Spoke". Then routes from the spokes could be distributed to the hub, without causing routes from the hub to be distributed to the spokes.

Suppose one has a number of sites which are in an intranet and an extranet, as well as a number of sites which are in the intranet only. Then there may be both intranet and extranet routes which have a Target VPN identifying the entire set of sites. The sites which are to have intranet routes only can filter out all routes with the "wrong" VPN of Origin.

These two attributes allow great flexibility in allowing one to control the distribution of routing information among various sets of sites, which in turn provides great flexibility in constructing VPNs.

5. Forwarding Across the Backbone

If the intermediate routes in the backbone do not have any information about the routes to the VPNs, how are packets forwarded from one VPN site to another?

This is done by means of MPLS with a two-level label stack.

PE routers (and ASBRs which redistribute VPN-IPv4 addresses) need to insert /32 address prefixes for themselves into the IGP routing tables of the backbone. This enables MPLS, at each node in the backbone network, to assign a label corresponding to the route to each PE router. (Certain procedures for setting up label switched paths in the backbone may not require the presence of the /32 address prefixes.)

When a PE receives a packet from a CE device, it chooses a particular per-site forwarding table in which to look up the packet's destination address. Assume that a match is found.

If the packet is destined for a CE device attached to this same PE, the packet is sent directly to that CE device. If the packet is not destined for a CE device attached to this same PE, the packet's "BGP Next Hop" is found, as well as the label which that BGP next hop assigned for the packet's destination address. This label is pushed onto the packet's label stack, and becomes the bottom label. Then the PE looks up the IGP route to the BGP Next Hop, and thus determines the IGP next hop, as well as the label assigned to the address of the BGP next hop by the IGP next hop. This label gets pushed on as the packet's top label, and the packet is then forwarded to the IGP next hop. (If the BGP next hop is the same as the IGP next hop, the second label may not need to be pushed on, however.)

At this point, MPLS will carry the packet across the backbone and into the appropriate CE device. That is, all forwarding decisions by P routers and PE routers are now made by means of MPLS, and the packet's IP header is not looked at again until the packet reaches the CE device. The final PE router will pop the last label from the MPLS label stack before sending the packet to the CE device, thus the CE device will just see an ordinary IP packet. (Though see section 8 for some discussion of the case where the CE desires to received labeled packets.)

When a packet enters the backbone from a particular site via a particular PE router, the packet's route is determined by the contents of the forwarding table which that PE router associated with that site. The forwarding tables of the PE router where the packet leaves the backbone are not relevant. As a result, one may have multiple routes to the same system, where the particular route chosen for a particular packet is based on the site from which the packet enters the backbone.

Note that it is the two-level labeling that makes it possible to keep all the VPN routes out of the P routers, and this in turn is crucial to ensuring the scalability of the model. The backbone does not even need to have routes to the CEs, only to the PEs.

6. How PEs Learn Routes from CEs

The PE routers which attach to a particular VPN need to know, for each of that VPN's sites, which addresses in that VPN are at each site. In the case where the CE device is a host or a switch, this set of addresses will generally be configured into the PE router attaching to that device. In the case where the CE device is a router, there are a number of possible ways that a PE router can obtain this set of addresses.

The PE translates these addresses into VPN-IPv4 addresses, using a configured RD. The PE then treats these VPN-IPv4 routes as input to BGP. In no case will routes from a site ever be leaked into the backbone's IGP.

Exactly which PE/CE route distribution techniques are possible depends on whether a particular CE is in a "transit VPN" or not. A "transit VPN" is one which contains a router that receives routes from a "third party" (i.e., from a router which is not in the VPN, but is not a PE router), and that redistributes those routes to a PE router. A VPN which is not a transit VPN is a "stub VPN". The vast majority of VPNs, including just about all corporate enterprise networks, would be expected to be "stubs" in this sense.

The possible PE/CE distribution techniques are:

Static routing (i.e., configuration) may be used. (This is likely to be useful only in stub VPNs.)

PE and CE routers may be RIP peers, and the CE may use RIP to tell the PE router the set of address prefixes which are reachable at the CE router's site. When RIP is configured in the CE, care must be taken to ensure that address prefixes from other sites (i.e., address prefixes learned by the CE router from the PE router) are never advertised to the PE. More precisely: if a PE router, say PE1, receives a VPN-IPv4 route R1, and as a result distributes an IPv4 route R2 to a CE, then R2 must not be distributed back from that CE's site to a PE router, say PE2, (where PE1 and PE2 may be the same router or different routers), unless PE2 maps R2 to a VPN-IPv4 route which is different than (i.e., contains a different RD than) R1.

The PE and CE routers may be OSPF peers. In this case, the site should be a single OSPF area, the CE should be an ABR in that area, and the PE should be an ABR which is not in that area. Also, the PE should report no router links other than those to the CEs which are at the same site. (This technique should be used only in stub VPNs.)

The PE and CE routers may be BGP peers, and the CE router may use BGP (in particular, EBGP to tell the PE router the set of address prefixes which are at the CE router's site. (This technique can be used in stub VPNs or transit VPNs.)

From a purely technical perspective, this is by far the best technique:

a) Unlike the IGP alternatives, this does not require the PE to run multiple routing algorithm instances in order to talk to multiple CEs

b) BGP is explicitly designed for just this function: passing routing information between systems run by different administrations

c) If the site contains "BGP backdoors", i.e., routers with BGP connections to routers other than PE routers, this procedure will work correctly in all circumstances. The other procedures may or may not work, depending on the precise circumstances.

d) Use of BGP makes it easy for the CE to pass attributes of the routes to the PE. For example, the CE may suggest a particular Target for each route, from among the Target attributes that the PE is authorized to attach to the route.

On the other hand, using BGP is likely to be something new for the CE administrators, except in the case where the customer itself is already an Internet Service Provider (ISP).

If a site is not in a transit VPN, note that it need not have a unique Autonomous System Number (ASN). Every CE whose site which is not in a transit VPN can use the same ASN. This can be chosen from the private ASN space, and it will be stripped out by the PE. Routing loops are prevented by use of the Site of Origin Attribute (see below).

If a set of sites constitute a transit VPN, it is convenient to represent them as a BGP Confederation, so that the internal structure of the VPN is hidden from any router which is not within the VPN. In this case, each site in the VPN would need two BGP connections to the backbone, one which is internal to the confederation and one which is external to it. The usual intra-confederation procedures would have to be slightly modified in order to take account for the fact that the backbone and the sites may have different policies. The backbone is a member of the confederation on one of the connections, but is not a member on the other. These techniques may be useful if the customer for the VPN service is an ISP. This technique allows a customer that is an ISP to obtain VPN backbone service from one of its ISP peers.

(However, if a VPN customer is itself an ISP, and its CE routers support MPLS, a much simpler technique can be used, wherein the ISP is regarded as a stub VPN. See section 8.)

When we do not need to distinguish among the different ways in which a PE can be informed of the address prefixes which exist at a given site, we will simply say that the PE has "learned" the routes from that site.

Before a PE can redistribute a VPN-IPv4 route learned from a site, it must assign certain attributes to the route. There are three such attributes:

- Site of Origin

This attribute uniquely identifies the site from which the PE router learned the route. All routes learned from a particular site must be assigned the same Site of Origin attribute, even if a site is multiply connected to a single PE, or is connected to multiple PEs. Distinct Site of Origin attributes must be used for distinct sites. This attribute could be encoded as an extended BGP communities attribute (section 4.2.1).

- VPN of Origin

- Target VPN

7. How CEs learn Routes from PEs

In this section, we assume that the CE device is a router.

In general, a PE may distribute to a CE any route which the PE has placed in the forwarding table which it uses to route packets from that CE. There is one exception: if a route's Site of Origin attribute identifies a particular site, that route must never be redistributed to any CE at that site.

In most cases, however, it will be sufficient for the PE to simply distribute the default route to the CE. (In some cases, it may even be sufficient for the CE to be configured with a default route pointing to the PE.) This will generally work at any site which does not itself need to distribute the default route to other sites. (E.g., if one site in a corporate VPN has the corporation's access to the Internet, that site might need to have default distributed to the other site, but one could not distribute default to that site itself.)

Whatever procedure is used to distribute routes from CE to PE will also be used to distribute routes from PE to CE.

8. What if the CE Supports MPLS?

In the case where the CE supports MPLS, AND is willing to import the complete set of routes from its VPNs, the PE can distribute to it a label for each such route. When the PE receives a packet from the CE with such a label, it (a) replaces that label with the corresponding label that it learned via BGP, and (b) pushes on a label corresponding to the BGP next hop for the corresponding route.

8.1. Virtual Sites

If the CE/PE route distribution is done via BGP, the CE can use MPLS to support multiple virtual sites. The CE may itself contain a separate forwarding table for each virtual site, which it populates as indicated by the VPN of Origin and Target VPN attributes of the routes it receives from the PE. If the CE receives the full set of routes from the PE, the PE will not need to do any address lookup at all on packets received from the CE. Alternatively, the PE may in some cases be able to distribute to the CE a single (labeled) default route for each VPN. Then when the PE receives a labeled packet from the CE, it would know which forwarding table to look in; the label placed on the packet by the CE would identify only the virtual site from which the packet is coming.

8.2. Representing an ISP VPN as a Stub VPN

If a particular VPN is actually an ISP, but its CE routers support MPLS, then the VPN can actually be treated as a stub VPN. The CE and PE routers need only exchange routes which are internal to the VPN.
The PE router would distribute to the CE router a label for each of these routes. Routers at different sites in the VPN can then become BGP peers. When the CE router looks up a packet's destination address, the routing lookup always resolves to an internal address, usually the address of the packet's BGP next hop. The CE labels the packet appropriately and sends the packet to the PE.

9. Security

Under the following conditions:

a) labeled packets are not accepted by backbone routers from untrusted or unreliable sources, unless it is known that such packets will leave the backbone before the IP header or any labels lower in the stack will be inspected, and

b) labeled VPN-IPv4 routes are not accepted from untrusted or unreliable sources, the security provided by this architecture is virtually identical to that provided to VPNs by Frame Relay or ATM backbones.

It is worth noting that the use of MPLS makes it much simpler to provide this level of security than would be possible if one attempted to use some form of IP-within-IP tunneling in place of MPLS. It is a simple matter to refuse to accept a labeled packet unless the first of the above conditions applies to it. It is rather more difficult to configure the a router to refuse to accept an IP packet if that packet is an IP-within-IP tunnelled packet which is going to a "wrong" place.

The use of MPLS also allows a VPN to span multiple SPs without depending in any way on the inter-domain distribution of IPv4 routing information.

It is also possible for a VPN user to provide himself with enhanced security by making use of Tunnel Mode IPSEC [5]. This is discussed in the remainder of this section.

9.1. Point-to-Point Security Tunnels between CE Routers

A security-conscious VPN user might want to ensure that some or all of the packets which traverse the backbone are authenticated and/or encrypted. The standard way to obtain this functionality today would be to create a "security tunnel" between every pair of CE routers in a VPN, using IPSEC Tunnel Mode.

However, the procedures described so far do not enable the CE router transmitting a packet to determine the identify of the next CE router that the packet will traverse. Yet that information is required in order to use Tunnel Mode IPSEC. So we must extend those procedures to make this information available.

A way to do this is suggested in [6]. Every VPN-IPv4 route can have an attribute which identifies the next CE router that will be traversed if that route is followed. If this information is provided to all the CE routers in the VPN, standard IPSEC Tunnel Mode can be used.

If the CE and PE are BGP peers, it is natural to present this information as a BGP attribute.

Each CE that is to use IPSEC should also be configured with a set of address prefixes, such that it is prohibited from sending insecure traffic to any of those addresses. This prevents the CE from sending insecure traffic if, for some reason, it fails to obtain the necessary information.

When MPLS is used to carry packets between the two endpoints of an IPSEC tunnel, the IPSEC outer header does not really perform any function. It might be beneficial to develop a form of IPSEC tunnel mode which allows the outer header to be omitted when MPLS is used.

9.2. Multi-Party Security Associations

Instead of setting up a security tunnel between each pair of CE routers, it may be advantageous to set up a single, multiparty security association. In such a security association, all the CE outers which are in a particular VPN would share the same security parameters (.e.g., same secret, same algorithm, etc.). Then the ingress CE wouldn't have to know which CE is the next one to receive the data, it would only have to know which VPN the data is going to. A CE which is in multiple VPNs could use different security parameters for each one, thus protecting, e.g., intranet packets from being exposed to the extranet.

With such a scheme, standard Tunnel Mode IPSEC could not be used, because there is no way to fill in the IP destination address field of the "outer header". However, when MPLS is used for forwarding, there is no real need for this outer header anyway; the PE router can use MPLS to get a packet to a tunnel endpoint without even knowing the IP address of that endpoint; it only needs to see the IP destination address of the "inner header".

A significant advantage of a scheme like this is that it makes routing changes (in particular, a change of egress CE for a particular address prefix) transparent to the security mechanism. This could be particularly important in the case of multi-provider VPNs, where the need to distribute information about such routing changes simply to support the security mechanisms could result in scalability issues. Another advantage is that it eliminates the need for the outer IP header, since the MPLS encapsulation performs its role.

10. Quality of Service

Although not the focus of this paper, Quality of Service is a key component of any VPN service. In MPLS/BGP VPNs, existing L3 QoS capabilities can be applied to labeled packets through the use of the "experimental" bits in the shim header [10], or, where ATM is used as the backbone, through the use of ATM QoS capabilities. The traffic engineering work discussed in [1] is also directly applicable to MPLS/BGP VPNs. Traffic engineering could even be used to establish LSPs with particular QoS characteristics between particular pairs of sites, if that is desirable. Where an MPLS/BGP VPN spans multiple SPs, the architecture described in [7] may be useful. An SP may apply either intserv or diffserv capabilities to a particular VPN, as appropriate.

11. Scalability

We have discussed scalability issues throughout this paper. In this section, we briefly summarize the main characteristics of our model with respect to scalability.

The Service Provider backbone network consists of (a) PE routers, (b) BGP Route Reflectors, (c) P routers (which are neither PE routers nor Route Reflectors), and, in the case of multi-provider VPNs, (d) ASBRs.

P routers do not maintain any VPN routes. In order to properly forward VPN traffic, the P routers need only maintain routes to the PE routers and the ASBRs. The use of two levels of labeling is what makes it possible to keep the VPN routes out of the P routers. A PE router to maintains VPN routes, but only for those VPNs to which it is directly attached.

Route reflectors and ASBRs can be partitioned among VPNs so that each partition carries routes for only a subset of the VPNs provided by the Service Provider. Thus no single Route Reflector or ASBR is required to maintain routes for all the VPNs. As a result, no single component within the Service Provider network has to maintain all the routes for all the VPNs. So the total capacity of the network to support increasing numbers of VPNs is not limited by the capacity of any individual component.

12. Intellectual Property Considerations

Cisco Systems may seek patent or other intellectual property protection for some of all of the technologies disclosed in this document. If any standards arising from this document are or become protected by one or more patents assigned to Cisco Systems, Cisco intends to disclose those patents and license them on reasonable and non-discriminatory terms.

13. Security Considerations

Security issues are discussed throughout this memo.

14. Acknowledgments

Significant contributions to this work have been made by Ravi Chandra, Dan Tappan and Bob Thomas.

15. Authors' Addresses

Eric C. Rosen
Cisco Systems, Inc.
250 Apollo Drive
Chelmsford, MA, 01824

EMail: erosen@cisco.com

Yakov Rekhter
Cisco Systems, Inc.
170 Tasman Drive
San Jose, CA, 95134

EMail: yakov@cisco.com

16. References

[1] Awduche, Berger, Gan, Li, Swallow, and Srinavasan, "Extensions to RSVP for LSP Tunnels", Work in Progress.

