UNIVERSIDAD NACIONAL DE INGENIERÍA

FACULTAD DE INGENIERÍA ELÉCTRICA Y ELECTRÓNICA

PROTOCOLO DE TRANSMISIÓN DE CONTROL DE FLUJO (SCTP)

INFORME DE SUFICIENCIA

PARA OPTAR EL TÍTULO PROFESIONAL DE:

INGENIERO ELECTRÓNICO

PRESENTADO POR:

JORGE VÍCTOR EGOÁVIL RETAMOZO

PROMOCIÓN
2000- II

LIMA – PERÚ
2008

PROTOCOLO DE TRANSMISIÓN DE CONTROL DE FLUJO (SCTP)

Dedico este trabajo a:

A mi esposa, por su amor, su comprensión y apoyo incondicional,

a mi madre, por su cariño, sus consejos y su confianza

y a mi hija, por su cariño y su comprensión.

SUMARIO

El presente informe es una presentación del protocolo SCTP, el tercer protocolo de transporte oficial de la pila TCP/IP. SCTP se deriva del comité SIGTRAN (SIGnaling TRANsport, *Transporte de Señalización*). Este protocolo fue desarrollado en un principio para transportar mensajes de señalización telefónica digital SS7, conocida como 'Número 7', pero su diseño, el cual recoge las bondades y elimina muchos inconvenientes de los protocolos TCP y UDP, lo hace propicio para muchas otras aplicaciones, que demandan fiabilidad en las comunicaciones y al mismo tiempo exigen soporte de alto tráfico de información, como es el caso de la telefonía y otros servicios de transmisión de voz, audio y vídeo en tiempo real.

En el presente trabajo se hará una introducción al protocolo SCTP, las razones de su surgimiento y estandarización. Asimismo se darán detalles del protocolo, sus características, sus ventajas y limitaciones respecto de los protocolos TCP y UDP. Finalmente se brinda un alcance de las posibles mejoras que se lograrían en aplicaciones conocidas basadas en TCP y UDP si se empleara el protocolo SCTP.

ÍNDICE

PRÓLOGO		**1**
CAPÍTULO I		**2**
DESCRIPCIÓN GENERAL DEL PROTOCOLO		**2**
1.1.	Introducción	2
1.2.	Antecedentes	4
1.3.	Resumen	6
CAPÍTULO II		**7**
CARACTERÍSTICAS DEL PROTOCOLO SCTP		**7**
2.1.	Introducción	7
2.2.	Principales ventajas del protocolo SCTP	8
2.3.	Etapas del proceso de transmisión	9
2.4.	Estructura de los paquetes SCTP	10
2.5.	Descripción del proceso de transmisión	13
2.5.1.	Inicio de la asociación	13
2.5.2.	Transmisión de datos en la asociación	14
2.5.3.	Cierre de la asociación	15
CAPÍTULO III		**16**
APLICABILIDAD DEL PROTOCOLO SCTP		**16**
3.1.	SCTP en sustitución de TCP	16
3.2.	SCTP en sustitución de UDP	19
CONCLUSIONES		**20**
ANEXO A		**21**
LISTA DE SIGLAS Y TÉRMINOS		**22**
BIBLIOGRAFÍA		**25**

PRÓLOGO

El presente informe, desarrollado en base a una extensa búsqueda bibliográfica, tiene como objetivo general presentar al protocolo de transporte SCTP como una alternativa viable a los protocolos TCP y UDP, que predominan en las aplicaciones IP de hoy en día.

Este protocolo existe como estándar de Internet (RFC 2960) desde el año 2000, y desempeña un papel vital en las redes de telecomunicaciones. En los capítulos siguientes se verá que SCTP resuelve varios problemas inherentes a TCP, inclusive problemas de seguridad, y sin embargo, su existencia es desconocida por gran parte del mundo académico y profesional.

En el primer capítulo se narra los antecedentes del protocolo SCTP, la necesidad que lo hizo surgir, la decisión final de implementarlo como un protocolo de transporte sobre IP en lugar de desarrollarlo sobre UDP, sus características principales y una breve comparación con los protocolos TCP y UDP.

En el segundo capítulo se detalla los conceptos asociados al protocolo, como son las definiciones de punto terminal y asociación, las tramas básicas de los mensajes SCTP, el proceso de establecimiento de la conexión, la transferencia de los mensajes y el cierre de la conexión.

En el tercer capítulo se da una lista de posibles aplicaciones que podrían beneficiarse de una migración de TCP o UDP hacia SCTP.

CAPÍTULO I
DESCRIPCIÓN GENERAL DEL PROTOCOLO SCTP

1.1. Introducción

La gran mayoría de aplicaciones cliente/servidor, desde las más simples hasta las más complejas como un *cluster* Beowulf, hacen uso de TCP/IP para la comunicación de redes, debido a que la pila de protocolos TCP/IP es ampliamente soportada.

Un requisito para una aplicación como Beowulf es la transmisión confirmada de mensajes indivisibles. Sin embargo, casi la totalidad de implementaciones de TCP/IP ofrece sólo transporte no confirmado de mensajes indivisibles (UDP) y transporte confirmado de una secuencia de octetos (TCP).

Para adecuarse a lo que TCP/IP ofrece, cada protocolo de aplicación debe implementar su propio método de separación de mensajes sobre TCP, implementar control de errores sobre UDP o implementar algo que alivie dichos inconvenientes de acuerdo con la solución a desarrollar. Desarrollar un control de errores puede ser una tarea muy complicada, por lo que la mayoría de protocolos de capa superior usa TCP empleando separación de mensajes.

Los protocolos populares de Internet que operan sobre TCP (SMTP, FTP, HTTP) utilizan líneas de texto ASCII para la comunicación, y un fin de línea o retorno de carro como separadores de mensajes. Es más, en los últimos años se está utilizando XML sobre HTTP como método de llamada a funciones de servidor. Aunque el uso de líneas de texto es ineficiente comparado con estructuras binarias (más del doble de caracteres empleados para la transferencia de mensajes), facilita la separación de mensajes, lo que explica la adopción generalizada de TCP.

En los protocolos que usan UDP como transporte, predomina el uso de estructuras binarias extremadamente eficientes en tamaño y procesamiento (la mitad de caracteres de un mensaje TCP y ahorro de tiempo, al obviar el proceso de conversión tanto de binario a ASCII en el transmisor como de ASCII a binario en el receptor, sin olvidarnos de que estas conversiones se tienen que hacer en la capa de aplicación). Ejemplos: DNS, NFS, NTP. Es mucho más fácil tratar un mensaje UDP, pues es recibido exactamente como es transmitido. También usan UDP los protocolos de multimedia en tiempo real, asumiendo el control de errores y el reordenamiento de mensajes en el receptor. Las

aplicaciones que necesitan *broadcast* o *multicast* necesitan usar UDP, pues la pila TCP/IP soporta sólo difusión en UDP, la cual es difusión no confirmada, es decir, transporte no confirmado.