[2] Bates, T. and R. Chandrasekaran, "BGP Route Reflection: An alternative to full mesh IBGP", RFC 1966, June 1996.

[3] Bates, T., Chandra, R., Katz, D. and Y. Rekhter, "Multiprotocol Extensions for BGP4", RFC 2283, February 1998.

[4] Gleeson, Heinanen, and Armitage, "A Framework for IP Based Virtual Private Networks", Work in Progress.

[5] Kent and Atkinson, "Security Architecture for the Internet Protocol", RFC 2401, November 1998.

[6] Li, "CPE based VPNs using MPLS", October 1998, Work in Progress.

[7] Li, T. and Y. Rekhter, "A Provider Architecture for Differentiated Services and Traffic Engineering (PASTE)", RFC2430, October 1998.

[8] Rekhter and Rosen, "Carrying Label Information in BGP4", Work in Progress.

[9] Rosen, Viswanathan, and Callon, "Multiprotocol Label Switching Architecture", Work in Progress.

[10] Rosen, Rekhter, Tappan, Farinacci, Fedorkow, Li, and Conta, "MPLS Label Stack Encoding", Work in Progress.

17. Full Copyright Statement

RFC 2048

Network Working Group N. Freed
Request for Comments: 2048 Innosoft
BCP: 13 J. Klensin
Obsoletes: 1521, 1522, 1590 MCI
Category: Best Current Practice J. Postel
ISI
November 1996

Multipurpose Internet Mail Extensions (MIME) Part Four: Registration Procedures

Status of this Memo

This document specifies an Internet Best Current Practices for the Internet Community, and requests discussion and suggestions for improvements. Distribution of this memo is unlimited.

Abstract

STD 11, RFC 822, defines a message representation protocol specifying considerable detail about US-SCII message headers, and leaves the message content, or message body, as flat US-ASCII text. This set of documents, collectively called the Multipurpose Internet Mail Extensions, or MIME, redefines the format of messages to allow for

These documents are based on earlier work documented in RFC 934, STD 11, and RFC 1049, but extends and revises them. Because RFC 822 said so little about message bodies, these documents are largely orthogonal to (rather than a revision of) RFC 822.

This fourth document, RFC 2048, specifies various IANA registration procedures for the following MIME facilities:

(1) media types,

(2) external body access types,

(3) content-transfer-encodings.

Registration of character sets for use in MIME is covered elsewhere and is no longer addressed by this document.

These documents are revisions of RFCs 1521 and 1522, which themselves were revisions of RFCs 1341 and 1342. An appendix in RFC 2049 describes differences and changes from previous versions.

Table of Contents

1. Introduction

Recent Internet protocols have been carefully designed to be easily extensible in certain areas. In particular, MIME [RFC 2045] is an open-ended framework and can accommodate additional object types, character sets, and access methods without any changes to the basic protocol. A registration process is needed, however, to ensure that the set of such values is developed in an orderly, well-specified, and public manner.

This document defines registration procedures which use the Internet Assigned Numbers Authority (IANA) as a central registry for such values.

Historical Note: The registration process for media types was initially defined in the context of the asynchronous Internet mail environment. In this mail environment there is a need to limit the number of possible media types to increase the likelihood of interoperability when the capabilities of the remote mail system are not known. As media types are used in new environments, where the proliferation of media types

is not a hindrance to interoperability, the original procedure was excessively restrictive and had to be generalized.

2. Media Type Registration

Registration of a new media type or types starts with the construction of a registration proposal. registration may occur in several different registration trees, which have different requirements as discussed below. In general, the new registration proposal is circulated and reviewed in a fashion appropriate to the tree involved. The media type is then registered if the proposal is acceptable. The following sections describe the requirements and procedures used for each of the different registration trees.

2.1. Registration Trees and Subtype Names

In order to increase the efficiency and flexibility of the registration process, different structures of subtype names may be registered to accomodate the different natural requirements for, e.g., a subtype that will be recommended for wide support and implementation by the Internet Community or a subtype that is used to move files associated with proprietary software. The following subsections define registration "trees", distinguished by the use of faceted names (e.g., names of the form "tree.subtree...type"). Note that some media types defined prior to this document do not conform to the naming conventions described below. See Appendix A for a discussion of them.

2.1.1. IETF Tree

The IETF tree is intended for types of general interest to the Internet Community. Registration in the IETF tree requires approval by the IESG and publication of the media type registration as some form of RFC.

Media types in the IETF tree are normally denoted by names that are not explicitly faceted, i.e., do not contain period (".", full stop) characters.

The "owner" of a media type registration in the IETF tree is assumed to be the IETF itself. Modification or alteration of the specification requires the same level of processing (e.g. standards track) required for the initial registration.

2.1.2. Vendor Tree

The vendor tree is used for media types associated with commercially available products. "Vendor" or "producer" are construed as equivalent and very broadly in this context. A registration may be placed in the vendor tree by anyone who has need to interchange files associated with the particular product. However, the registration formally belongs to the vendor or organization producing the software or file format. Changes to the specification will be made at their request, as discussed in subsequent sections.

Registrations in the vendor tree will be distinguished by the leading facet "vnd.". That may be followed, at the discretion of the registration, by either a media type name from a well-known producer (e.g., vnd.mudpie") or by an IANA-approved designation of the producer's name which is then followed by a media type or product designation (e.g., vnd.bigcompany.funnypictures).

While public exposure and review of media types to be registered in the vendor tree is not required, using the ietf-types list for review is strongly encouraged to improve the quality of those specifications. Registrations in the vendor tree may be submitted directly to the IANA.

2.1.3. Personal or Vanity Tree

Registrations for media types created experimentally or as part of products that are not distributed commercially may be registered in the personal or vanity tree. The registrations are distinguished by the leading facet "prs.". The owner of "personal" registrations and associated specifications is the person or entity making the registration, or one to whom responsibility has been transferred as described below.

While public exposure and review of media types to be registered in the personal tree is not required, using the ietf-types list for review is strongly encouraged to improve the quality of those specifications. Registrations in the personl tree may be submitted directly to the IANA.

2.1.4. Special `x.' Tree

For convenience and symmetry with this registration scheme, media type names with "x." as the first facet may be used for the same purposes for which names starting in "x-" are normally used. These types are unregistered, experimental, and should be used only with the active agreement of the parties exchanging them.

However, with the simplified registration procedures described above for vendor and personal trees, it should rarely, if ever, be necessary to use unregistered experimental types, and as such use of both "x-" and "x." forms is discouraged.

2.1.5. Additional Registration Trees

From time to time and as required by the community, the IANA may, with the advice and consent of the IESG, create new top-level registration trees. It is explicitly assumed that these trees may be created for external registration and management by well-known permanent bodies, such as scientific societies for media types specific to the sciences they cover. In general, the quality of review of specifications for one of these additional registration trees is expected to be equivalent to that which IETF would give to registrations in its own tree. Establishment of these new trees will be announced through RFC publication approved by the IESG.

2.2. Registration Requirements

Media type registration proposals are all expected to conform to various requirements laid out in the following sections. Note that requirement specifics sometimes vary depending on the registration tree, again as detailed in the following sections.

2.2.1. Functionality Requirement

Media types must function as an actual media format: Registration of things that are better thought of as a transfer encoding, as a character set, or as a collection of separate entities of another type, is not allowed. For example, although applications exist to decode the base64 transfer encoding [RFC 2045], base64 cannot be registered as a media type.

This requirement applies regardless of the registration tree involved.

2.2.2. Naming Requirements

All registered media types must be assigned MIME type and subtype names. The combination of these names then serves to uniquely identify the media type and the format of the subtype name identifies the registration tree.

The choice of top-level type name must take the nature of media type involved into account. For example, media normally used for representing still images should be a subtype of the image content type, whereas media capable of representing audio information belongs under the audio content type. See RFC 2046 for additional information on the basic set of top-level types and their characteristics.

New subtypes of top-level types must conform to the restrictions of the top-level type, if any. For example, all subtypes of the multipart content type must use the same encapsulation syntax.

In some cases a new media type may not "fit" under any currently defined top-level content type. Such cases are expected to be quite rare. However, if such a case arises a new top-level type can be defined to accommodate it. Such a definition must be done via standards-track RFC; no other mechanism can be used to define additional top-level content types.

These requirements apply regardless of the registration tree involved.

2.2.3. Parameter Requirements

Media types may elect to use one or more MIME content type parameters, or some parameters may be automatically made available to the media type by virtue of being a subtype of a content type that defines a set of parameters applicable to any of its subtypes. In either case, the names, values, and meanings of any parameters must be fully specified when a media type is registered in the IETF tree, and should be specified as completely as possible when media types are registered in the vendor or personal trees.

New parameters must not be defined as a way to introduce new functionality in types registered in the IETF tree, although new parameters may be added to convey additional information that does not otherwise change existing functionality. An example of this would be a "revision" parameter to indicate a revision level of an external specification such as JPEG. Similar behavior is encouraged for media types registered in the vendor or personal trees but is not required.

2.2.4. Canonicalization and Format Requirements

All registered media types must employ a single, canonical data format, regardless of registration tree.

A precise and openly available specification of the format of each media type is required for all types registered in the IETF tree and must at a minimum be referenced by, if it isn't actually included in, the media type registration proposal itself.

The specifications of format and processing particulars may or may not be publically available for media types registered in the vendor tree, and such registration proposals are explicitly permitted to include only a specification of which software and version produce or process such media types. References to or inclusion of format specifications in registration proposals is encouraged but not required.

Format specifications are still required for registration in the personal tree, but may be either published as RFCs or otherwise deposited with IANA. The deposited specifications will meet the same criteria as those required to register a well-known TCP port and, in particular, need not be made public.

Some media types involve the use of patented technology. The registration of media types involving patented technology is specifically permitted. However, the restrictions set forth in RFC 1602 on the use of patented technology in standards-track protocols must be respected when the specification of a media type is part of a standards-track protocol.

2.2.5. Interchange Recommendations

Media types should, whenever possible, interoperate across as many systems and applications as possible. However, some media types will inevitably have problems interoperating across different platforms. Problems with different versions, byte ordering, and specifics of gateway handling can and will arise.

Universal interoperability of media types is not required, but known interoperability issues should be identified whenever possible. Publication of a media type does not require an exhaustive review of interoperability, and the interoperability considerations section is subject to continuing evaluation.

These recommendations apply regardless of the registration tree involved.

2.2.6. Security Requirements

An analysis of security issues is required for for all types registered in the IETF Tree. (This is in accordance with the basic requirements for all IETF protocols.) A similar analysis for media types registered in the vendor or personal trees is encouraged but not required. However, regardless of what security analysis has or has not been done, all descriptions of security issues must be as accurate as possible regardless of registration tree. In particular, a statement that there are "no security issues associated with this type" must not be confused with "the security issues associates with this type have not been assessed".

There is absolutely no requirement that media types registered in any tree be secure or completely free from risks. Nevertheless, all known security risks must be identified in the registration of a media type, again regardless of registration tree.

The security considerations section of all registrations is subject to continuing evaluation and modification, and in particular may be extended by use of the "comments on media types" mechanism described in subsequent sections.

Some of the issues that should be looked at in a security analysis of a media type are:

Complex media types may include provisions for directives that institute actions on a recipient's files or other resources. In many cases provision is made for originators to specify arbitrary actions in an unrestricted fashion which may then have devastating effects. See the registration of the application/postscript media type in RFC 2046 for an example of such directives and how to handle them.

Complex media types may include provisions for directives that institute actions which, while not directly harmful to the recipient, may result in disclosure of information that either facilitates a subsequent attack or else violates a recipient's privacy in some way. Again, the registration of the application/postscript media type illustrates how such directives can be handled.

A media type might be targeted for applications that require some sort of security assurance but not provide the necessary security mechanisms themselves. For example, a media type could be defined for storage of confidential medical information which in turn requires an external confidentiality service.

2.2.7. Usage and Implementation Non-requirements

In the asynchronous mail environment, where information on the capabilities of the remote mail agent is frequently not available to the sender, maximum interoperability is attained by restricting the number of media types used to those "common" formats expected to be widely implemented. This was asserted in the past as a reason to limit the number of possible media types and resulted in a registration process with a significant hurdle and delay for those registering media types.

However, the need for "common" media types does not require limiting the registration of new media types. If a limited set of media types is recommended for a particular application, that should be asserted by a separate applicability statement specific for the application and/or environment.

As such, universal support and implementation of a media type is NOT a requirement for registration. If, however, a media type is explicitly intended for limited use, this should be noted in its registration.

2.2.8. Publication Requirements

Proposals for media types registered in the IETF tree must be published as RFCs. RFC publication of vendor and personal media type proposals is encouraged but not required. In all cases IANA will retain copies of all media type proposals and "publish" them as part of the media types registration tree itself.

Other than in the IETF tree, the registration of a data type does not imply endorsement, approval, or recommendation by IANA or IETF or even certification that the specification is adequate. To become Internet Standards, protocol, data objects, or whatever must go through the IETF standards process. This is too difficult and too lengthy a process for the convenient registration of media types.

The IETF tree exists for media types that do require require a substantive review and approval process with the vendor and personal trees exist for those that do not. It is expected that applicability statements for particular applications will be published from time to time that recommend implementation of, and support for, media types that have proven particularly useful in those contexts.

As discussed above, registration of a top-level type requires standards-track processing and, hence, RFC publication.

2.2.9. Additional Information

Various sorts of optional information may be included in the specification of a media type if it is available:

Magic number(s) (length, octet values). Magic numbers are byte sequences that are always present and thus can be used to identify entities as being of a given media type.

File extension(s) commonly used on one or more platforms to indicate that some file containing a given type of media.

Macintosh File Type code(s) (4 octets) used to label files containing a given type of media.

Such information is often quite useful to implementors and if available should be provided.

2.3. Registration Procedure

The following procedure has been implemented by the IANA for review and approval of new media types. This is not a formal standards process, but rather an administrative procedure intended to allow community comment and sanity checking without excessive time delay. For registration in the IETF tree, the normal IETF processes should be followed, treating posting of an internet-draft and announcement on the ietf-types list (as described in the next subsection) as a first step. For registrations in the vendor or personal tree, the initial review step described below may be omitted and the type registered directly by submitting the template and an explanation directly to IANA (at iana@iana.org). However, authors of vendor or personal media type specifications are encouraged to seek community review and comment whenever that is feasible.

2.3.1. Present the Media Type to the Community for Review

Send a proposed media type registration to the `ietf-types@iana.org` mailing list for a two week review period. This mailing list has been established for the purpose of reviewing proposed media and access types. Proposed media types are not formally registered and must not be used; the "x-" prefix specified in RFC 2045 can be used until registration is complete.

The intent of the public posting is to solicit comments and feedback on the choice of type/subtype name, the unambiguity of the references with respect to versions and external profiling information, and a review of any interoperability or security considerations. The submitter may submit a revised registration, or withdraw the registration completely, at any time.

2.3.2. IESG Approval

Media types registered in the IETF tree must be submitted to the IESG for approval.

2.3.3. IANA Registration

Provided that the media type meets the requirements for media types and has obtained approval that is necessary, the author may submit the registration request to the IANA, which will register the media type and make the media type registration available to the community.

2.4. Comments on Media Type Registrations

Comments on registered media types may be submitted by members of the community to IANA. These comments will be passed on to the "owner" of the media type if possible. Submitters of comments may request that their comment be attached to the media type registration itself, and if IANA approves of this the comment will be made accessible in conjunction with the type registration itself.

2.5. Location of Registered Media Type List

Media type registrations will be posted in the anonymous FTP directory "ftp://ftp.isi.edu/in-notes/iana/assignments/media-types/" and all registered media types will be listed in the periodically issued "Assigned Numbers" RFC [currently STD 2, RFC 1700]. The media type description and other supporting

material may also be published as an Informational RFC by sending it to "rfc-editor@isi.edu" (please follow the instructions to RFC authors [RFC-1543]).

2.6. IANA Procedures for Registering Media Types

The IANA will only register media types in the IETF tree in response to a communication from the IESG stating that a given registration has been approved. Vendor and personal types will be registered by the IANA automatically and without any formal review as long as the following minimal conditions are met:

Media types must function as an actual media format In particular, character sets and transfer encodings may not be registered as media types.

All media types must have properly formed type and subtype names. All type names must be defined by a standards-track RFC. All subtype names must be unique, must conform to the MIME grammar for such names, and must contain the proper tree prefix.

Types registered in the personal tree must either provide a format specification or a pointer to one.

Any security considerations given must not be obviously bogus. (It is neither possible nor necessary for the IANA to conduct a comprehensive security review of media type registrations. Nevertheless, IANA has the authority to identify obviously incompetent material and exclude it.)

2.7. Change Control

Once a media type has been published by IANA, the author may request a change to its definition. The descriptions of the different registration trees above designate the "owners" of each type of registration. The change request follows the same procedure as the registration request:

(1) Publish the revised template on the ietf-types list.

(2) Leave at least two weeks for comments.

(3) Publish using IANA after formal review if required.

Changes should be requested only when there are serious omission or errors in the published specification. When review is required, a change request may be denied if it renders entities that were valid under the previous definition invalid under the new definition.

The owner of a content type may pass responsibility for the content type to another person or agency by informing IANA and the ietf-types list; this can be done without discussion or review.

The IESG may reassign responsibility for a media type. The most common case of this will be to enable changes to be made to types where the author of the registration has died, moved out of contact or is otherwise unable to make changes that are important to the community.

Media type registrations may not be deleted; media types which are no longer believed appropriate for use can be declared OBSOLETE by a change to their "intended use" field; such media types will be clearly marked in the lists published by IANA.

2.8. Registration Template

To: ietf-types@iana.org
Subject: Registration of MIME media type XXX/YYY

MIME media type name:

MIME subtype name:

Required parameters:

Optional parameters:

Encoding considerations:

Security considerations:

Interoperability considerations:

Published specification:

Applications which use this media type:

Additional information:

Magic number(s):
File extension(s):
Macintosh File Type Code(s):

Person & email address to contact for further information:

Intended usage:

(One of COMMON, LIMITED USE or OBSOLETE)

Author/Change controller:

(Any other information that the author deems interesting may be added below this line.)

3. External Body Access Types

RFC 2046 defines the message/external-body media type, whereby a MIME entity can act as pointer to the actual body data in lieu of including the data directly in the entity body. Each message/external-body reference specifies an access type, which determines the mechanism used to retrieve the actual body data. RFC 2046 defines an initial set of access types, but allows for the registration of additional access types to accommodate new retrieval mechanisms.

3.1. Registration Requirements

New access type specifications must conform to a number of requirements as described below.

3.1.1. Naming Requirements

Each access type must have a unique name. This name appears in the access-type parameter in the message/external-body content-type header field, and must conform to MIME content type parameter syntax.

3.1.2. Mechanism Specification Requirements

All of the protocols, transports, and procedures used by a given access type must be described, either in the specification of the access type itself or in some other publicly available specification, in sufficient detail for the access type to be implemented by any competent implementor. Use of secret and/or proprietary methods in access types are expressly prohibited. The restrictions imposed by RFC 1602 on the standardization of patented algorithms must be respected as well.

3.1.3. Publication Requirements

All access types must be described by an RFC. The RFC may be informational rather than standards-track, although standard-track review and approval are encouraged for all access types.

3.1.4. Security Requirements

Any known security issues that arise from the use of the access type must be completely and fully described. It is not required that the access type be secure or that it be free from risks, but that the known risks be identified. Publication of a new access type does not require an exhaustive security review, and the security considerations section is subject to continuing evaluation. Additional security considerations should be addressed by publishing revised versions of the access type specification.

3.2. Registration Procedure

Registration of a new access type starts with the construction of a draft of an RFC.

3.2.1. Present the Access Type to the Community

Send a proposed access type specification to the "ietf- types@iana.org" mailing list for a two week review period. This mailing list has been established for the purpose of reviewing proposed access and media types. Proposed access types are not formally registered and must not be used.

The intent of the public posting is to solicit comments and feedback on the access type specification and a review of any security considerations.

3.2.2. Access Type Reviewer

When the two week period has passed, the access type reviewer, who is appointed by the IETF Applications Area Director, either forwards the request to iana@isi.edu, or rejects it because of significant objections raised on the list.

Decisions made by the reviewer must be posted to the ietf-types mailing list within 14 days. Decisions made by the reviewer may be appealed to the IESG.

3.2.3. IANA Registration

Provided that the access type has either passed review or has been successfully appealed to the IESG, the IANA will register the access type and make the registration available to the community. The specification of the access type must also be published as an RFC. Informational RFCs are published by sending them to "rfc- editor@isi.edu" (please follow the instructions to RFC authors [RFC- 1543]).

3.3. Location of Registered Access Type List

Access type registrations will be posted in the anonymous FTP directory "ftp://ftp.isi.edu/in-notes/iana/assignments/access-types/" and all registered access types will be listed in the periodically issued "Assigned Numbers" RFC [currently RFC-1700].

3.4. IANA Procedures for Registering Access Types

The identity of the access type reviewer is communicated to the IANA by the IESG. The IANA then only acts in response to access type definitions that either are approved by the access type reviewer and forwarded by the reviewer to the IANA for registration, or in response to a communication from the IESG that an access type definition appeal has overturned the access type reviewer's ruling.

4. Transfer Encodings

Transfer encodings are tranformations applied to MIME media types after conversion to the media type's canonical form. Transfer encodings are used for several purposes:

(1) Many transports, especially message transports, can only handle data consisting of relatively short lines of text. There can also be severe restrictions on what characters can be used in these lines of text – some transports are restricted to a small subset of US-ASCII and others cannot handle certain character sequences.

Transfer encodings are used to transform binary data into textual form that can survive such transports.

Examples of this sort of transfer encoding include the base64 and quoted-printable transfer encodings defined in RFC 2045.

(2) Image, audio, video, and even application entities are sometimes quite large. Compression algorithms are often quite effective in reducing the size of large entities. Transfer encodings can be used to apply general-purpose non-lossy compression algorithms to MIME entities.

(3) Transport encodings can be defined as a means of representing existing encoding formats in a MIME context.

IMPORTANT: The standardization of a large numbers of different transfer encodings is seen as a significant barrier to widespread interoperability and is expressely discouraged. Nevertheless, the following procedure has been defined to provide a means of defining additional transfer encodings, should standardization actually be justified.

4.1. Transfer Encoding Requirements

Transfer encoding specifications must conform to a number of requirements as described below.

4.1.1. Naming Requirements

Each transfer encoding must have a unique name. This name appears in the Content-Transfer-Encoding header field and must conform to the syntax of that field.

4.1.2. Algorithm Specification Requirements

All of the algorithms used in a transfer encoding (e.g. conversion to printable form, compression) must be described in their entirety in the transfer encoding specification. Use of secret and/or proprietary algorithms in standardized transfer encodings are expressly prohibited. The restrictions imposed by RFC 1602 on the standardization of patented algorithms must be respected as well.

4.1.3. Input Domain Requirements

All transfer encodings must be applicable to an arbitrary sequence of octets of any length. Dependence on particular input forms is not allowed.

It should be noted that the 7bit and 8bit encodings do not conform to this requirement. Aside from the undesireability of having specialized encodings, the intent here is to forbid the addition of additional encodings along the lines of 7bit and 8bit.

4.1.4. Output Range Requirements

There is no requirement that a particular tranfer encoding produce a particular form of encoded output. However, the output format for each transfer encoding must be fully and completely documented. In particular, each specification must clearly state whether the output format always lies within the confines of 7bit data, 8bit data, or is simply pure binary data.

4.1.5. Data Integrity and Generality Requirements

All transfer encodings must be fully invertible on any platform; it must be possible for anyone to recover the original data by performing the corresponding decoding operation. Note that this requirement effectively excludes all forms of lossy compression as well as all forms of encryption from use as a transfer encoding.

4.1.6. New Functionality Requirements

All transfer encodings must provide some sort of new functionality. Some degree of functionality overlap with previously defined transfer encodings is acceptable, but any new transfer encoding must also offer something no other transfer encoding provides.

4.2. Transfer Encoding Definition Procedure

Definition of a new transfer encoding starts with the construction of a draft of a standards-track RFC. The RFC must define the transfer encoding precisely and completely, and must also provide substantial justification for defining and standardizing a new transfer encoding. This specification must then be presented to the IESG for consideration. The IESG can

(1) reject the specification outright as being inappropriate for standardization,

(2) approve the formation of an IETF working group to work on the specification in accordance with IETF procedures, or,

(3) accept the specification as-is and put it directly on the standards track.

Transfer encoding specifications on the standards track follow normal IETF rules for standards track documents. A transfer encoding is considered to be defined and available for use once it is on the standards track.

4.3. IANA Procedures for Transfer Encoding Registration

There is no need for a special procedure for registering Transfer Encodings with the IANA. All legitimate transfer encoding registrations must appear as a standards-track RFC, so it is the IESG's responsibility to notify the IANA when a new transfer encoding has been approved.

4.4. Location of Registered Transfer Encodings List

Transfer encoding registrations will be posted in the anonymous FTP directory "ftp://ftp.isi.edu/in-notes/iana/assignments/transfer- encodings/" and all registered transfer encodings will be listed in the periodically issued "Assigned Numbers" RFC [currently RFC-1700].

5. Authors' Addresses

For more information, the authors of this document are best contacted via Internet mail:

Ned Freed
Innosoft International, Inc.
1050 East Garvey Avenue South
West Covina, CA 91790
USA

Phone: +1 818 919 3600
Fax: +1 818 919 3614
EMail: ned@innosoft.com

John Klensin
MCI
2100 Reston Parkway
Reston, VA 22091

Phone: +1 703 715-7361
Fax: +1 703 715-7436
EMail: klensin@mci.net

RFC 2409

Network Working Group D. Harkins
Request for Comments: 2409 D. Carrel
Category: Standards Track cisco Systems
November 1998

The Internet Key Exchange (IKE)

Status of this Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

Table Of Contents

1. Abstract

ISAKMP ([MSST98]) provides a framework for authentication and key exchange but does not define them. ISAKMP is designed to be key exchange independant; that is, it is designed to support many different key exchanges.

Oakley ([Orm96]) describes a series of key exchanges—called "modes"-- and details the services provided by each (e.g. perfect forward secrecy for keys, identity protection, and authentication).

SKEME ([SKEME]) describes a versatile key exchange technique which provides anonymity, epudiability, and quick key refreshment.

This document describes a protocol using part of Oakley and part of SKEME in conjunction with ISAKMP to obtain authenticated keying material for use with ISAKMP, and for other security associations such as AH and ESP for the IETF IPsec DOI.

2. Discussion

This memo describes a hybrid protocol. The purpose is to negotiate, and provide authenticated keying material for, security associations in a protected manner.

Processes which implement this memo can be used for negotiating virtual private networks (VPNs) and also for providing a remote user from a remote site (whose IP address need not be known beforehand) access to a secure host or network.

Client negotiation is supported. Client mode is where the negotiating parties are not the endpoints for which security association negotiation is taking place. When used in client mode, the identities of the end parties remain hidden.

This does not implement the entire Oakley protocol, but only a subset necessary to satisfy its goals. It does not claim conformance or compliance with the entire Oakley protocol nor is it dependant in any way on the Oakley protocol. Likewise, this does not implement the entire SKEME protocol, but only the method of public key encryption for authentication and its concept of fast re-keying using an exchange of nonces. This protocol is not dependant in any way on the SKEME protocol.

3. Terms and Definitions

3.1 Requirements Terminology

Keywords "MUST", "MUST NOT", "REQUIRED", "SHOULD", "SHOULD NOT" and "MAY" that appear in this document are to be interpreted as described in [Bra97].

3.2 Notation

The following notation is used throughout this memo.

HDR is an ISAKMP header whose exchange type is the mode. When writen as HDR* it indicates payload encryption. SA is an SA negotiation payload with one or more proposals. An initiator MAY provide multiple proposals for negotiation; a responder MUST reply with only one.

<P>_b indicates the body of payload <P>-- the ISAKMP generic vpayload is not included.

SAi_b is the entire body of the SA payload (minus the ISAKMP generic header)-- i.e. the DOI, situation, all proposals and all transforms offered by the Initiator.

CKY-I and CKY-R are the Initiator's cookie and the Responder's cookie, respectively, from the ISAKMP header.

g^xi and g^xr are the Diffie-Hellman ([DH]) public values of the initiator and responder respectively.

g^xy is the Diffie-Hellman shared secret. KE is the key exchange payload which contains the public information exchanged in a Diffie-Hellman exchange. There is no particular encoding (e.g. a TLV) used for the data of a KE payload.

Nx is the nonce payload; x can be: i or r for the ISAKMP initiator and responder respectively.

IDx is the identification payload for "x". x can be: "ii" or "ir" for the ISAKMP initiator and responder respectively during phase one negotiation; or "ui" or "ur" for the user initiator and responder respectively during phase two. The ID payload format for the Internet DOI is defined in [Pip97].

SIG is the signature payload. The data to sign is exchange-specific.

CERT is the certificate payload.

HASH (and any derivitive such as HASH(2) or HASH_I) is the hash payload. The contents of the hash are specific to the authentication method. prf(key, msg) is the keyed pseudo-random function-- often a keyed hash function-- used to generate a deterministic output that appears pseudo-random. prf's are used both for key derivations and for authentication (i.e. as a keyed MAC). (See [KBC96]).

SKEYID is a string derived from secret material known only to the active players in the exchange.

SKEYID_e is the keying material used by the ISAKMP SA to protect the confidentiality of its messages.

SKEYID_a is the keying material used by the ISAKMP SA to authenticate its messages.

SKEYID_d is the keying material used to derive keys for non-ISAKMP security associations.

<x>y indicates that "x" is encrypted with the key "y".

--> signifies "initiator to responder" communication (requests).

<-- signifies "responder to initiator" communication (replies).

| signifies concatenation of information-- e.g. X | Y is the concatentation of X with Y.

[x] indicates that x is optional.

Message encryption (when noted by a '*' after the ISAKMP header) MUST begin immediately after the ISAKMP header. When communication is protected, all payloads following the ISAKMP header MUST be encrypted. Encryption keys are generated from SKEYID_e in a manner that is defined for each algorithm.

3.3 Perfect Forward Secrecy

When used in the memo Perfect Forward Secrecy (PFS) refers to the notion that compromise of a single key will permit access to only data protected by a single key. For PFS to exist the key used to protect transmission of data MUST NOT be used to derive any additional keys, and if the key used to protect transmission of data was derived from some other keying material, that material MUST NOT be used to derive any more keys. Perfect Forward Secrecy for both keys and identities is provided in this protocol. (Sections 5.5 and 8).

3.4 Security Association

A security association (SA) is a set of policy and key(s) used to protect information. The ISAKMP SA is the shared policy and key(s) used by the negotiating peers in this protocol to protect their communication.