A pesar de estas ventajas de eficiencia y *broadcast*, el empleo de UDP ha ido disminuyendo, incluso en aplicaciones multimedia comerciales de tiempo real, que han migrado a TCP, no obstante sus desventajas en este tipo de aplicaciones. Este tipo de migración tendría como un motivo importante que al usar TCP se puede pasar por alto cortafuegos, *proxies* y ruteadores NAT. Por ejemplo, *RealPlayer*, una de las primera aplicaciones multimedia de tiempo real en Internet, comenzó usando UDP como transporte. Hoy, prácticamente cualquier contenido basado en *RealPlayer* es servido sobre TCP.

Otro factor importante para el abandono de UDP es la inexistencia de recursos de seguridad contra el secuestro de conexiones. Por tanto, una conexión UDP está sujeta al secuestro de conexiones y a la falsificación de paquetes. TCP es menos débil ante estos ataques.

SCTP es un protocolo relativamente nuevo, cuya principal característica es la transmisión de mensajes indivisibles confirmados. El propósito inicial de SCTP fue la integración de las pilas de redes TCP/IP y SS7 para el intercambio de mensajes de señalización telefónica. Esta señalización telefónica es un flujo de mensajes administrativos, como mensajes SMS, tarificación, servicios de usuario, señal de ocupado, etc., transmitidos por una red de conmutación de paquetes. La transmisión de voz hace uso de una red de conmutación de circuitos separada.

Esta función de transmitir señalización telefónica es bastante crítica, tanto para operadores como usuarios. Es en esta tarea en la que SCTP es ampliamente utilizado, y lo será más en la medida que las empresas telefónicas migren hacia redes de señalización basadas enteramente en TCP/IP.

Más aún, SCTP también tiene el potencial de ser usado fuera del ambiente de telecomunicaciones. Puede sustituir con ventajas a los protocolos clásicos de transporte (TCP y UDP) en buena parte de los protocolos de aplicación de Internet. Un estímulo adicional para el uso de SCTP es la presencia de características modernas de seguridad, como son las *cookies* de inicialización, la redundancia gracias al *multi-homing* y el control de errores incorporado para mensajes indivisibles.

1.2. Antecedentes

La cada vez creciente necesidad de integración entre las redes de telefonía y las redes de ordenadores da lugar a nuevas aplicaciones tales como *Voz sobre IP (Voice over IP, VoIP)* o la implantación de la *3ª Generación* de telefonía móvil.

El grupo de trabajo SIGTRAN de la IETF es el que se encarga de producir los estándares necesarios para hacer posible la integración de dichas redes. El propósito principal de este grupo de trabajo es encargarse del transporte de señalización de PSTN basadas en conmutación de paquetes sobre redes IP, teniendo en cuenta las funciones y prestaciones requeridas para el transporte de dicha señalización.

Uno de estos nuevos estándares surgidos del trabajo conjunto de múltiples ingenieros en SIGTRAN es el protocolo SCTP. SCTP es un nuevo protocolo de transporte fiable. El objetivo inicial de este nuevo protocolo era el transporte de los paquetes de señalización de redes SS7 sobre redes IP.

El diseño de SCTP se inició en 1998 por Randall R. Stewart y Qiaobing Xie, cuando crearon su protocolo MDTP, el cual se basaba en el protocolo TCP. Debe tenerse en cuenta que este protocolo se comenzó a diseñar antes siquiera de la existencia de SIGTRAN, y su objetivo original era subsanar algunos de los problemas encontrados al usar TCP. Tiempo después, al crearse SIGTRAN y comenzar a buscar el protocolo de transporte idóneo para sus propósitos, llegaron a la conclusión de que MDTP era lo más parecido a aquello que andaban buscando. Desde este momento el interés por MDTP subió, y su diseño comenzó a debatirse en la lista de distribución que con tal propósito SIGTRAN había abierto.

Durante su fase de diseño, MDTP sufrió muchos cambios, pues había que adaptarlo a las necesidades específicas de SIGTRAN, el transporte de señalización de las redes telefónicas, sobre todo de la red SS7. El diseño final de SCTP fue publicado en la RFC2960 a finales de octubre de 2000.

SCTP incluye muchas mejoras sobre TCP que lo hacen más apropiado que éste para el transporte de señalización, e incluso puede competir con él como protocolo de transporte fiable general en Internet.

SCTP tiene un mecanismo para establecer *asociaciones* (el equivalente a las *conexiones* de TCP) que le hace inmune al ataque por inundación de datagramas con la bandera de *SYN* fijada. SCTP utiliza un mecanismo de cuatro pasos en vez de los tres que usa TCP. Esto le permite a los servidores el autenticar la dirección IP fuente del datagrama que tiene la bandera *SYN* fijada antes de reservar ningún recurso y así imposibilitar este ataque.

En TCP sólo se pueden establecer conexiones de una dirección IP a otra dirección IP. Una conexión TCP se identifica por la dirección IP y puerto tanto del cliente como del servidor. Así si una máquina posee diferentes tarjetas de red con sus respectivas direcciones IP asociadas, no puede usar más que una de ellas para establecer una conexión TCP con otra máquina. En SCTP, una asociación se identifica por una serie de direcciones IP y un puerto del cliente, y el conjunto de direcciones IP del servidor y su puerto. De esta manera, en caso de que una de las direcciones IP deje de funcionar, siempre se puede seguir utilizando cualquiera de las otras.

Otra innovación frente a TCP es que SCTP puede evitar el bloqueo *Head-Of-Line Blocking* mediante el uso de *flujos (streams)*. Este bloqueo se da cuando en TCP enviamos varios mensajes independientes troceados en datagramas usando una única conexión. En esta situación, aunque un mensaje haya llegado completamente al receptor, éste no se podrá pasar al usuario antes de que todos los mensajes anteriores hayan llegado también completos. SCTP permite el uso de flujos, que son subconexiones dentro de una asociación SCTP de manera que datagramas dirigidos a flujos distintos se tratan independientemente. Además, con SCTP podemos diferenciar distintos mensajes dentro del flujo de bytes con lo que el usuario no debe incluir sus propias marcas. Incluso se pueden enviar mensajes de forma que el receptor los pase al usuario nada más recibirlos, sin guardar el orden en que fueron enviados.

SCTP puede utilizar varias direcciones IP *(multihoming)* tanto en el cliente como en el servidor. Sin embargo, se utiliza tan sólo una de ellas para enviar los datos, la *dirección primaria (primary address)*. El resto se reserva y sólo se utilizan en caso de que la dirección primaria falle. Por ello, para saber el estado en que se encuentran dichas direcciones IP de reserva, SCTP tiene el llamado *mecanismo de latidos de corazón (heartbeat mechanism)*. Consiste en enviar mensajes a las direcciones IP que no se usan para enviar datos. Dichos mensajes, o *latidos*, se deben responder, de manera que al recibir la respuesta se sabe que esas direcciones siguen activas.