4. Introduction

Oakley and SKEME each define a method to establish an authenticated key exchange. This includes payloads construction, the information payloads carry, the order in which they are processed and how they

are used.

While Oakley defines "modes", ISAKMP defines "phases". The relationship between the two is very straightforward and IKE presents different exchanges as modes which operate in one of two phases.

Phase 1 is where the two ISAKMP peers establish a secure, authenticated channel with which to communicate. This is called the ISAKMP Security Association (SA). "Main Mode" and "Aggressive Mode" each accomplish a phase 1 exchange. "Main Mode" and "Aggressive Mode" MUST ONLY be used in phase 1.

Phase 2 is where Security Associations are negotiated on behalf of services such as IPsec or any other service which needs key material and/or parameter negotiation. "Quick Mode" accomplishes a phase 2 exchange. "Quick Mode" MUST ONLY be used in phase 2.

"New Group Mode" is not really a phase 1 or phase 2. It follows phase 1, but serves to establish a new group which can be used in future negotiations. "New Group Mode" MUST ONLY be used after phase 1.

The ISAKMP SA is bi-directional. That is, once established, either party may initiate Quick Mode, Informational, and New Group Mode Exchanges. Per the base ISAKMP document, the ISAKMP SA is identified by the Initiator's cookie followed by the Responder's cookie—the role of each party in the phase 1 exchange dictates which cookie is the Initiator's. The cookie order established by the phase 1 exchange continues to identify the ISAKMP SA regardless of the direction the Quick Mode, Informational, or New Group exchange. In other words, the cookies MUST NOT swap places when the direction of the ISAKMP SA changes.

With the use of ISAKMP phases, an implementation can accomplish very fast keying when necessary. A single phase 1 negotiation may be used for more than one phase 2 negotiation. Additionally a single phase 2 negotiation can request multiple Security Associations. With these optimizations, an implementation can see less than one round trip per SA as well as less than one DH exponentiation per SA. "Main Mode" for phase 1 provides identity protection. When identity protection is not needed, "Aggressive Mode" can be used to reduce round trips even further. Developer hints for doing these optimizations are included below. It should also be noted that using public key encryption to authenticate an Aggressive Mode exchange will still provide identity protection.

This protocol does not define its own DOI per se. The ISAKMP SA, established in phase 1, MAY use the DOI and situation from a non- ISAKMP service (such as the IETF IPSec DOI [Pip97]). In this case an implementation MAY choose to restrict use of the ISAKMP SA for establishment of SAs for services of the same DOI. Alternately, an ISAKMP SA MAY be established with the value zero in both the DOI and situation (see [MSST98] for a description of these fields) and in this case implementations will be free to establish security services for any defined DOI using this ISAKMP SA. If a DOI of zero is used for establishment of a phase 1 SA, the syntax of the identity payloads used in phase 1 is that defined in [MSST98] and not from any DOI-- e.g. [Pip97]-- which may further expand the syntax and semantics of identities.

The following attributes are used by IKE and are negotiated as part of the ISAKMP Secur ity Association. (These attributes pertain only to the ISAKMP Security Association and not to any Security Associations that ISAKMP may be negotiating on behalf of other services.)

- encryption algorithm

- hash algorithm

- authentication method

- information about a group over which to do Diffie-Hellman.

All of these attributes are mandatory and MUST be negotiated. In addition, it is possible to optionally negotiate a psuedo-random function ("prf"). (There are currently no negotiable pseudo-random functions defined in this document. Private use attribute values can be used for prf negotiation between consenting

parties). If a "prf" is not negotiation, the HMAC (see [KBC96]) version of the negotiated hash algorithm is used as a pseudo-random function. Other non- mandatory attributes are described in Appendix A. The selected hash algorithm MUST support both native and HMAC modes.

The Diffie-Hellman group MUST be either specified using a defined group description (section 6) or by defining all attributes of a group (section 5.6). Group attributes (such as group type or prime-- see Appendix A) MUST NOT be offered in conjunction with a previously defined group (either a reserved group scription or a private use description that is established after conclusion of a New Group Mode exchange).

IKE implementations MUST support the following attribute values:

- DES [DES] in CBC mode with a weak, and semi-weak, key check (weak and semi-weak keys are referenced in [Sch96] and listed in Appendix A). The key is derived according to Appendix B.

- MD5 [MD5] and SHA [SHA}.

- Authentication via pre-shared keys.

- MODP over default group number one (see below).

In addition, IKE implementations SHOULD support: 3DES for encryption; Tiger ([TIGER]) for hash; the Digital Signature Standard, RSA [RSA] signatures and authentication with RSA public key encryption; and MODP group number 2. IKE implementations MAY support any additional encryption algorithms defined in Appendix A and MAY support ECP and EC2N groups.

The IKE modes described here MUST be implemented whenever the IETF IPsec DOI [Pip97] is implemented. Other DOIs MAY use the modes described here.

5. Exchanges

There are two basic methods used to establish an authenticated key exchange: Main Mode and Aggressive Mode. Each generates authenticated keying material from an ephemeral Diffie-Hellman exchange. Main mode MUST be implemented; Aggressive Mode SHOULD be implemented. In addition, Quick Mode MUST be implemented as a mechanism to generate fresh keying material and negotiate non-ISAKMP security services. In addition, New Group Mode SHOULD be implemented as a mechanism to define private groups for Diffie-Hellman exchanges. Implementations MUST NOT switch exchange types in the middle of an exchange.

Exchanges conform to standard ISAKMP payload syntax, attribute encoding, timeouts and retransmits of messages, and informational messages-- e.g a notify response is sent when, for example, a proposal is unacceptable, or a signature verification or decryption was unsuccessful, etc.

The SA payload MUST precede all other payloads in a phase 1 exchange. Except where otherwise noted, there are no requirements for ISAKMP payloads in any message to be in any particular order.

The Diffie-Hellman public value passed in a KE payload, in either a phase 1 or phase 2 exchange, MUST be the length of the negotiated Diffie-Hellman group enforced, if necessary, by pre-pending the value with zeros.

The length of nonce payload MUST be between 8 and 256 bytes inclusive.

Main Mode is an instantiation of the ISAKMP Identity Protect Exchange: The first two messages negotiate policy; the next two exchange Diffie-Hellman public values and ancillary data (e.g. nonces) necessary for the exchange; and the last two messages authenticate the Diffie-Hellman Exchange. The authentication method negotiated as part of the initial ISAKMP exchange influences the composition of the payloads but not their purpose. The XCHG for Main Mode is ISAKMP Identity Protect.

Similarly, Aggressive Mode is an instantiation of the ISAKMP Aggressive Exchange. The first two messages negotiate policy, exchange Diffie-Hellman public values and ancillary data necessary for the exchange, and

identities. In addition the second message authenticates the responder. The third message authenticates the initiator and provides a proof of participation in the exchange. The XCHG for Aggressive Mode is ISAKMP Aggressive. The final message MAY NOT be sent under protection of the ISAKMP SA allowing each party t postpone exponentiation, if desired, until negotiation of this exchange is complete. The graphic depictions of Aggressive Mode show the final payload in the clear; it need not be.

Exchanges in IKE are not open ended and have a fixed number of messages. Receipt of a Certificate Request payload MUST NOT extend the number of messages transmitted or expected.

Security Association negotiation is limited with Aggressive Mode. Due to message construction requirements the group in which the Diffie- Hellman exchange is performed cannot be negotiated. In addition, different authentication methods may further constrain attribute negotiation. For example, authentication with public key encryption cannot be negotiated and when using the revised method of public key encryption for authentication the cipher and hash cannot be negotiated. For situations where the rich attribute negotiation capabilities of IKE are required Main Mode may be required.

Quick Mode and New Group Mode have no analog in ISAKMP. The XCHG values for Quick Mode and new Group Mode are defined in Appendix A.

Main Mode, Aggressive Mode, and Quick Mode do security association negotiation. Security Association offers take the form of Tranform Payload(s) encapsulated in Proposal Payload(s) encapsulated in Security Association (SA) payload(s). If multiple offers are being made for phase 1 exchanges (Main Mode and aggressive Mode) they MUST take the form of multiple Transform Payloads for a single Proposal Payload in a single SA payload. To put it another way, for phase 1 exchanges there MUST NOT be multiple Proposal Payloads for a single SA payload and there MUST NOT be multiple SA payloads. This document does not proscribe such behavior on offers in phase 2 exchanges.

There is no limit on the number of offers the initiator may send to the responder but conformant implementations MAY choose to limit the number of offers it will inspect for performance reasons.

During security association negotiation, initiators present offers for potential security associations to responders. Responders MUST NOT modify attributes of any offer, attribute encoding excepted (see Appendix A). If the initiator of an exchange notices that attribute values have changed or attributes have been added or deleted from an offer made, that response MUST be rejected.

Four different authentication methods are allowed with either Main Mode or Aggressive Mode-- digital signature, two forms of authentication with public key encryption, or pre-shared key. The value SKEYID is computed seperately for each authentication method.

For signatures: SKEYID = prf(Ni_b | Nr_b, g^xy)
For public key encryption: SKEYID = prf(hash(Ni_b | Nr_b), CKY-I | CKY-R)
For pre-shared keys: SKEYID = prf(pre-shared-key, Ni_b | Nr_b)

The result of either Main Mode or Aggressive Mode is three groups of authenticated keying material:

SKEYID_d = prf(SKEYID, g^xy | CKY-I | CKY-R | 0)
SKEYID_a = prf(SKEYID, SKEYID_d | g^xy | CKY-I | CKY-R | 1)
SKEYID_e = prf(SKEYID, SKEYID_a | g^xy | CKY-I | CKY-R | 2)

and agreed upon policy to protect further communications. The values of 0, 1, and 2 above are represented by a single octet. The key used for encryption is derived from SKEYID_e in an algorithm-specific manner (see appendix B).

To authenticate either exchange the initiator of the protocol generates HASH_I and the responder generates HASH_R where:

HASH_I = prf(SKEYID, g^xi | g^xr | CKY-I | CKY-R | SAi_b | IDii_b) HASH_R = prf(SKEYID, g^xr | g^xi | CKY-R | CKY-I | SAi_b | IDir_b)

For authentication with digital signatures, HASH_I and HASH_R are signed and verified; for authentication with either public key encryption or pre-shared keys, HASH_I and HASH_R directly authenticate the exchange. The entire ID payload (including ID type, port, and protocol but excluding the generic header) is hashed into both HASH_I and HASH_R.

As mentioned above, the negotiated authentication method influences the content and use of messages for Phase 1 Modes, but not their intent. When using public keys for authentication, the Phase 1 exchange can be accomplished either by using signatures or by using public key encryption (if the algorithm supports it). Following are Phase 1 exchanges with different authentication options.

5.1 IKE Phase 1 Authenticated With Signatures

Using signatures, the ancillary information exchanged during the second roundtrip are nonces; the exchange is authenticated by signing a mutually obtainable hash. Main Mode with signature authentication is described as follows:

```
Initiator                   Responder
-----------                 -----------
HDR, SA                 -->
                    <--   HDR, SA
HDR, KE, Ni              -->
                    <--   HDR, KE, Nr
HDR*, IDii, [ CERT, ] SIG_I -->
                    <--   HDR*, IDir, [ CERT, ] SIG_R
```

Aggressive mode with signatures in conjunction with ISAKMP is described as follows:

```
Initiator                   Responder
-----------                 -----------
HDR, SA, KE, Ni, IDii       -->
                    <--   HDR, SA, KE, Nr, IDir,
                            [ CERT, ] SIG_R
HDR, [ CERT, ] SIG_I        -->
```

In both modes, the signed data, SIG_I or SIG_R, is the result of the negotiated digital signature algorithm applied to HASH_I or HASH_R respectively.

In general the signature will be over HASH_I and HASH_R as above using the negotiated prf, or the HMAC version of the negotiated hash function (if no prf is negotiated). However, this can be overridden for construction of the signature if the signature algorithm is tied to a particular hash algorithm (e.g. DSS is only defined with SHA's 160 bit output). In this case, the signature will be over HASH_I and HASH_R as above, except using the HMAC version of the hash algorithm associated with the signature method. The negotiated prf and hash function would continue to be used for all other prescribed pseudo- random functions.

Since the hash algorithm used is already known there is no need to encode its OID into the signature. In addition, there is no binding between the OIDs used for RSA signatures in PKCS #1 and those used in this document. Therefore, RSA signatures MUST be encoded as a private key encryption in PKCS #1 format and not as a signature in PKCS #1 format (which includes the OID of the hash algorithm). DSS signatures MUST be encoded as r followed by s.

One or more certificate payloads MAY be optionally passed.

5.2 Phase 1 Authenticated With Public Key Encryption

Using public key encryption to authenticate the exchange, the ancillary information exchanged is encrypted nonces. Each party's ability to reconstruct a hash (proving that the other party decrypted the nonce) authenticates the exchange.

In order to perform the public key encryption, the initiator must already have the responder's public key. In the case where the responder has multiple public keys, a hash of the certificate the initiator is using to encrypt the ancillary information is passed as part of the third message. In this way the responder can determine which corresponding private key to use to decrypt the encrypted payloads and identity protection is retained.

In addition to the nonce, the identities of the parties (IDii and IDir) are also encrypted with the other party's public key. If the authentication method is public key encryption, the nonce and identity payloads MUST be encrypted with the public key of the other party. Only the body of the payloads are encrypted, the payload headers are left in the clear.

When using encryption for authentication, Main Mode is defined as follows.

```
Initiator                  Responder
-----------                -----------
HDR, SA                 -->
                   <--  HDR, SA
HDR, KE, [ HASH(1), ]
  <IDii_b>PubKey_r,
    <Ni_b>PubKey_r      -->
                         HDR, KE, <IDir_b>PubKey_i,
                   <--       <Nr_b>PubKey_i
HDR*, HASH_I            -->
                   <--  HDR*, HASH_R
```

Aggressive Mode authenticated with encryption is described as follows:

```
Initiator                  Responder
-----------                -----------
HDR, SA, [ HASH(1),] KE,
  <IDii_b>Pubkey_r,
   <Ni_b>Pubkey_r       -->
                         HDR, SA, KE, <IDir_b>PubKey_i,
                   <--       <Nr_b>PubKey_i, HASH_R
HDR, HASH_I             -->
```

Where HASH(1) is a hash (using the negotiated hash function) of the certificate which the initiator is using to encrypt the nonce and identity.

RSA encryption MUST be encoded in PKCS #1 format. While only the body of the ID and nonce payloads is encrypted, the encrypted data must be preceded by a valid ISAKMP generic header. The payload length is the length of the entire encrypted payload plus header. The PKCS #1 encoding allows for determination of the actual length of the cleartext payload upon decryption.

Using encryption for authentication provides for a plausably deniable exchange. There is no proof (as with a digital signature) that the conversation ever took place since each party can completely reconstruct both sides of the exchange. In addition, security is added to secret generation since an attacker would have to successfully break not only the Diffie-Hellman exchange but also both RSA encryptions. This exchange was motivated by [SKEME].

Note that, unlike other authentication methods, authentication with public key encryption allows for identity protection with Aggressive Mode.

5.3 Phase 1 Authenticated With a Revised Mode of Public Key Encryption

Authentication with Public Key Encryption has significant advantages over authentication with signatures (see section 5.2 above). Unfortunately, this is at the cost of 4 public key operations—two public key encryptions and two private key decryptions. This authentication mode retains the advantages of authentication using public key encryption but does so with half the public key operations.

In this mode, the nonce is still encrypted using the public key of the peer, however the peer's identity (and the certificate if it is sent) is encrypted using the negotiated symmetric encryption algorithm (from the SA payload) with a key derived from the nonce. This solution adds minimal complexity and state yet saves two costly public key operations on each side. In addition, the Key Exchange payload is also encrypted using the same derived key. This provides additional protection against cryptanalysis of the Diffie-Hellman exchange.

As with the public key encryption method of authentication (section 5.2), a HASH payload may be sent to identify a certificate if the responder has multiple certificates which contain useable public keys (e.g. if the certificate is not for signatures only, either due to certificate restrictions or algorithmic restrictions). If the HASh payload is sent it MUST be the first payload of the second message exchange and MUST be followed by the encrypted nonce. If the HASH payload is not sent, the first payload of the second message exchange MUST be the encrypted nonce. In addition, the initiator my optionally send a certificate payload to provide the responder with a public key with which to respond.

When using the revised encryption mode for authentication, Main Mode is defined as follows.

```
Initiator                  Responder
-----------                -----------
HDR, SA                 -->
                   <--  HDR, SA
HDR, [ HASH(1), ]
  <Ni_b>Pubkey_r,
  <KE_b>Ke_i,
  <IDii_b>Ke_i,
  [<<Cert-I_b>Ke_i]     -->
                         HDR, <Nr_b>PubKey_i,
                            <KE_b>Ke_r,
                   <--      <IDir_b>Ke_r,
HDR*, HASH_I             -->
                   <--  HDR*, HASH_R
```

Aggressive Mode authenticated with the revised encryption method is described as follows:

```
Initiator                  Responder
-----------                -----------
HDR, SA, [ HASH(1),]
  <Ni_b>Pubkey_r,
  <KE_b>Ke_i, <IDii_b>Ke_i
  [, <Cert-I_b>Ke_i ]    -->
                         HDR, SA, <Nr_b>PubKey_i,
                            <KE_b>Ke_r, <IDir_b>Ke_r,
                   <--      HASH_R
HDR, HASH_I              -->
```

where HASH(1) is identical to section 5.2. Ke_i and Ke_r are keys to the symmetric encryption algorithm negotiated in the SA payload exchange. Only the body of the payloads are encrypted (in both public key and symmetric operations), the generic payload headers are left in the clear. The payload length includes that added to perform encryption.

The symmetric cipher keys are derived from the decrypted nonces as follows. First the values Ne_i and Ne_r are computed:

Ne_i = prf(Ni_b, CKY-I)
Ne_r = prf(Nr_b, CKY-R)

The keys Ke_i and Ke_r are then taken from Ne_i and Ne_r respectively in the manner described in Appendix B used to derive symmetric keys for use with the negotiated encryption algorithm. If the length of the output of the negotiated prf is greater than or equal to the key length requirements of the cipher, Ke_i and Ke_r are derived from the most significant bits of Ne_i and Ne_r respectively. If the desired length of Ke_i and Ke_r exceed the length of the output of the prf the necessary number of bits is obtained by repeatedly

feeding the results of the prf back into itself and concatenating the result until the necessary number has been achieved. For example, if the negotiated encryption algorithm requires 320 bits of key and the output of the prf is only 128 bits, Ke_i is the most significant 320 bits of K, where

K = K1 | K2 | K3 and
K1 = prf(Ne_i, 0)
K2 = prf(Ne_i, K1)
K3 = prf(Ne_i, K2)

For brevity, only derivation of Ke_i is shown; Ke_r is identical. The length of the value 0 in the computation of K1 is a single octet. Note that Ne_i, Ne_r, Ke_i, and Ke_r are all ephemeral and MUST be discarded after use.

Save the requirements on the location of the optional HASH payload and the mandatory nonce payload there are no further payload requirements. All payloads-- in whatever order-- following the encrypted nonce MUST be encrypted with Ke_i or Ke_r depending on the direction.

If CBC mode is used for the symmetric encryption then the initialization vectors (IVs) are set as follows. The IV for encrypting the first payload following the nonce is set to 0 (zero). The IV for subsequent payloads encrypted with the ephemeral symmetric cipher key, Ke_i, is the last ciphertext block of the previous payload. Encrypted payloads are padded up to the nearest block size. All padding bytes, except for the last one, contain 0x00. The last byte of the padding contains the number of the padding bytes used, excluding the last one. Note that this means there will always be padding.

5.4 Phase 1 Authenticated With a Pre-Shared Key

A key derived by some out-of-band mechanism may also be used to authenticate the exchange. The actual establishment of this key is out of the scope of this document.

When doing a pre-shared key authentication, Main Mode is defined as follows:

```
 Initiator                 Responder
----------                 -----------
 HDR, SA          -->
                  <--   HDR, SA
 HDR, KE, Ni      -->
                  <--   HDR, KE, Nr
 HDR*, IDii, HASH_I -->
                  <--   HDR*, IDir, HASH_R
```

Aggressive mode with a pre-shared key is described as follows:

```
 Initiator                 Responder
-----------                -----------
 HDR, SA, KE, Ni, IDii -->
                  <--   HDR, SA, KE, Nr, IDir, HASH_R
 HDR, HASH_I      -->
```

When using pre-shared key authentication with Main Mode the key can only be identified by the IP address of the peers since HASH_I must be computed before the initiator has processed IDir. Aggressive Mode allows for a wider range of identifiers of the pre-shared secret to be used. In addition, Aggressive Mode allows two parties to maintain multiple, different pre-shared keys and identify the correct one for a particular exchange.

5.5 Phase 2 - Quick Mode

Quick Mode is not a complete exchange itself (in that it is bound to a phase 1 exchange), but is used as part of the SA negotiation process (phase 2) to derive keying material and negotiate shared policy for non-ISAKMP SAs. The information exchanged along with Quick Mode MUST be protected by the ISAKMP SA-

- i.e. all payloads except the ISAKMP header are encrypted. In Quick Mode, a HASH payload MUST immediately follow the ISAKMP header and a SA payload MUST immediately follow the HASH. This HASH authenticates the message and also provides liveliness proofs.

The message ID in the ISAKMP header identifies a Quick Mode in progress for a particular ISAKMP SA which itself is identified by the cookies in the ISAKMP header. Since each instance of a Quick Mode uses a unique initialization vector (see Appendix B) it is possible to have multiple simultaneous Quick Modes, based off a single ISAKMP SA, in progress at any one time.

Quick Mode is essentially a SA negotiation and an exchange of nonces that provides replay protection. The nonces are used to generate fresh key material and prevent replay attacks from generating bogus security associations. An optional Key Exchange payload can be exchanged to allow for an additional Diffie-Hellman exchange and exponentiation per Quick Mode. While use of the key exchange payload with Quick Mode is optional it MUST be supported.

Base Quick Mode (without the KE payload) refreshes the keying material derived from the exponentiation in phase 1. This does not provide PFS. Using the optional KE payload, an additional exponentiation is performed and PFS is provided for the keying material.

The identities of the SAs negotiated in Quick Mode are implicitly assumed to be the IP addresses of the ISAKMP peers, without any I mplied constraints on the protocol or port numbers allowed, unless client identifiers are specified in Quick Mode. If ISAKMP is acting as a client negotiator on behalf of another party, the identities of the parties MUST be passed as IDci and then IDcr. Local policy will dictate whether the proposals are acceptable for the identities specified. If the client identities are not acceptable to the Quick Mode responder (due to policy or other reasons), a Notify payload with Notify Message Type INVALID-ID-INFORMATION (18) SHOULD be sent.

The client identities are used to identify and direct traffic to the appropriate tunnel in cases where multiple tunnels exist between two peers and also to allow for unique and shared SAs with different granularities.

All offers made during a Quick Mode are logically related and must be consistant. For example, if a KE payload is sent, the attribute describing the Diffie-Hellman group (see section 6.1 and [Pip97]) MUST be included in every transform of every proposal of every SA being negotiated. Similarly, if client identities are used, they MUST apply to every SA in the negotiation.

Quick Mode is defined as follows:

```
   Initiator                    Responder
  -----------                  -----------
   HDR*, HASH(1), SA, Ni
     [, KE ] [, IDci, IDcr ] -->
                        <--   HDR*, HASH(2), SA, Nr
                                 [, KE ] [, IDci, IDcr ]
   HDR*, HASH(3)           -->
```

Where:
HASH(1) is the prf over the message id (M-ID) from the ISAKMP header concatenated with the entire message that follows the hash including all payload headers, but excluding any padding added for encryption. HASH(2) is identical to HASH(1) except the initiator's nonce-- Ni, minus the payload header-- is added after M-ID but before the complete message. The addition of the nonce to HASH(2) is for a liveliness proof. HASH(3)-- for liveliness-- is the prf over the value zero represented as a single octet, followed by a concatenation of the message id and the two nonces-- the initiator's followed by the responder's-- minus the payload header. In other words, the hashes for the above exchange are:

```
HASH(1) = prf(SKEYID_a, M-ID | SA | Ni [ | KE ] [ | IDci | IDcr )
HASH(2) = prf(SKEYID_a, M-ID | Ni_b | SA | Nr [ | KE ] [ | IDci |    IDcr )
HASH(3) = prf(SKEYID_a, 0 | M-ID | Ni_b | Nr_b)
```

With the exception of the HASH, SA, and the optional ID payloads, there are no payload ordering restrictions on Quick Mode. HASH(1) and HASH(2) may differ from the illustration above if the order of payloads in the message differs from the illustrative example or if any optional payloads, for example a notify payload, have been chained to the message.

If PFS is not needed, and KE payloads are not exchanged, the new keying material is defined as

KEYMAT = prf(SKEYID_d, protocol | SPI | Ni_b | Nr_b).

If PFS is desired and KE payloads were exchanged, the new keying material is defined as

KEYMAT = prf(SKEYID_d, g(qm)^xy | protocol | SPI | Ni_b | Nr_b)

where g(qm)^xy is the shared secret from the ephemeral Diffie-Hellman exchange of this Quick Mode.

In either case, "protocol" and "SPI" are from the ISAKMP Proposal Payload that contained the negotiated Transform. A single SA negotiation results in two security assocations—one inbound and one outbound. Different PIs for each SA (one chosen by the initiator, the other by the responder) guarantee a different key for each ection. The SPI chosen by the destination of the SA is used to derive KEYMAT for that SA.

For situations where the amount of keying material desired is greater than that supplied by the prf, KEYMAT is expanded by feeding the results of the prf back into itself and concatenating results until the required keying material has been reached. In other words,

```
KEYMAT = K1 | K2 | K3 | ...
where
  K1 = prf(SKEYID_d, [ g(qm)^xy | ] protocol | SPI | Ni_b | Nr_b)
  K2 = prf(SKEYID_d, K1 | [ g(qm)^xy | ] protocol | SPI | Ni_b | Nr_b)
  K3 = prf(SKEYID_d, K2 | [ g(qm)^xy | ] protocol | SPI | Ni_b | Nr_b)
  etc.
```

This keying material (whether with PFS or without, and whether derived directly or through concatenation) MUST be used with the negotiated SA. It is up to the service to define how keys are derived from the keying material.

In the case of an ephemeral Diffie-Hellman exchange in Quick Mode, the exponential (g(qm)^xy) is irretreivably removed from the current state and SKEYID_e and SKEYID_a (derived from phase 1 negotiation) continue to protect and authenticate the ISAKMP SA and SKEYID_d continues to be used to derive keys.

Using Quick Mode, multiple SA's and keys can be negotiated with one exchange as follows:

```
 Initiator                  Responder
-----------                 -----------
 HDR*, HASH(1), SA0, SA1, Ni,
   [, KE ] [, IDci, IDcr ] -->
                    <--   HDR*, HASH(2), SA0, SA1, Nr,
                          [, KE ] [, IDci, IDcr ]
 HDR*, HASH(3)            -->
```

The keying material is derived identically as in the case of a single SA. In this case (negotiation of two SA payloads) the result would be four security associations-- two each way for both SAs.

5.6 New Group Mode

New Group Mode MUST NOT be used prior to establishment of an ISAKMP SA. The description of a new group MUST only follow phase 1 negotiation. (It is not a phase 2 exchange, though).

```
 Initiator                  Responder
```

```
-----------                 -----------
 HDR*, HASH(1), SA      -->
                   <--    HDR*, HASH(2), SA
```

where HASH(1) is the prf output, using SKEYID_a as the key, and the message-ID from the ISAKMP header concatenated with the entire SA proposal, body and header, as the data; HASH(2) is the prf output, using SKEYID_a as the key, and the message-ID from the ISAKMP header concatenated with the reply as the data. In other words the hashes for the above exchange are:

HASH(1) = prf(SKEYID_a, M-ID | SA)
HASH(2) = prf(SKEYID_a, M-ID | SA)

The proposal will specify the characteristics of the group (see appendix A, "Attribute Assigned Numbers"). Group descriptions for private Groups MUST be greater than or equal to 2^15. If the group is not acceptable, the responder MUST reply with a Notify payload with the message type set to ATTRIBUTES-NOT-SUPPORTED (13).

ISAKMP implementations MAY require private groups to expire with the SA under which they were established.

Groups may be directly negotiated in the SA proposal with Main Mode. To do this the component parts-- for a MODP group, the type, prime and generator; for a EC2N group the type, the Irreducible Polynomial, Group Generator One, Group Generator Two, Group Curve A, Group Curve B and Group Order-- are passed as SA attributes (see Appendix A). Alternately, the nature of the group can be hidden using New Group Mode and only the group identifier is passed in the clear during phase 1 negotiation.

5.7 ISAKMP Informational Exchanges

This protocol protects ISAKMP Informational Exchanges when possible. Once the ISAKMP security association has been established (and SKEYID_e and SKEYID_a have been generated) ISAKMP information Exchanges, when used with this protocol, are as follows:

```
Initiator                 Responder
-----------               -----------
 HDR*, HASH(1), N/D     -->
```

where N/D is either an ISAKMP Notify Payload or an ISAKMP Delete Payload and HASH(1) is the prf output, using SKEYID_a as the key, and a M-ID unique to this exchange concatenated with the entire informational payload (either a Notify or Delete) as the data. In other words, the hash for the above exchange is:

HASH(1) = prf(SKEYID_a, M-ID | N/D)

As noted the message ID in the ISAKMP header-- and used in the prf computation-- is unique to this exchange and MUST NOT be the same as the message ID of another phase 2 exchange which generated this informational exchange. The derivation of the initialization vector, used with SKEYID_e to encrypt this message, is described in Appendix B.

If the ISAKMP security association has not yet been established at the time of the Informational Exchange, the exchange is done in the clear without an accompanying HASH payload.

6 Oakley Groups

With IKE, the group in which to do the Diffie-Hellman exchange is negotiated. Four groups-- values 1 through 4-- are defined below. These groups originated with the Oakley protocol and are therefore called "Oakley Groups". The attribute class for "Group" is defined in Appendix A. All values 2^15 and higher are used for private group identifiers. For a discussion on the strength of the default Oakley groups please see the Security Considerations section below.

These groups were all generated by Richard Schroeppel at the University of Arizona. Properties of these groups are described in [Orm96].

6.1 First Oakley Default Group

Oakley implementations MUST support a MODP group with the following prime and generator. This group is assigned id 1 (one).

The prime is: 2^768 - 2 ^704 - 1 + 2^64 * { [2^638 pi] + 149686 } Its hexadecimal value is

FFFFFFFF FFFFFFFF C90FDAA2 2168C234 C4C6628B 80DC1CD1
29024E08 8A67CC74 020BBEA6 3B139B22 514A0879 8E3404DD
EF9519B3 CD3A431B 302B0A6D F25F1437 4FE1356D 6D51C245
E485B576 625E7EC6 F44C42E9 A63A3620 FFFFFFFF FFFFFFFF

The generator is: 2.