Uno de los principales problemas de TCP es que es muy difícil de extender. Cuando se quiere añadir una nueva característica a TCP, el limitado espacio que se dejó reservado para uso futuro cuando TCP se diseñó hace muchas veces que esto no sea posible. SCTP es un protocolo muy abierto que ha sido diseñado para que sea extensible por naturaleza. SCTP contiene una serie de funciones básicas, y ha sido pensado para que toda aquella característica adicional que quiera ser añadida en el futuro, pueda incluirse con gran facilidad. Un terminal que tiene una asociación SCTP con otro, puede enviarle mensajes de error, de manera que ciertos errores a nivel del protocolo de transporte pueden resolverse sin afectar al usuario. Estos mensajes de error sirven

también para negociar el uso de funciones opcionales, de manera que versiones antiguas de SCTP que no soporten dicha función nueva tengan una manera de expresar dicha carencia enviando el mensaje de error apropiado.

TCP ha sido el protocolo de transporte fiable por excelencia de las últimas dos décadas. Es por ello que muchas de las características que tiene SCTP han sido tomadas directamente de TCP. La mayoría de las extensiones que se han escrito para TCP han sido incluidas en SCTP en su versión básica. Entre ellas podemos mencionar el uso de *asentimientos selectivos (selective acknowledgements)*, la posibilidad de alertar de la recepción de *datagramas duplicados*, o el soporte para la *Notificación Explícita de Congestión (Explicit Congestion Notification, ECN)*. Además, SCTP usa los mismos algoritmos que TCP para evitar la congestión. De esta manera, cuando haya convivencia entre aplicaciones que usen bien SCTP o TCP como su protocolo de transporte, el ancho de banda adjudicado a una asociación SCTP o una conexión TCP sea el mismo.

1.3. Resumen

El Protocolo de Transmisión de Flujo (SCTP) es un protocolo punto a punto, orientado a la conexión, que transporta datos en secuencias de flujos independientes. Los nodos ('*endpoints*') SCTP soportan '*multi-homing*' (múltiples direcciones); por consiguiente, existe una interface de redundancia incluida en el protocolo. A través de mecanismos selectivos de transmisión, SCTP resuelve errores y almacena temporalmente los datos (sirve de *buffer*) en el proceso de transmisión.

SCTP proporciona a las aplicaciones mejoras en rendimiento, confiabilidad y funciones de control. Este protocolo es esencial cuando se trata de detectar fallas en la conexión y de monitorear las comunicaciones. Más aún, SCTP podría ser implementado en sistemas de redes y aplicaciones que entregan voz/datos y que soportan servicios de calidad en tiempo real, por ejemplo, vídeo en streaming y multimedia.

CAPÍTULO II
CARACTERÍSTICAS DEL PROTOCOLO SCTP

2.1. Introducción

Como se ve en la *Figura 2.1*, la capa de transporte de SCTP está ubicada entre la capa de aplicación y la capa de red. Puesto que SCTP está diseñado para ser interface entre dos nodos SCTP, existen ciertas APIs que se encuentran entre la capa de transporte y la capa de aplicación de SCTP. Además, cada nodo posee múltiples direcciones IP.

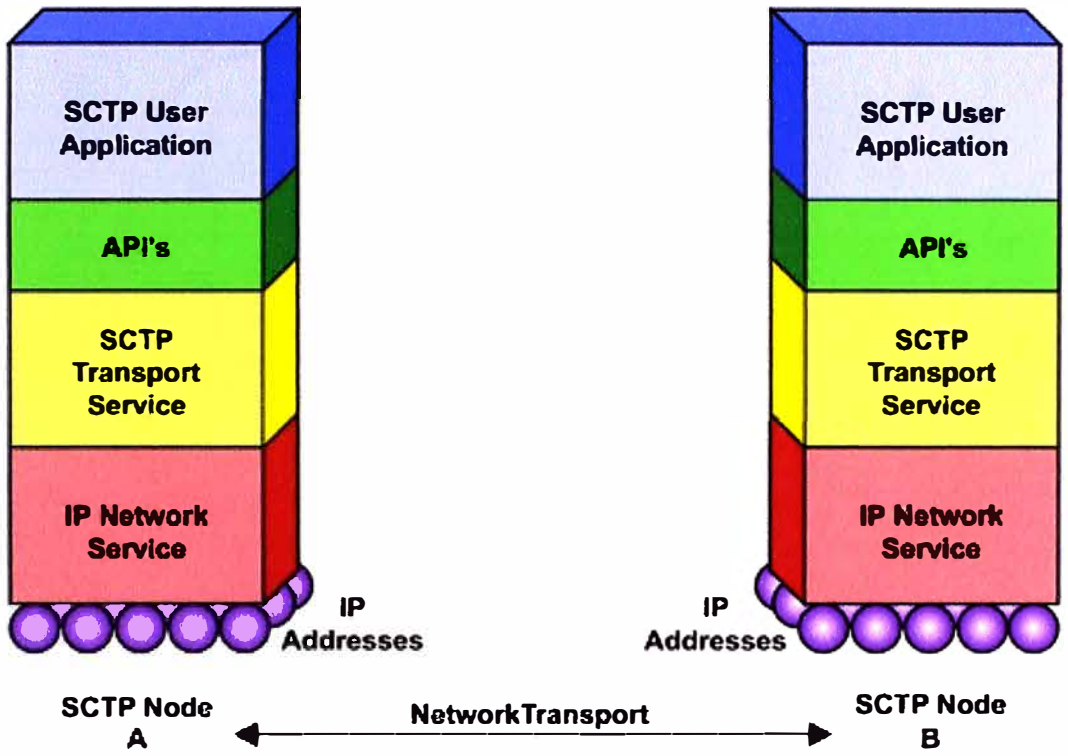

Fig. 2.1. Capas de servicio del protocolo SCTP **[1]**

SCTP puede utilizar múltiples caminos y flujos para transportar mensajes entre dos nodos. En SCTP, los datos se transmiten entre nodos a través de una conexión referida como "asociación". Una asociación comienza con una "iniciación" y se mantiene hasta que todos los datos hayan sido exitosamente transmitidos y recibidos. Una vez que todos los datos han sido recibidos, la asociación es terminada de manera no abrupta mediante un "cierre" (*shutdown*).

Dentro del protocolo SCTP, los datos de usuario y la información de control son ensamblados en 'bloques' (*chunks*). Múltiples bloques y una cabecera común conforman una unidad de datos del protocolo (PDU), conocida también como "paquete SCTP". Los paquetes SCTP contienen bloques de datos y bloques de control. SCTP proporciona entrega de mensajes ordenados dentro de "flujos SCTP" y posee tolerancia a fallas en la red en entornos multidirecciones.