6.2 Second Oakley Group

IKE implementations SHOULD support a MODP group with the following prime and generator. This group is assigned id 2 (two).

The prime is 2^1024 - 2^960 - 1 + 2^64 * { [2^894 pi] + 129093 }. Its hexadecimal value is

FFFFFFFF FFFFFFFF C90FDAA2 2168C234 C4C6628B 80DC1CD1
29024E08 8A67CC74 020BBEA6 3B139B22 514A0879 8E3404DD
EF9519B3 CD3A431B 302B0A6D F25F1437 4FE1356D 6D51C245
E485B576 625E7EC6 F44C42E9 A637ED6B 0BFF5CB6 F406B7ED
EE386BFB 5A899FA5 AE9F2411 7C4B1FE6 49286651 ECE65381
FFFFFFFF FFFFFFFF

The generator is 2 (decimal)

6.3 Third Oakley Group

IKE implementations SHOULD support a EC2N group with the following characteristics. This group is assigned id 3 (three). The curve is based on the Galois Field GF[2^155]. The field size is 155. The irreducible polynomial for the field is:

u^155 + u^62 + 1.
The equation for the elliptic curve is:
y^2 + xy = x^3 + ax^2 + b.

Field Size: 155
Group Prime/Irreducible Polynomial:
0x080000000000000000000004000000000000001
Group Generator One: 0x7b
Group Curve A: 0x0
Group Curve B: 0x07338f

Group Order: 0X080000000000000000057db5698537193aef944

The data in the KE payload when using this group is the value x from the solution (x,y), the point on the curve chosen by taking the randomly chosen secret Ka and computing Ka*P, where * is the repetition of the group addition and double operations, P is the curve point with x coordinate equal to generator 1 and the coordinate determined from the defining equation. The equation of curve is implicitly known by the Group Type and the A and B coefficients. There are two possible values for the y coordinate; either one can be used successfully (the two parties need not agree on the selection).

6.4 Fourth Oakley Group

IKE implementations SHOULD support a EC2N group with the following characteristics. This group is assigned id 4 (four). The curve is based on the Galois Field GF[2^185]. The field size is 185. The irreducible polynomial for the field is:
u^185 + u^69 + 1. The
equation for the elliptic curve is:
y^2 + xy = x^3 + ax^2 + b.

Field Size: 185
Group Prime/Irreducible Polynomial:
0x020000000000000000000000000020000000000000000001
Group Generator One: 0x18
Group Curve A: 0x0
Group Curve B: 0x1ee9

Group Order: 0X01ffffffffffffffffffffdbf2f889b73e484175f94ebc

The data in the KE payload when using this group will be identical to that as when using Oakley Group 3 (three).

Other groups can be defined using New Group Mode. These default groups were generated by Richard Schroeppel at the University of Arizona. Properties of these primes are described in [Orm96].

7. Payload Explosion for a Complete IKE Exchange

This section illustrates how the IKE protocol is used to:

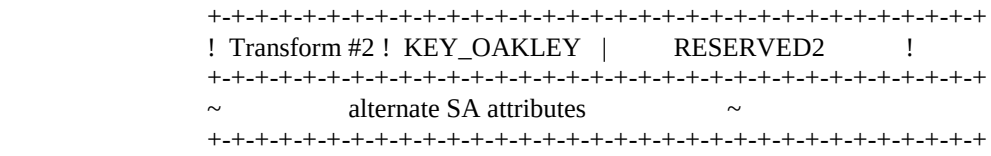
- establish a secure and authenticated channel between ISAKMP processes (phase 1); and

- generate key material for, and negotiate, an IPsec SA (phase 2).

7.1 Phase 1 using Main Mode

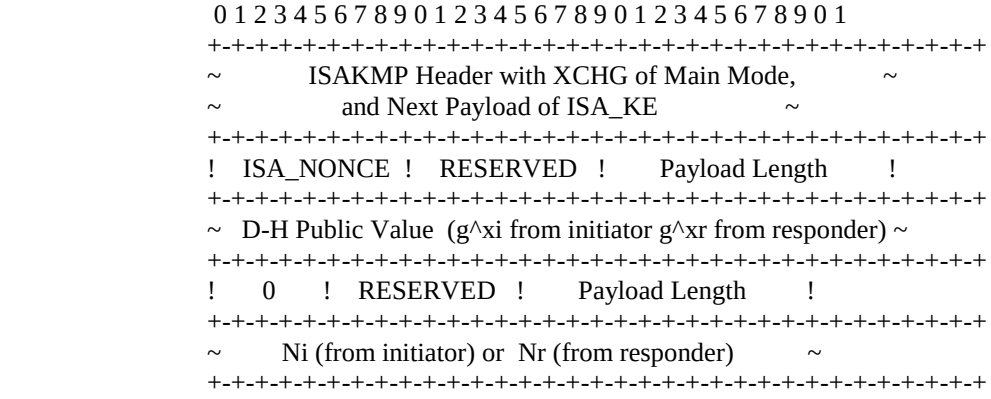
The following diagram illustrates the payloads exchanged between the two parties in the first round trip exchange. The initiator MAY propose several proposals; the responder MUST reply with one.

```
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~           ISAKMP Header with XCHG of Main Mode,           ~
~              and Next Payload of ISA_SA                   ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!      0      !    RESERVED   !       Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!              Domain of Interpretation                     !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!                    Situation                              !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!      0      !    RESERVED   !       Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!  Proposal #1  ! PROTO_ISAKMP  ! SPI size = 0  | # Transforms  !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!   ISA_TRANS   !    RESERVED   !       Payload Length        !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!  Transform #1 !  KEY_OAKLEY   |        RESERVED2           !
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
~              prefered SA attributes                       ~
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
!      0      !    RESERVED   !       Payload Length        !
```

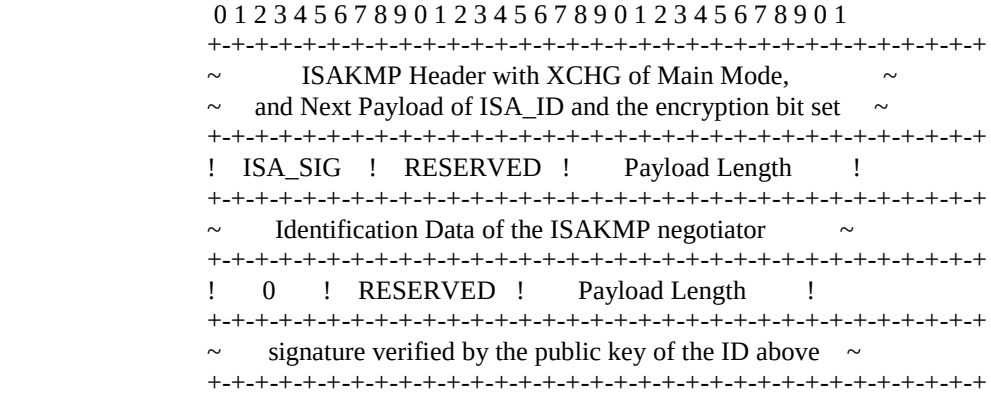


The responder replies in kind but selects, and returns, one transform proposal (the ISAKMP SA attributes).

The second exchange consists of the following payloads:



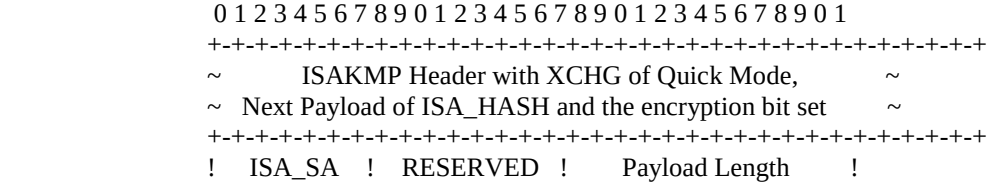
The shared keys, SKEYID_e and SKEYID_a, are now used to protect and authenticate all further communication. Note that both SKEYID_e and SKEYID_a are unauthenticated.

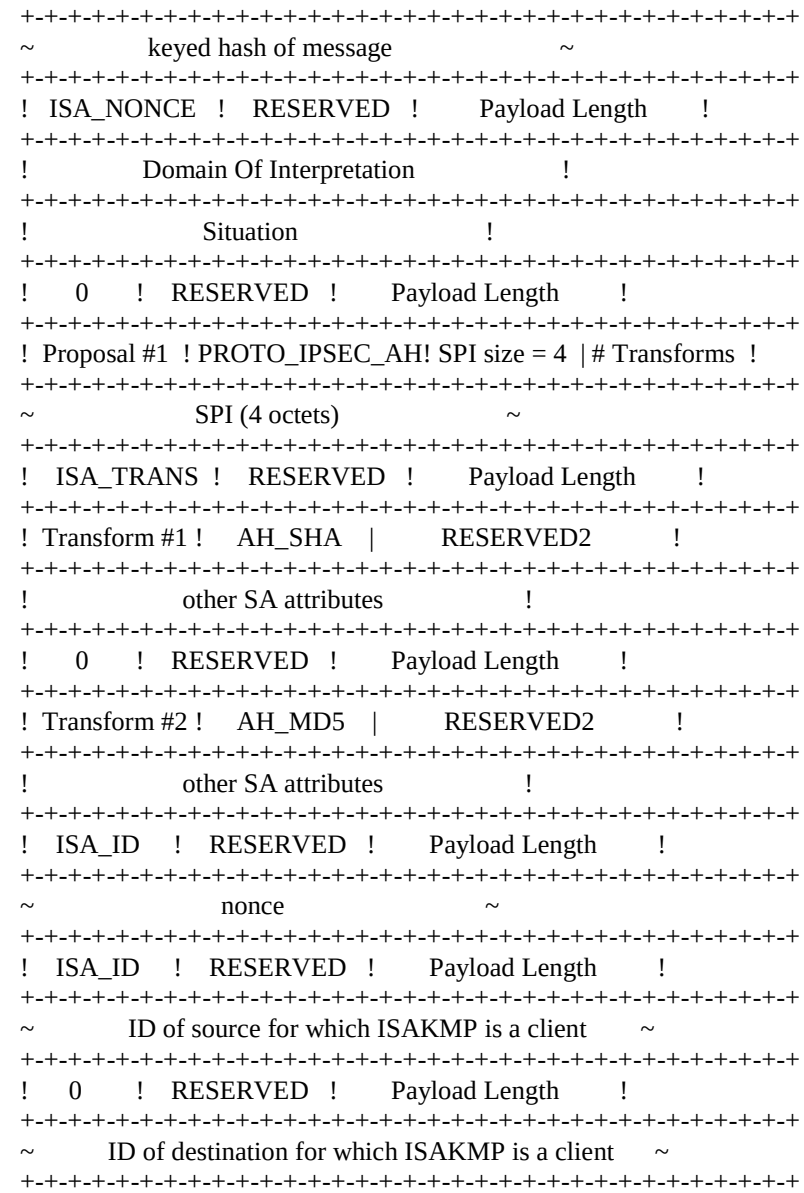


The key exchange is authenticated over a signed hash as described in section 5.1. Once the signature has been verified using the authentication algorithm negotiated as part of the ISAKMP SA, the shared keys, SKEYID_e and SKEYID_a can be marked as authenticated. (For brevity, certificate payloads were not exchanged).

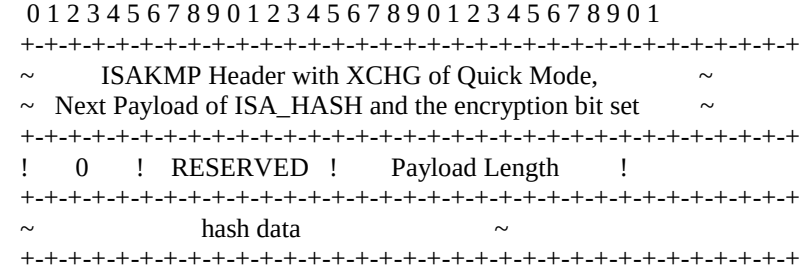
7.2 Phase 2 using Quick Mode

The following payloads are exchanged in the first round of Quick Mode with ISAKMP SA negotiation. In this hypothetical exchange, the ISAKMP negotiators are proxies for other parties which have requested authentication.





where the contents of the hash are described in 5.5 above. The responder replies with a similar message which only contains one transform-- the selected AH transform. Upon receipt, the initiator can provide the key engine with the negotiated security association and the keying material. As a check against replay attacks, the responder waits until receipt of the next message.



where the contents of the hash are described in 5.5 above.

8. Perfect Forward Secrecy Example

This protocol can provide PFS of both keys and identities. The identies of both the ISAKMP negotiating peer and, if applicable, the identities for whom the peers are negotiating can be protected with PFS.

To provide Perfect Forward Secrecy of both keys and all identities, two parties would perform the following:

A Main Mode Exchange to protect the identities of the ISAKMP peers.
This establishes an ISAKMP SA. o A Quick Mode Exchange to negotiate other security protocol protection. This establishes a SA on each end for this protocol. o Delete the ISAKMP SA and its associated state.

Since the key for use in the non-ISAKMP SA was derived from the single ephemeral Diffie-Hellman exchange PFS is preserved.

To provide Perfect Forward Secrecy of merely the keys of a non-ISAKMP security association, it in not necessary to do a phase 1 exchange if an ISAKMP SA exists between the two peers. A single Quick Mode in which the optional KE payload is passed, and an additional Diffie-Hellman exchange is performed, is all that is required. At this point the state derived from this Quick Mode must be deleted from the ISAKMP SA as described in section 5.5.

9. Implementation Hints

Using a single ISAKMP Phase 1 negotiation makes subsequent Phase 2 negotiations extremely quick. As long as the Phase 1 state remains cached, and PFS is not needed, Phase 2 can proceed without any exponentiation. How many Phase 2 negotiations can be performed for a single Phase 1 is a local policy issue. The decision will depend on the strength of the algorithms being used and level of trust in the peer system.

An implementation may wish to negotiate a range of SAs when performing Quick Mode. By doing this they can speed up the "re- keying". Quick Mode defines how KEYMAT is defined for a range of SAs. When one peer feels it is time to change SAs they simply use the next one within the stated range. A range of SAs can be established by negotiating multiple SAs (identical attributes, different SPIs) with one Quick Mode.

An optimization that is often useful is to establish Security Associations with peers before they are needed so that when they become needed they are already in place. This ensures there would be no delays due to key management before initial data transmission. This optimization is easily implemented by setting up more than one Security Association with a peer for each requested Security Association and caching those not immediately used.

Also, if an ISAKMP implementation is alerted that a SA will soon be needed (e.g. to replace an existing SA that will expire in the near future), then it can establish the new SA before that new SA is needed.

The base ISAKMP specification describes conditions in which one party of the protocol may inform the other party of some activity—either deletion of a security association or in response to some error in the protocol such as a signature verification failed or a payload failed to decrypt. It is strongly suggested that these Informational exchanges not be responded to under any circumstances. Such a condition may result in a "notify war" in which failure to understand a message results in a notify to the peer who cannot understand it and sends his own notify back which is also not understood.

10. Security Considerations

This entire memo discusses a hybrid protocol, combining parts of Oakley and parts of SKEME with ISAKMP, to negotiate, and derive keying material for, security associations in a secure and authenticated manner.

Confidentiality is assured by the use of a negotiated encryption algorithm. Authentication is assured by the use of a negotiated method: a digital signature algorithm; a public key algorithm which supports encryption; or, a pre-shared key. The confidentiality and authentication of this exchange is only as good as the attributes negotiated as part of the ISAKMP security association.

Repeated re-keying using Quick Mode can consume the entropy of the Diffie-Hellman shared secret. Implementors should take note of this fact and set a limit on Quick Mode Exchanges between exponentiations. This memo does not prescribe such a limit.

Perfect Forward Secrecy (PFS) of both keying material and identities is possible with this protocol. By specifying a Diffie-Hellman group, and passing public values in KE payloads, ISAKMP peers can establish PFS of keys-- the identities would be protected by SKEYID_e from the ISAKMP SA and would therefore not be protected by PFS. If PFS of both keying material and identities is desired, an ISAKMP peer MUST establish only one non-ISAKMP security association (e.g. Ipsec Security Association) per ISAKMP SA. PFS for keys and identities is accomplished by deleting the ISAKMP SA (and optionally issuing a DELETE message) upon establishment of the single non-ISAKMP SA. In this way a phase one negotiation is uniquely tied to a single phase two negotiation, and the ISAKMP SA established during phase one negotiation is never used again.

The strength of a key derived from a Diffie-Hellman exchange using any of the groups defined here depends on the inherent strength of the group, the size of the exponent used, and the entropy provided by the random number generator used. Due to these inputs it is difficult to determine the strength of a key for any of the defined groups. The default Diffie-Hellman group (number one) when used with a strong random number generator and an exponent no less than 160 bits is sufficient to use for DES. Groups two through four provide greater security. Implementations should make note of these conservative estimates when establishing policy and negotiating security parameters.

Note that these limitations are on the Diffie-Hellman groups themselves. There is nothing in IKE which prohibits using stronger groups nor is there anything which will dilute the strength obtained from stronger groups. In fact, the extensible framework of IKE encourages the definition of more groups; use of elliptical curve groups will greatly increase strength using much smaller numbers. For situations where defined groups provide insufficient strength New Group Mode can be used to exchange a Diffie-Hellman group which provides the necessary strength. In is incumbent upon implementations to check the primality in groups being offered and independently arrive at strength estimates.

It is assumed that the Diffie-Hellman exponents in this exchange are erased from memory after use. In particular, these exponents must not be derived from long-lived secrets like the seed to a pseudo-random generator.

IKE exchanges maintain running initialization vectors (IV) where the last ciphertext block of the last message is the IV for the next message. To prevent retransmissions (or forged messages with valid cookies) from causing exchanges to get out of sync IKE implementations SHOULD NOT update their running IV until the decrypted message has passed a basic sanity check and has been determined to actually advance the IKE state machine-- i.e. it is not a retransmission.

While the last roundtrip of Main Mode (and optionally the last message of Aggressive Mode) is encrypted it is not, strictly speaking, authenticated. An active substitution attack on the ciphertext could result in payload corruption. If such an attack corrupts mandatory payloads it would be detected by an authentication failure, but if it corrupts any optional payloads (e.g. notify payloads chained onto the last message of a Main Mode exchange) it might not be detectable.

11. IANA Considerations

This document contains many "magic numbers" to be maintained by the IANA. This section explains the criteria to be used by the IANA to assign additional numbers in each of these lists.

11.1 Attribute Classes

Attributes negotiated in this protocol are identified by their class. Requests for assignment of new classes must be accompanied by a standards-track RFC which describes the use of this attribute.

11.2 Encryption Algorithm Class

Values of the Encryption Algorithm Class define an encryption algorithm to use when called for in this document. Requests for assignment of new encryption algorithm values must be accompanied by a reference to a standards-track or Informational RFC or a reference to published cryptographic literature which escribes this algorithm.

11.3 Hash Algorithm

Values of the Hash Algorithm Class define a hash algorithm to use when called for in this document. Requests for assignment of new hash algorithm values must be accompanied by a reference to a standards-track or Informational RFC or a reference to published cryptographic literature which describes this algorithm. Due to the key derivation and key expansion uses of HMAC forms of hash algorithms in IKE, requests for assignment of new hash algorithm values must take into account the cryptographic properties-- e.g it's resistance to collision-- of the hash algorithm itself.

11.4 Group Description and Group Type

Values of the Group Description Class identify a group to use in a Diffie-Hellman exchange. Values of the Group Type Class define the type of group. Requests for assignment of new groups must be accompanied by a reference to a standards-track or Informational RFC which describes this group. Requests for assignment of new group types must be accompanied by a reference to a standards-track or Informational RFC or by a reference to published cryptographic or mathmatical literature which describes the new type.

11.5 Life Type

Values of the Life Type Class define a type of lifetime to which the ISAKMP Security Association applies. Requests for assignment of new life types must be accompanied by a detailed description of the units of this type and its expiry.

12. Acknowledgements

This document is the result of close consultation with Hugo Krawczyk, Douglas Maughan, Hilarie Orman, Mark Schertler, Mark Schneider, and Jeff Turner. It relies on protocols which were written by them. Without their interest and dedication, this would not have been written.

Special thanks Rob Adams, Cheryl Madson, Derrell Piper, Harry Varnis, and Elfed Weaver for technical input, encouragement, and various sanity checks along the way.

We would also like to thank the many members of the IPSec working group that contributed to the development of this protocol over the past year.

13. References

[CAST] Adams, C., "The CAST-128 Encryption Algorithm", RFC 2144, May 1997.

[BLOW] Schneier, B., "The Blowfish Encryption Algorithm", Dr. Dobb's Journal, v. 19, n. 4, April 1994.

[Bra97] Bradner, S., "Key Words for use in RFCs to indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[DES] ANSI X3.106, "American National Standard for Information Systems-Data Link Encryption", American National Standards Institute, 1983.

[DH] Diffie, W., and Hellman M., "New Directions in Cryptography", IEEE Transactions on Information Theory, V. IT-22, n. 6, June 1977.

[DSS] NIST, "Digital Signature Standard", FIPS 186, National Institute of Standards and Technology, U.S. Department of Commerce, May, 1994.

[IDEA] Lai, X., "On the Design and Security of Block Ciphers," ETH Series in Information Processing, v. 1, Konstanz: Hartung- Gorre Verlag, 1992

[KBC96] Krawczyk, H., Bellare, M., and R. Canetti, "HMAC: Keyed- Hashing for Message Authentication", RFC 2104, February 1997.

[SKEME] Krawczyk, H., "SKEME: A Versatile Secure Key Exchange Mechanism for Internet", from IEEE Proceedings of the 1996 Symposium on Network and Distributed Systems Security.

[MD5] Rivest, R., "The MD5 Message Digest Algorithm", RFC 1321, April 1992.

[MSST98] Maughhan, D., Schertler, M., Schneider, M., and J. Turner, "Internet Security Association and Key Management Protocol (ISAKMP)", RFC 2408, November 1998.

[Orm96] Orman, H., "The Oakley Key Determination Protocol", RFC 2412, November 1998.

[PKCS1] RSA Laboratories, "PKCS #1: RSA Encryption Standard", November 1993.

[Pip98] Piper, D., "The Internet IP Security Domain Of Interpretation for ISAKMP", RFC 2407, November 1998.

[RC5] Rivest, R., "The RC5 Encryption Algorithm", Dr. Dobb's Journal, v. 20, n. 1, January 1995.

[RSA] Rivest, R., Shamir, A., and Adleman, L., "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems", Communications of the ACM, v. 21, n. 2, February 1978.

[Sch96] Schneier, B., "Applied Cryptography, Protocols, Algorithms, and Source Code in C", 2nd edition.

[SHA] NIST, "Secure Hash Standard", FIPS 180-1, National Institue of Standards and Technology, U.S. Department of Commerce, May 1994.

[TIGER] Anderson, R., and Biham, E., "Fast Software Encryption", Springer LNCS v. 1039, 1996.

Appendix A

This is a list of DES Weak and Semi-Weak keys. The keys come from [Sch96]. All keys are listed in hexidecimal.

DES Weak Keys
0101 0101 0101 0101
1F1F 1F1F E0E0 E0E0
E0E0 E0E0 1F1F 1F1F
FEFE FEFE FEFE FEFE

DES Semi-Weak Keys
01FE 01FE 01FE 01FE
1FE0 1FE0 0EF1 0EF1
01E0 01E0 01F1 01F1
1FFE 1FFE 0EFE 0EFE
011F 011F 010E 010E
E0FE E0FE F1FE F1FE

FE01 FE01 FE01 FE01
E01F E01F F10E F10E
E001 E001 F101 F101
FE1F FE1F FE0E FE0E
1F01 1F01 0E01 0E01
FEE0 FEE0 FEF1 FEF1

Attribute Assigned Numbers

Attributes negotiated during phase one use the following definitions. Phase two attributes are defined in the applicable DOI specification (for example, IPsec attributes are defined in the IPsec DOI), with the exception of a group description when Quick Mode includes an ephemeral Diffie-Hellman exchange. Attribute types can be either Basic (B) or Variable-length (V). Encoding of these attributes is defined in the base ISAKMP specification as Type/Value (Basic) and Type/Length/Value (Variable).

Attributes described as basic MUST NOT be encoded as variable. Variable length attributes MAY be encoded as basic attributes if their value can fit into two octets. If this is the case, an attribute offered as variable (or basic) by the initiator of this protocol MAY be returned to the initiator as a basic (or variable).

Attribute Classes

class	value	type
Encryption Algorithm	1	B
Hash Algorithm	2	B
Authentication Method	3	B
Group Description	4	B
Group Type	5	B
Group Prime/Irreducible Polynomial	6	V
Group Generator One	7	V
Group Generator Two	8	V
Group Curve A	9	V
Group Curve B	10	V
Life Type	11	B
Life Duration	12	V
PRF	13	B
Key Length	14	B
Field Size	15	B
Group Order	16	V

values 17-16383 are reserved to IANA. Values 16384-32767 are for private use among mutually consenting parties.

Class Values

- Encryption Algorithm

		Defined In
DES-CBC	1	RFC 2405
IDEA-CBC	2	
Blowfish-CBC	3	
RC5-R16-B64-CBC	4	
3DES-CBC	5	
CAST-CBC	6	

values 7-65000 are reserved to IANA. Values 65001-65535 are for private use among mutually consenting parties.

- Hash Algorithm

		Defined In
MD5	1	RFC 1321
SHA	2	FIPS 180-1
Tiger	3	See Reference [TIGER]

values 4-65000 are reserved to IANA. Values 65001-65535 are for rivate use among mutually consenting parties.

- Authentication Method

pre-shared key 1
DSS signatures 2
RSA signatures 3
Encryption with RSA 4
Revised encryption with RSA 5

values 6-65000 are reserved to IANA. Values 65001-65535 are for private use among mutually consenting parties.

- Group Description
default 768-bit MODP group (section 6.1) 1

alternate 1024-bit MODP group (section 6.2) 2

EC2N group on GP[2^155] (section 6.3) 3

EC2N group on GP[2^185] (section 6.4) 4

values 5-32767 are reserved to IANA. Values 32768-65535 are for private use among mutually consenting parties.

- Group Type
MODP (modular exponentiation group) 1
ECP (elliptic curve group over GF[P]) 2
EC2N (elliptic curve group over GF[2^N]) 3

values 4-65000 are reserved to IANA. Values 65001-65535 are for private use among mutually consenting parties.

- Life Type
seconds 1
kilobytes 2

values 3-65000 are reserved to IANA. Values 65001-65535 are for private use among mutually consenting parties. For a given "Life Type" the value of the "Life Duration" attribute defines the actual length of the SA life-- either a number of seconds, or a number of kbytes protected.

- PRF
There are currently no pseudo-random functions defined.

values 1-65000 are reserved to IANA. Values 65001-65535 are for private use among mutually consenting parties.

- Key Length

When using an Encryption Algorithm that has a variable length key, this attribute specifies the key length in bits. (MUST use network byte order). This attribute MUST NOT be used when the specified Encryption Algorithm uses a fixed length key.

- Field Size

The field size, in bits, of a Diffie-Hellman group.

- Group Order

The group order of an elliptical curve group. Note the length of his attribute depends on the field size.

Additional Exchanges Defined-- XCHG values
Quick Mode 32

New Group Mode 33

Appendix B

This appendix describes encryption details to be used ONLY when encrypting ISAKMP messages. When a service (such as an IPSEC transform) utilizes ISAKMP to generate keying material, all encryption algorithm specific details (such as key and IV generation, padding, etc...) MUST be defined by that service. ISAKMP does not purport to ever produce keys that are suitable for any encryption algorithm. ISAKMP produces the requested amount of keying material from which the service MUST generate a suitable key. Details, such as weak key checks, are the responsibility of the service.

Use of negotiated PRFs may require the PRF output to be expanded due to the PRF feedback mechanism employed by this document. For example, if the (ficticious) DOORAK-MAC requires 24 bytes of key but produces only 8 bytes of output, the output must be expanded three times before being used as the key for another instance of itself. The output of a PRF is expanded by feeding back the results of the PRF into itself to generate successive blocks. These blocks are concatenated until the requisite number of bytes has been acheived. For example, for pre-shared key authentication with DOORAK-MAC as the negotiated PRF:

```
  BLOCK1-8 = prf(pre-shared-key, Ni_b | Nr_b)
  BLOCK9-16 = prf(pre-shared-key, BLOCK1-8 | Ni_b | Nr_b)
  BLOCK17-24 = prf(pre-shared-key, BLOCK9-16 | Ni_b | Nr_b)
and
  SKEYID = BLOCK1-8 | BLOCK9-16 | BLOCK17-24
```

so therefore to derive SKEYID_d:

```
  BLOCK1-8 = prf(SKEYID, g^xy | CKY-I | CKY-R | 0)
  BLOCK9-16 = prf(SKEYID, BLOCK1-8 | g^xy | CKY-I | CKY-R | 0)
  BLOCK17-24 = prf(SKEYID, BLOCK9-16 | g^xy | CKY-I | CKY-R | 0)
and
  SKEYID_d = BLOCK1-8 | BLOCK9-16 | BLOCK17-24
```

Subsequent PRF derivations are done similarly.

Encryption keys used to protect the ISAKMP SA are derived from SKEYID_e in an algorithm-specific manner. When SKEYID_e is not long enough to supply all the necessary keying material an algorithm requires, the key is derived from feeding the results of a pseudo- random function into itself, concatenating the results, and taking the highest necessary bits.

For example, if (ficticious) algorithm AKULA requires 320-bits of key (and has no weak key check) and the prf used to generate SKEYID_e only generates 120 bits of material, the key for AKULA, would be the first 320-bits of Ka, where:

```
    Ka = K1 | K2 | K3
and
    K1 = prf(SKEYID_e, 0)
    K2 = prf(SKEYID_e, K1)
    K3 = prf(SKEYID_e, K2)
```

where prf is the negotiated prf or the HMAC version of the negotiated hash function (if no prf was negotiated) and 0 is represented by a single octet. Each result of the prf provides 120 bits of material for a total of 360 bits. AKULA would use the first 320 bits of that 360 bit string.

In phase 1, material for the initialization vector (IV material) for CBC mode encryption algorithms is derived from a hash of a concatenation of the initiator's public Diffie-Hellman value and the responder's public Diffie-Hellman value using the negotiated hash algorithm. This is used for the first message only. Each message should be padded up to the nearest block size using bytes containing 0x00. The message length in the header MUST include the length of the pad since this reflects the size of the ciphertext. Subsequent messages MUST use the last CBC encryption block from the previous message as their initialization vector.