Las propiedades fundamentales de SCTP son las siguientes:

- **Mecanismos de validación y reconocimiento** — Durante la iniciación, el mecanismo de validación empaqueta los datos en un "cookie" que incluye un *hash* seguro de valores y una clave secreta. Los cookies están firmados digitalmente con códigos de autenticación de mensajes (MAC), los cuales pueden usarse para prevenir ataques de denegación de servicio. Dentro de una asociación, los bloques del reconocimiento selectivo (SACK) acusan el recibo de los bloques de datos. Los bloques SACK se usan también para informar al nodo de bloques duplicados o perdidos.
- **Selección y Monitoreo de Rutas** — Los paquetes SCTP son enrutados a la dirección IP destino de un nodo mediante un "camino primario". El camino primario permite al usuario determinar la ruta primaria para el flujo de datos. Además, existen caminos alternativos para cada dirección IP que soporte el nodo destino. SCTP monitorea de cerca los caminos de transmisión hacia el nodo destino empleando bloques *HEARTBEAT* que prueban la conectividad de un camino. En SCTP, un camino es considerado "activo" si ha sido reconocido por el nodo destino o ha sido previamente usado para transferencia de paquetes SCTP. Un camino es considerado "inactivo" si previas transmisiones han fallado.
- **Control de Flujo y Congestión** — Mientras el control del flujo en SCTP se hace por cada asociación, el control de congestión se establece en cada camino de transmisión. El nodo destino asigna una variable de ventana de receptor para el control de flujo. Esta variable alerta al nodo origen de la cantidad de espacio disponible en el buffer de entrada del nodo destino. SCTP realiza control de congestión en cada flujo usando una variable de ventana de congestión. Esta variable limita el número de bytes que se puede enviar antes que se reciba un reconocimiento. Un conjunto de parámetros de control de flujo y de congestión es sutilmente retenido en la asociación y en cada camino de transmisión.

2.2 . Principales ventajas del protocolo SCTP

SCTP le lleva ventaja a TCP por sus características únicas. Esta sección explora

cómo el multi-homing, multi-streaming, y otras características de SCTP contribuyen a que el uso de SCTP sea una ventaja.

2.2.1. SCTP Multi-homing

La característica multi-homing permite a los nodos SCTP soportar múltiples direcciones IP. Multi-homing protege una asociación de fallas potenciales de red redirigiendo el tráfico a direcciones IP alternas. Durante la iniciación de una asociación, los nodos SCTP intercambian listas de direcciones IP. En consecuencia, cada nodo puede enviar y recibir mensajes desde cualquiera de las direcciones IP listadas en el nodo remoto. Por ejemplo, una de las direcciones IP listadas será designada como la dirección primaria durante la iniciación. Pero si la dirección primaria pierde repetidamente bloques, todos los bloques serán transmitidos a una dirección alterna hasta que la conexión a la dirección primaria sea reestablecida.

Multi-homing es un paso adelante sobre sesiones de intercambio de datos 'single-home' (dirección única) como TCP. En entornos de dirección única, la pérdida de la sesión podría darse por fallas en el núcleo de la red o por aislamiento de nodos. Como multi-homing dirige el tráfico en diferentes caminos hacia direcciones IP separadas, una pérdida de sesión debido a fallas físicas en la red es virtualmente inexistente en SCTP.

2.2.2. SCTP Multi-streaming

La característica multi-streaming separa y transmite los datos de usuario en múltiples flujos (*streams*) SCTP. Estos flujos son capaces de entrega independiente y en secuencia. La pérdida de mensaje en un flujo en particular sólo obstaculizará la entrega de ese flujo. Por tanto, los otros flujos dentro de la asociación no son afectados.

Mediante el multi-streaming, SCTP elimina bloqueos innecesarios que ocurren a menudo en transmisiones TCP. En TCP, un flujo está definido como una secuencia de bytes que siguen una estricta secuencia de entrega. Esta entrega secuencial resulta en un gran lastre conocido como "bloqueo del inicio de la fila" ("*head-of-line blocking*") donde los mensajes de un flujo no pueden saltarse unos a otros. Como los flujos SCTP son mensajes independientes, aquellos mensajes que son retransmitidos y los de alta prioridad pueden saltarse otros mensajes menos significativos.

2.3 . Etapas del proceso de transmisión de SCTP

En las tres etapas de la asociación, SCTP aplica mecanismos que lo diferencian de TCP y UDP:

a) **Iniciación** — En contraste con el saludo (*handshake*) de tres mensajes de TCP, SCTP usa un saludo de cuatro mensajes para iniciar una asociación. Este saludo de cuatro mensajes protege de intentos de denegación de servicio causados por atacantes que bombardeen los nodos SCTP con PDUs adulterados. Adicionalmente, los paquetes SCTP que contienen *tags* de verificación inválidos son identificados durante la iniciación y eliminados del camino de transmisión. Los valores del tag de verificación y el mecanismo de cookie blindan el procedimiento de iniciación de ataques tipo SYN (conocidos comúnmente como ataques ciegos) que son lugar común en TCP.

b) **Transmisión de Datos** — Durante la transmisión de los datos, la característica de empaquetamiento de los bloques permite a los bloques de datos estar multiplexados con bloques de control. El nodo destino reconoce la recepción de un bloque de datos enviando un bloque SACK. Los bloques SACK contienen Números de Secuencia de Transmisión (TSN) que revelan cualquier vacío en la secuencia de bloques de datos. Dentro de cada flujo, a los paquetes SCTP también se les asigna Números de Secuencia de Flujo (SSN). El SSN determina la secuencia de entrega de datos dentro de cada flujo independiente. Si el nodo destino indica vacíos en el SSN, entonces el mensaje no será entregado hasta que el vacío sea llenado.

c) **Cierre** — El procedimiento de cierre en SCTP tiene ventajas significativas sobre TCP. Por ejemplo, una conexión TCP es considerada "semiabierta" cuando un nodo continúa enviando datos aun cuando el nodo destino no esté transmitiendo más datos. Por el contrario, SCTP implementa un cierre no abrupto de una asociación mediante el intercambio de tres mensajes. Estos mensajes indican que ambos nodos cesarán de transmitir datos.

2.4 . Estructura de los paquetes SCTP

La *Figura 2.2* muestra el detalle de un paquete SCTP. La cabecera común incluye lo siguiente:

- La dirección del puerto origen.
- La dirección del puerto destino.
- El *tag* de verificación.
- El *checksum* de todo el paquete.