In phase 2, material for the initialization vector for CBC mode encryption of the first message of a Quick Mode exchange is derived from a hash of a concatenation of the last phase 1 CBC output block and the phase 2 message id using the negotiated hash algorithm. The IV for subsequent messages within a Quick Mode exchange is the CBC output block from the previous message. Padding and IVs for subsequent messages are done as in phase 1.

After the ISAKMP SA has been authenticated all Informational Exchanges are encrypted using KEYID_e. The initiaization vector for these exchanges is derived in exactly the same fashion as that for a Quick Mode--i.e. it is derived from a hash of a concatenation of the last phase 1 CBC output block and the message id from the ISAKMP header of the Informational Exchange (not the message id from the message that may have prompted the Informational Exchange). Note that the final phase 1 CBC output block, the result of encryption/decryption of the last phase 1 message, must be retained in the ISAKMP SA state to allow for generation of unique IVs for each Quick Mode. Each post- phase 1 exchange (Quick Modes and

Informational Exchanges) generates IVs independantly to prevent Ivs from getting out of sync when two different exchanges are started simultaneously.

In all cases, there is a single bidirectional cipher/IV context. Having each Quick Mode and Informational Exchange maintain a unique context prevents IVs from getting out of sync.

The key for DES-CBC is derived from the first eight (8) non-weak and non-semi-weak (see Appendix A) bytes of SKEYID_e. The IV is the first 8 bytes of the IV material derived above.

The key for IDEA-CBC is derived from the first sixteen (16) bytes of SKEYID_e. The IV is the first eight (8) bytes of the IV material derived above.

The key for Blowfish-CBC is either the negotiated key size, or the first fifty-six (56) bytes of a key (if no key size is negotiated) derived in the aforementioned pseudo-random function feedback method. The IV is the first eight (8) bytes of the IV material derived above.

The key for RC5-R16-B64-CBC is the negotiated key size, or the first sixteen (16) bytes of a key (if no key size is negotiated) derived from the aforementioned pseudo-random function feedback method if necessary. The IV is the first eight (8) bytes of the IV material derived above. The number of rounds MUST be 16 and the block size MUST be 64.

The key for 3DES-CBC is the first twenty-four (24) bytes of a key derived in the aforementioned pseudo-random function feedback method. 3DES-CBC is an encrypt-decrypt-encrypt operation using the first, middle, and last eight (8) bytes of the entire 3DES-CBC key. The IV is the first eight (8) bytes of the IV material derived above.

The key for CAST-CBC is either the negotiated key size, or the first sixteen (16) bytes of a key derived in the aforementioned pseudo- random function feedback method. The IV is the first eight (8) bytes of the IV material derived above.

Support for algorithms other than DES-CBC is purely optional. Some optional algorithms may be subject to intellectual property claims.

Authors' Addresses

Dan Harkins
cisco Systems
170 W. Tasman Dr.
San Jose, California, 95134-1706
United States of America

Phone: +1 408 526 4000
EMail: dharkins@cisco.com

Dave Carrel
76 Lippard Ave.
San Francisco, CA 94131-2947
United States of America

Phone: +1 415 337 8469
EMail: carrel@ipsec.org

Authors' Note

The authors encourage independent implementation, and interoperability testing, of this hybrid protocol.

Full Copyright Statement

MANUAL DE OpenSimMPLS

En el siguiente enlace se puede encontrar todo lo referente al simulador “OpenSimMPLS”
http://gitaca.unex.es/opensimmpls/

IMPLEMENTACIÓN DE NS

INSTALACIÓN DE NS BAJO WINDOWS

Los pasos a seguir serán los siguientes

1. Descargar TCL (versión 8.3.5) disponible en :

http://ftp.activestate.com/ActiveTcl/Windows/8.3.5/ActiveTcl8.3.5.0-2-win32-ix86.exe

2. Instalar TCL

3. Reiniciar el ordenador

4. Crear le directorio para guardar NS, NAM y los ejecutables a simular. C:\ns

5. Descargar el fichero nam-1.0alla-win32.exe en el directorio ns, se puede descargar de:

http://www.isi.edu/nsnam/dist/binary/nam-1.0a11a-win32.exe

6. Para mayor comodidad lo renombraremos como:

nam.exe

7. Descargar el fichero ns2-2.1b9a-win32.exe en el directorio ns, se puede descargar de

http://www.isi.edu/nsnam/dist/binary/ns-2.1b9a-win32.exe

8. Para mayor comodidad lo renombraremos como

ns.exe

SIMULACIONES CON NS

Para empezar un script en Tcl

Siempre empezaremos un programa de la siguiente forma:

set ns [new Simulator]

Esta orden crea una instancia del objeto simulador, diciéndole que vamos a realizar una nueva simulación.

Después abriremos el fichero donde escribiremos los datos obtenidos por el simulador, para después leerlo con NAM y los represente de forma gráfica.

set nf [pen out.nam w]

$ns namtrace-al $nf

Lo que hacemos es abrir o crear un fichero llamado "out.nam", donde escribiremos y leeremos los datos creados por la simulación. En la línea siguiente, los datos creados por NS se guardan en el out.nam que los visualizará. Debemos poner en el programa un procedimiento "finish", que deberá ser definido al principio del programa, el cual cerrará el fichero de valores de trazado (out.nam) y pondrá en marcha a NAM, este deberá tener la siguiente estructura:

```
pocfinish{} {
  global ns nf
     $ns flush-trace
     close $nf
   exec nam out.nam &
   exit=
   }
```

Las siguiente líneas que escribiremos será el tiempo que vamos a simular la red

$ns at<tiempo><elemento>

Donde <tiempo> será el valor en segundos y <elemento> será en que procedimiento se cierra la simulación, PE:

$ns at 5.0 "finish"

Así le decimos a NS que ejecute la simulación durante 5.0 seg y después ejecute el procedimiento "finish". En la última línea del programa será para que arranque la simulación de la siguiente forma

$ns run

Como en todo programa desearemos poner comentarios, para poderlo entender en lecturas posteriores, estos los haremos así:

#Comentario

Este script no es para ejecutar con NS ya que deberemos hacerlo con la red completa, aquí no están definidos ni los nodos, ni los enlaces, etc., que necesitamos en una red y que lo haremos posteriormente.

Estos dos nodos están conectados por una línea duplex de 1Mbps y la transmisión tarda en ir de uno a otro 10ms, es decir, la distancia que se encuentra uno del otro es de 10ms por la velocidad de propagación de la señal en el medio en que se realiza dicha transmisión. P.e. si es una fibra donde v=0.7*c la distancia seria L=0.01*0.7*3 exp 8=2100Km.

Los nodos en NS se definen con la instrucción

set <nombre nodo>[$ns node]

El comando set en OTcl significa que creamos una instancia del objeto a simular.

Siempre que cremos nodos deberemos poner el comando "$ns node". Tenemos dos nodos, n0 y n1, los definiremos en el script de la siguiente forma

ns n0[$ns node]

ns n1[$ns node]

El siguiente paso es unir ambos nodos por una línea, en este caso una línea duplex de la forma

$ns duplex-link $n0 $n1 1Mb 10ms Droptail

En nuestro caso, N1,N2, N3, N4 son los nodos UIO, antenas-UIO, GYE, antenas-GYE; N0 es internet, N5 es datos y msm; N6 y N7 son Perú y Colombia respectivamente. Por tanto el primer paso será crear el agente que envía los datos del nodo 0 (n0), al agente que recibirá los datos en el nodo 1 (n1).

#Crear un agente UDP y unirlo al nodo n0

```
set udp0 [new Agent/UDP]
$ns attach-agent $n0 $udp0
```

Después deberemos crear un genrador de tráfico, CBR que se unirá al agente UDP. Debemos definir el tamaño de los paquetes a enviar, en este caso los haremos de 500bytes y cada paquete será enviado cada 0.005 segundos, es decir se enviarán 200 paquetes por segundo.
Todo esto lo escribiremos así.

```
#creamos una fuente de tráfico CBR? que se une a udp0
set cbr0 [new Aplication/Traffic/CBR]
$cbr0 set paquetSize_500
$cbr0 set interval_0.005
$cbr attach-agent $udp0
```

Después debemos crear el agente que asociaremos all nodo 1, en este caso crearemos un agente nulo.

```
#Creamos un agente de tipo NULL asociados al nodo 1
set null0 [new Agent/Null]
$ns attach-agent $ns1 $null0
```

Ahora debemos conectar ambos agentes

```
#Conectamos el tráfico de la fuente con el del receptor
$ns connect $udp0 $null0
```

También debemos decir cuando el agente CBR cuando enviará datos y cuando dejará de enviarlos y deberemos hacerlo antes de la línea 'ns at 5.0 "finish".

```
#Fijamos el funcionamiento de eventos de CBR
#Arrancará a los 0.5 seg de empezar la simulación
$ns at 0.5 "$cbr0 start"
#Y hará la parada a los 4.5 segundos
$ns at 4.5 "$cbr0 stop"
```

SCRIPT

```
# Creamos un objeto a simular
set ns [new Simulator]
# Abrimos el fichero de trazado de NAM
set nf [pen out.nam w]
$ns namtrace-al $nf
# Definimos el procedimiento "finish"
  poc finish{} {
       global ns nf
       $ns flush-trace
# Cerramos el fichero de trazado
       close $nf
# Ejecutamos NAM con el fichero de trazado
    exec nam out.nam &
    exit 0
    }
#En este apartado definiremos los nodos a simular
set n0[$ns node]
set n1[$ns node]
set n2[$ns node]
set n3[$ns node]
set n4[$ns node]
set n5[$ns node]
set n6[$ns node]
set n7[$ns node]
set n8[$ns node]

#En este otro definiremos los enlaces entre los nodos
$ns duplex-link $n0 $n1 1Mb 10ms DropTail
$ns duplex-link $n0 $n2 1Mb 10ms DropTail
$ns duplex-link $n0 $n3 1Mb 10ms DropTail
$ns duplex-link $n0 $n4 1Mb 10ms DropTail
```

```
$ns duplex-link $n5 $n1 1Mb 10ms DropTail
$ns duplex-link $n5 $n2 1Mb 10ms DropTail
$ns duplex-link $n5 $n3 1Mb 10ms DropTail
$ns duplex-link $n5 $n4 1Mb 10ms DropTail
$ns duplex-link $n1 $n7 1Mb 10ms DropTail
$ns duplex-link $n3 $n6 1Mb 10ms DropTail

#Creamos una topología

$ns duplex-link-op $n0 $n1 orient up
$ns duplex-link-op $n0 $n2 orient right- up
$ns duplex-link-op $n0 $n3 orient right
$ns duplex-link-op $n0 $n4 orient right-down
$ns duplex-link-op $n0 $n5 orient down
$ns duplex-link-op $n1 $n7 orient up
$ns duplex-link-op $n3 $n6 orient right

#Creamos un agente UDP y lo unimos al nodo n0
set udp0 [new Agent/UDP]
$ns attach-agent $n0 $udp0

#Creamos una fuente de trafico CBR que se une a udp0
set cbr0 [new Aplication/Traffic/CBR]
$cbr0 set packetSice_500
$cbr0 set interval_0.005
$cbr0 attach-agent $udp0

#Creamos un agente UDP y lo unimos al nodo n1
set udp1 [new Agent/UDP]
$ns attach-agent $n1 $udp1

#Creamos una fuente de trafico CBR que se une a udp1
set cbr1 [new Aplication/Traffic/CBR]
```

```
$cbr0 set packetSice_500
$cbr0 set interval_0.005
$cbr0 attach-agent $udp1

#Creamos un agente nulo para en nodo n3
set null0 [new Agent/Null]
$ns attach-agent $n3 $null0

$ns connect $udp0 $null0
$ns connect $udp1 $null0

$ns at 0.5 "$cbr0 start"
$ns at 1.0 "$cbr1 start"
$ns at 4.0 "$cbr1 stop"
$ns at  4.5 "$cbr0 stop"
$udp0 class_1
$udp1 class_2

$ns color 1 Blue
$ns color 1 Red

$ns duplex-link-op $n2 $n3 queuePOs 0.5
$ns duplex-link $n3 $n2 1Mb 10ms SFQ

# Definiremos los protocolos que se van a ejecutar en la simulación

#Aplicaciones

#El tiempo de la simulación y llamada la procedimiento
$ns at 5.0 "finish"
# Arrancaremos  la simulación planteada
$ns run
```

```
set ns [new Simulator]
set node_(s1) [$ns node]
set node_(s2) [$ns node]
set node_(r1) [$ns node]
set node_(r2) [$ns node]
set node_(s3) [$ns node]
set node_(s4) [$ns node]
$ns duplex-link $node_(s1) $node_(r1) 10Mb 2ms DropTail
$ns duplex-link $node_(s2) $node_(r1) 10Mb 3ms DropTail
$ns duplex-link $node_(r1) $node_(r2) 1.5Mb 20ms RED
$ns queue-limit $node_(r1) $node_(r2) 25
$ns queue-limit $node_(r2) $node_(r1) 25
$ns duplex-link $node_(s3) $node_(r2) 10Mb 4ms DropTail
$ns duplex-link $node_(s4) $node_(r2) 10Mb 5ms DropTail
$ns duplex-link-op $node_(s1) $node_(r1) orient right-down
$ns duplex-link-op $node_(s2) $node_(r1) orient right-up
$ns duplex-link-op $node_(r1) $node_(r2) orient right
$ns duplex-link-op $node_(r1) $node_(r2) queuePos 0
$ns duplex-link-op $node_(r2) $node_(r1) queuePos 0
$ns duplex-link-op $node_(s3) $node_(r2) orient left-down
$ns duplex-link-op $node_(s4) $node_(r2) orient left-up
# Configuración de los agentes con ventana de máximo 25
set tcp1 [$ns create-connection TCP/Reno $node_(s1) TCPSink $node_(s3) 0]
$tcp1 set window_ 25
set tcp2 [$ns create-connection TCP/Reno $node_(s2) TCPSink $node_(s3) 1]
$tcp2 set window_ 25
set ftp1 [$tcp1 attach-source FTP]
set ftp2 [$tcp2 attach-source FTP]
# Trazado de la cola r1->r2 ira en all.q
set redq [[$ns link $node_(r1) $node_(r2)] queue]
set tchan_ [open all.q w]
# En el archivo irían estos tres valores que representan el tipo,
```

```
# tiempo promedio y tiempo en cola, respectivamente
$redq trace curq_
$redq trace ave_
$redq attach $tchan_
# Programación de los eventos
$ns at 0.0 "$ftp1 start"
$ns at 3.0 "$ftp2 start"
$ns at 10 "finish"
# Al finalizar se inicia el procesamiento para generar los archivos.
# Necesarios para graficar con Xgraph
proc finish {} {
global tchan_
# Esto se ejecuta para cada línea del archivo all.q
set awkCode {
{
if ($1 == "Q" && NF>2) {
print $2, $3 >> "temp.q";
set end $2
}
else if ($1 == "a" && NF>2)
print $2, $3 >> "temp.a";
}
}
set f [open temp.queue w]
puts $f "TitleText: red"
puts $f "Device: Postscript"
if { [info exists tchan_] } {
close $tchan_
}
exec rm -f temp.q temp.a
exec touch temp.a temp.q
exec awk $awkCode all.q
puts $f \"queue
```

```
exec cat temp.q >@ $f
puts $f \n\"ave_queue
exec cat temp.a >@ $f
close $f
# Acá se ejecuta xgraph y lleva como parámetro los archivos
# creados mediante el procesamiento anterior.
exec xgraph -bb -tk -x time -y queue temp.queue &
exit 0
}
$ns run
```