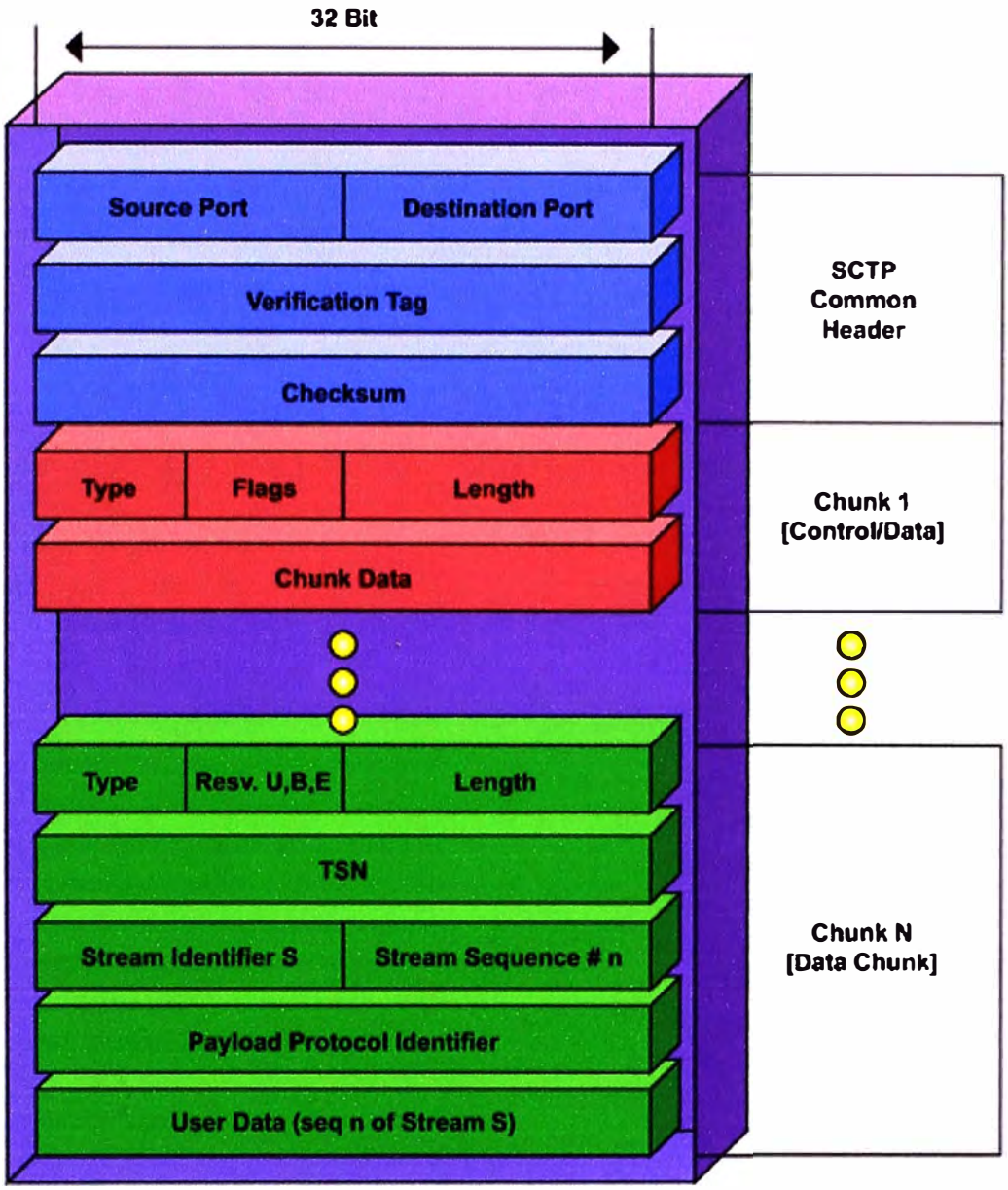

Fig. 2.2. Formato de paquete SCTP **[2]**

El número de puerto origen es usado por el nodo receptor para identificar la asociación a la cual el paquete SCTP pertenece. El número de puerto destino es la dirección del receptor SCTP para la cual el paquete está destinado. Cada nodo asigna un tag de verificación (valor de 32 bits) que identifica la asociación. El checksum actúa como una herramienta de integridad de datos para cada paquete SCTP.

Los campos de un bloque se describen como sigue:

- El tipo de bloque: Identifica el tipo de bloque que se transmite.
- El flag del bloque: Especifica cuáles bits serán usados en la asociación.
- La longitud del bloque: Determina el tamaño de todo el bloque en bytes.
- Los datos del bloque: Lleva la carga real de los datos.

TABLA Nº 2.1: Tipos de Bloques de Control y sus Definiciones por Función **[3]**

BLOQUE	DEFINICIÓN
Iniciación INIT	El bloque INIT es enviado para iniciar una asociación SCTP entre dos nodos.
Acuse de Iniciación INIT ACK	El bloque INIT ACK acusa la recepción de un bloque INIT. La recepción de un bloque INIT ACK establece una asociación.
Acuse selectivo SACK	Los bloques SACK acusan el recibo de bloques DATA.
Eco de Cookie COOKIE ECHO	El bloque COOKIE ECHO es usado extensivamente durante el proceso de iniciación y es enviado al nodo destino.
Acuse de Cookie COOKIE ACK	El bloque COOKIE ACK acusa recibo de un bloque COOKIE ECHO. El bloque COOKIE ACK debe tener precedencia sobre cualquier bloque DATA o SACK enviado en la asociación. El bloque COOKIE ACK puede estar empaquetado con bloques DATA o SACK.
Solicitud de Heartbeat HEARTBEAT	Los bloques HEARTBEAT son enviados desde un nodo SCTP a su par destino para probar la conectividad de una dirección de destino específica en la asociación.
Acuse de Heartbeat HEARTBEAT ACK	Cada vez que un bloque HEARTBEAT es recibido por un nodo, se envía un bloque HEARTBEAT a la dirección IP origen para acusar recibo del bloque HEARTBEAT.
Abortar Asociación ABORT	El bloque ABORT es una indicación al nodo destino para cerrar la asociación. Adicionalmente, el bloque ABORT informa al receptor la razón para abortar la asociación en los parámetros de causa.
Error de Operación ERROR	El bloque ERROR es enviado al nodo destino para reporta ciertas condiciones de error que puedan existir. El bloque ERROR puede contener parámetros que determinen en tipo de error que haya tenido lugar.
Cierre de Asociación SHUTDOWN	El bloque SHUTDOWN acitva un cierre no abrupto de una asociación con un nodo destino.
Acuse de Cierre SHUTDOWN ACK	Un bloque SHUTDOWN ACK se usa para acusar el recibo del bloque SHUTDOWN al final del proceso de cierre.
Cierre Completo SHUTDOWN COMPLETE	El bloque SHUTDOWN COMPLETE concluye el procedimiento de cierre.

Como se indica en la *Figura 2.2*, existen N bloques (número de bloques) indicados en un único paquete SCTP. El número N se determina por el tamaño de la unidad máxima de transmisión (MTU) del camino de transmisión. SCTP permite a los bloques ser multiplexados en un paquete para llenar la capacidad MTU, con excepción de los bloques de iniciación (INIT) y de reconocimiento de iniciación (INIT ACK). Existen 14 tipos de bloques en total, incluyendo un bloque DATA y 13 tipos de bloques de control. El bloque DATA contiene la carga real de los datos. La definición y los parámetros de los bloques de control están resumidos en la *Tabla Nº 2.1*.

2.5 . Descripción del proceso de transmisión

Esta sección describe el proceso de transporte de datos en las tres fases principales de una asociación SCTP: iniciación, transmisión de datos y cierre. El nodo que inicia la asociación será denominado "Nodo A"; el nodo destino que recibe las solicitudes de establecimiento de asociación será denominado "Nodo B".

Nota: Los bloques ABORT y ERROR pueden ser generados por cualquiera de los nodos y pueden ser enviados en cualquier momento durante la asociación. En estos dos casos, los nodos experimentan un cierre inmediato.

2.5.1. Inicio de la asociación

En la Fig. 2.3 se puede observar lo siguiente:

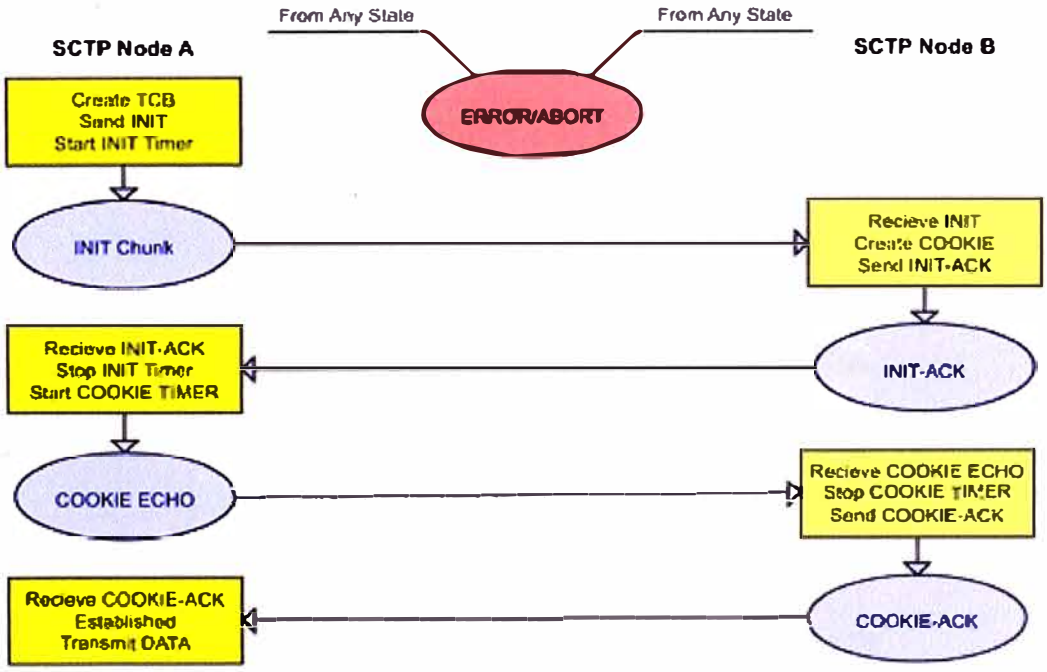

Fig. 2.3. El saludo de cuatro mensajes en la Iniciación de una Asociación SCTP **[4]**

a) El Nodo A genera un bloque INIT y lo envía al Nodo B. El Nodo A inicia el temporizador INIT.
b) Si el Nodo B desea aceptar la asociación, genera un bloque INIT ACK que incluye un cookie. Luego envía el bloque INIT ACK, junto con un cookie, de regreso hacia el Nodo A.
c) El Nodo A recibe el bloque INIT ACK y detiene el temporizador INIT. El Nodo A genera un bloque COOKIE ECHO, que es enviado al Nodo B. El Nodo A inicia el temporizador del cookie. Los bloques DATA pueden también ser empaquetados en este paquete.
d) El Nodo B verifica la validez del cookie. Luego de la validación envía un bloque COOKIE ACK de vuelta al Nodo A.
e) El Nodo A recibe el bloque COOKIE ACK e ingresa en la siguiente fase de transmisión de datos.

2.5.2. Transmisión de datos en la asociación

El Nodo A y el Nodo B transmiten datos en forma transparente. A través del, proceso de transmisión de datos, los bloques HEARTBEAT and HEARTBEAT ACK se intercambian entre los nodos en intervalos regulares de tiempo controlados por el temporizador de heartbeat. Estos bloqueos prueban la conectividad de los nodos, preservando de esa manera la transmisión de datos.

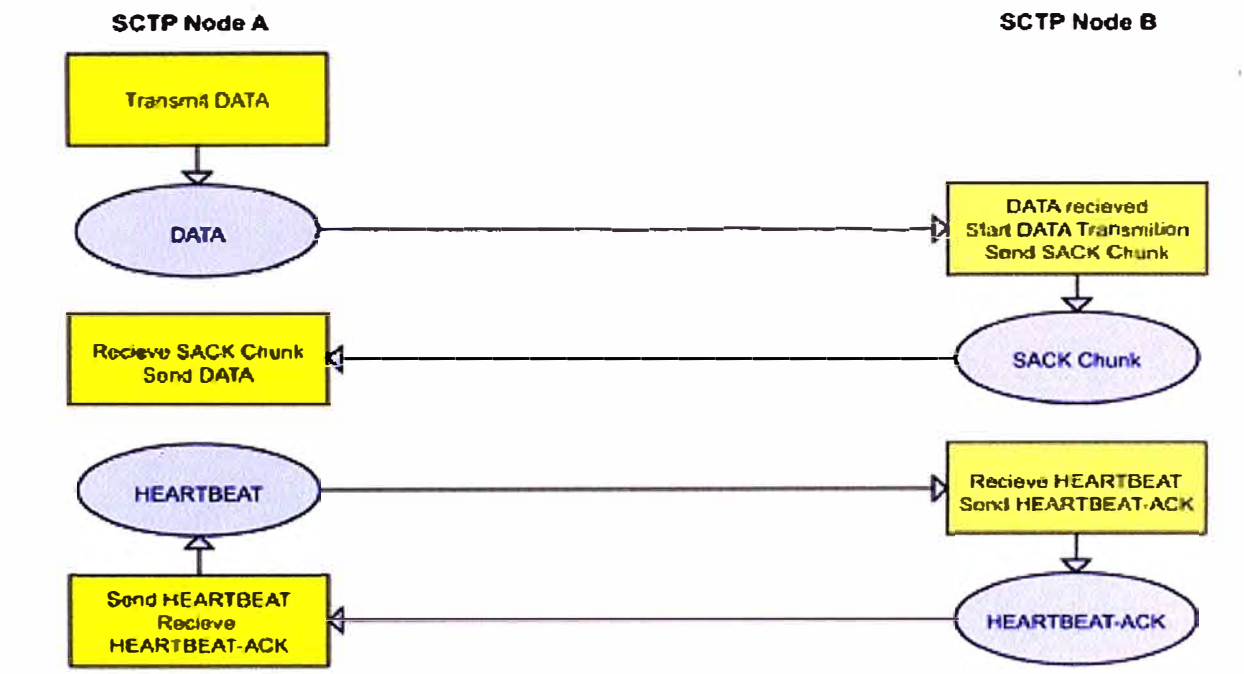

Fig. 2.4. Rastreando el Proceso de Transmisión de Datos en SCTP **[5]**

a) El Nodo A y el Nodo B intercambian bloques DATA.
b) Después de que cada bloque DATA es recibido, el nodo receptor devuelve un bloque SACK para acusar la recepción.
c) Los datos son transmitidos hasta que uno de los nodos decide cerrar la asociación enviando un bloque SHUTDOWN dentro de uno de los paquetes.

2.5.3. Cierre de la asociación

El bloque SHUTDOWN puede ser enviado por cualquiera de los nodos. Por razones ilustrativas, se considera aquí al Nodo A como el iniciador del proceso de Cierre.

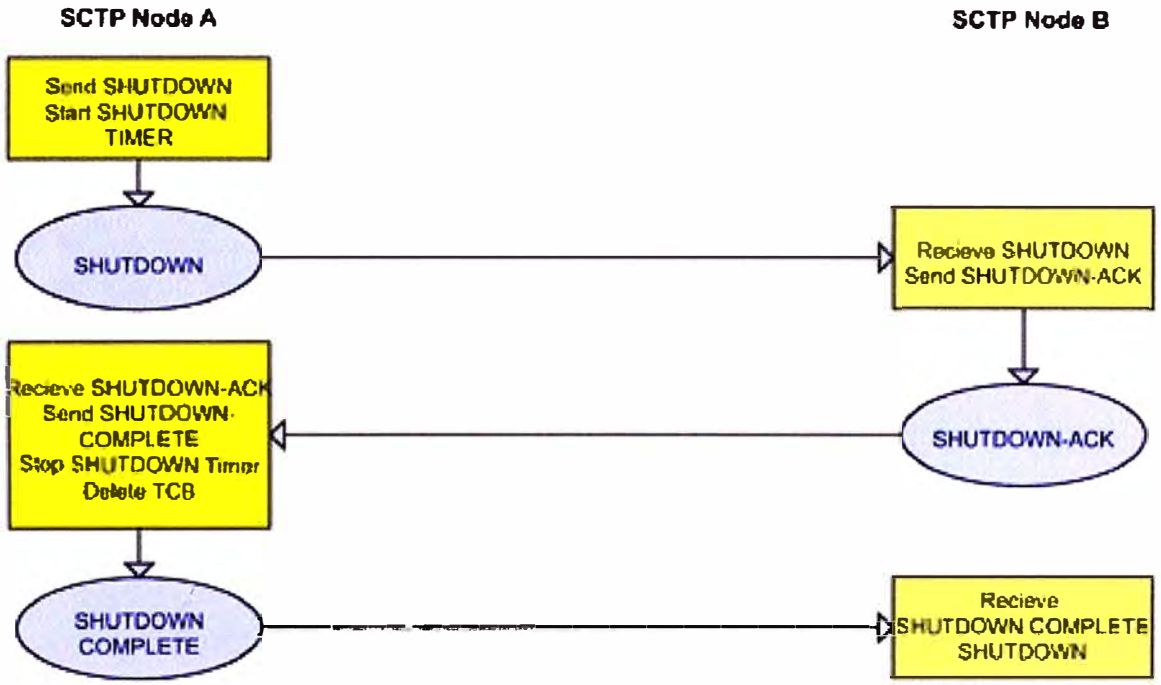

Fig. 2.5. Terminación No Abrupta de una Asociación SCTP **[6]**

a) El Nodo A envía un bloque SHUTDOWN al Nodo B e inicia el temporizador de SHUTDOWN.
b) El Nodo B acusa la recepción del bloque SHUTDOWN generando un bloque SHUTDOWN ACK, el cual es enviado al Nodo A.
c) El Nodo A recibe el bloque SHUTDOWN ACK y responde deteniendo el temporizador SHUTDOWN. Luego, el Nodo A genera un bloque SHUTDOWN COMPLETE y envía este bloque al Nodo B.

CAPÍTULO III
APLICABILIDAD DEL PROTOCOLO SCTP

3.1. SCTP en sustitución de TCP

3.1.1. Costos y beneficios

SCTP puede ser visto como una evolución del protocolo TCP. Cualquier aplicación que utilice TCP puede usar también SCTP, con resultados iguales e incluso mejores. Esta posibilidad es tan concreta que los puertos de servicios TCP conocidos y registrados en IANA (por ejemplo, 80 para HTTP, 21 para FTP, etc.) están ya reservadas en SCTP para las mismas aplicaciones.

Con sólo migrar de TCP a SCTP, que no es otra cosa que mudar la capa de transporte, la aplicación adquiere de inmediato las ventajas del *checksum* CRC-32c (resistencia a corrupción de datos) y del recurso de multicaminos (tolerancia a fallas de red). SCTP además es más firme que TCP manteniendo activa la conexión (asociación en SCTP).

Las transmisiones con SCTP tienen ventaja sobre cualquier protocolo de aplicación que haga uso de varias conexiones TCP en paralelo para realizar una única tarea. El empleo de una única asociación economiza recursos y facilita el tratamiento en ruteadores NAT.

La atomicidad de mensajes trae beneficios para los protocolos donde los mensajes encajan dentro de una estructura de lenguaje C y/o de tamaño previsible, pues elimina completamente la necesidad de separación de mensajes en la capa de aplicación.

En cada mensaje SCTP es posible transmitir un identificador de protocolo de 32 bits. Este identificador facilita la tarea de entregar eficientemente el mensaje al receptor.

Los mensajes de tamaño grande o no previsible, no deben transmitirse atómicamente, pues existen limitaciones en el tamaño del mensaje SCTP dependiendo de la implementación. Este tipo de mensajes debe seguir siendo tratado en la capa de aplicación.

3.1.2. Implementación en HTTP

En una página web típica, una página HTML contiene, además de texto, un número alto de elementos gráficos. Cada uno de estos elementos representa un archivo en el servidor, el cual debe ser solicitado individualmente por el cliente.

Existen dos métodos para solicitar todos estos elementos. La forma primitiva, soportada por todo cliente y servidor HTTP, es establecer una conexión TCP/IP por cada elemento. Este método simple tiene sus desventajas:

- Consume muchos recursos en el cliente, y en particular en el servidor debido al gran número de conexiones abiertas. Como el número de conexiones TCP simultáneas está limitado por el *kernel*, el servidor HTTP puede fácilmente ser saturado.
- El tráfico generado es desproporcionado en relación a otros servicios, porque abre múltiples conexiones para una sola tarea e indirectamente utiliza mayor ancho de banda para ella.
- Casi todos los clientes HTTP (navegadores, *proxy*, etc.) limitan el número de conexiones simultáneas a un mismo servidor, abriendo una conexión cuando se cierra otra. Si bien alivia de problemas al servidor, añade complejidad en los clientes.

Se puede utilizar una extensión de HTTP que permite secuenciar varios archivos a través de una única conexión TCP/IP. Esto resuelve efectivamente el exceso de conexiones, pero:

- Añade complejidad a las aplicaciones, que deben interpretar la secuencia de caracteres y separar las solicitudes de archivos, así como tratar la ocurrencia de errores u omisiones.
- Potencializa el *Head Of Line Blocking* (bloqueo de cabeza de línea). Por ejemplo, si el servidor entrega diez archivos, y un paquete IP del primer archivo se pierde, la entrega de los demás archivos será postergada hasta que el paquete perdido sea retransmitido.

Todos estos problemas son resueltos utilizando el protocolo SCTP:

- Cada flujo de transmisión contiene una solicitud, y cada flujo de recepción contiene un archivo.
- El retraso de un flujo, causado por la pérdida de un paquete IP, no retrasa la entrega de los demás flujos.
- El hecho de que cada solicitud use su propio flujo permite simplificar las aplicaciones, en las cuales no es necesario separar múltiples solicitudes.

- La sobrecarga de terminales se reduce porque sólo es necesaria una asociación entre cliente y servidor.

A pesar de estas ventajas en eficiencia de uso de recursos, en HTTP no se aprovecharía la característica de atomicidad de mensajes, por la naturaleza de las páginas web, en las que por cada elemento se realiza una solicitud.

3.1.3. Sistemas de archivos distribuidos

Prácticamente todos los sistemas de archivos distribuidos similares al NFS en Unix/Linux se podrían beneficiar de la atomicidad de los mensajes. La atomicidad elimina o por lo menos simplifica la decodificación y agiliza el procesamiento.

Si el sistema de archivos está operando en modo asíncrono, todos los mensajes pueden ser rotulados como 'urgentes' para evitar un bloqueo del primero en la cola (bloqueo HOL – *Head Of Line Blocking, HOLB*).

3.1.4. Multimedia vía Internet

Algunos servicios multimedia y de tiempo real usan HTTP sobre TCP como medio de transporte. Aunque es bastante inadecuada, esta modalidad es ampliamente utilizada por pasar fácilmente por ruteadores NAT, *proxies* y cortafuegos (*firewalls*).

Multimedia sobre TCP funciona en la práctica, mientras exista holgura de ancho de banda y baja pérdida de paquetes. Pero cuando ocurre un congestionamiento muy severo, la resincronización se torna imposible. Esto es particularmente cierto en aplicaciones multimedia de tiempo real, donde la generación de datos no se detiene y los datos retransmitidos simplemente son descartados por el receptor, ya que pierden su vigencia.

El uso de SCTP en lugar de TCP, aunque también inadecuado para este fin por ser transmisión confirmada, traería algunas ventajas marginales:

- El flujo multimedia podría ser enviado enteramente como mensajes urgentes, que pueden entregarse en desorden, evitando el bloqueo HOL (siempre que el protocolo de aplicación sea capaz de manejar flujos en desorden).
- La notificación de SCTP acerca de paquetes perdidos es más elaborada, posibilitando una economía de retransmisiones.

SCTP se torna ideal para multimedia en tiempo real, mejor que TCP y UDP, cuando se implementa la extensión de confiabilidad parcial (PR-SCTP).

3.2. SCTP en sustitución de UDP

Muchos libros y artículos distinguen UDP de TCP clasificando a UDP como protocolo "sin conexión". En realidad, se puede hablar de conexiones UDP identificadas por direcciones y números de puertos; de lo contrario, no podrían ser tratadas por ruteadores NAT; lo que pasa es que UDP delega a la capa superior de aplicación la responsabilidad de la administración de las conexiones.

En aplicaciones como DNS, que emplean mensajes cortos, no sería una ventaja migrar de UDP a SCTP. Lo mismo ocurre para aplicaciones que usan bastante el *broadcast* y el *multicast*.

Con la extensión de confiabilidad parcial PR-SCTP, se crea una oportunidad para sustituir UDP por SCTP en aplicaciones que creen conexiones de larga duración y alto tráfico de datos, como por ejemplo multimedia sobre Internet o telefonía sobre IP. En estas aplicaciones, el tráfico de señalización puede viajar en la misma asociación que el tráfico de contenido. Esto facilitaría la labor de un ruteador NAT, el cual sólo necesita tratar una asociación en particular, mientras que, por ejemplo, para el caso de H.323, el ruteador NAT necesita discernir qué flujo de UDP está relacionado con qué flujo de TCP, lo cual requiere análisis de tráfico en la capa de aplicación, lo cual viola la arquitectura de capas y es muy ineficiente.

CONCLUSIONES

1. El protocolo SCTP se muestra como una alternativa viable para sustituir aplicaciones basadas en TCP o UDP que manejan alto tráfico de datos y con baja tolerancia a errores.
2. La creación del protocolo SCTP supuso la generación de conceptos que eran necesarios en aplicaciones modernas, como resistencia al bloqueo del primero en la fila (*Head Of Line Blocking*), redundancia y tolerancia a fallos mediante el *multi-homing*, que significa darle a un terminal más de una dirección IP con la intención de poder enlazarse fácilmente hacia otros terminales. Esto resultaría muy útil y eficiente en servidores con carga balanceada de alta disponibilidad.
3. Como en gran parte de casos a nivel científico y tecnológico, la solución al problema particular de transportar señalización telefónica SS7 sobre redes IP, devino en un protocolo estándar que puede ser utilizado en bastantes aplicaciones.
4. El hecho de tener un protocolo de transporte fiable en redes IP, acrecienta aún más la vigencia de este tipo de redes y demuestra su versatilidad, aún en su versión IPv4.

ANEXO A
LISTA DE SIGLAS Y TÉRMINOS

LISTA DE SIGLAS Y TÉRMINOS

API: *Application Programming Interface*. Conjunto de funciones que un aplicativo debe usar para comunicarse con una biblioteca o con un sistema operativo.

Checksum: Suma de verificación empleada como forma de control de redundancia. Es una medida muy simple para proteger la integridad de datos, verificando que no hayan sido corrompidos.

CRC: *Cyclic Redundancy Check*. Algoritmo de suma de verificación basado en división de polinomios.

CRC-32: Versión de CRC que genera suma de 32 bits.

CRC-32c: Versión de CRC que genera suma de 32 bits, con polinomio divisor diferente de CRC-32.

DNS: *Domain Name System*. Sistema de Nombres de Dominios.

FTP: *File Transfer Protocol*. Protocolo de Transferencia de Archivos.

H.323: Conjunto de protocolos para transmisión comprimida de voz y vídeo, usado por algunos sistemas de VoIP.

Hash: Función o método para generar claves o llaves que representen de manera unívoca a un documento, registro, archivo, etc., resumiendo o identificando un dato a través de la probabilidad mediante una *función hash* o *algoritmo hash*. Un *hash* es el resultado de dicha función o algoritmo.

Heartbeat: En SCTP, mecanismo de envío y espera de respuesta para identificar el estado de un nodo. Si el nodo responde, estará activo y disponible para recepción de mensajes.

Head-Of-Line (HOL) Blocking: Problema de algunos protocolos con confirmación que causa retraso en la entrega de los datos.

HTTP: *HyperText Transfer Protocol*. Protocolo de Transferencia de Hipertexto.

IETF: *Internet Engineering Task Force*. Fuerza de Trabajo de Ingeniería de Internet.

IP: *Internet Protocol*. Protocolo de red de la pila TCP/IP.

IPv4: *Internet Protocol version 4*. Versión del protocolo IP.

Kernel: Núcleo del sistema operativo.

LDAP: *Lightweight Directory Access Protocol*. Protocolo abierto de acceso a directorios de redes que contienen información de usuarios, grupos, políticas y directivas de red.

MDTP: *Multi-Network Datagram Transmission Protocol*. Protocolo de Transmisión de Datagramas Multi-Red.

NAT: *Network Address Translation*. Sistema de conversión de direcciones IP para redireccionar tramas de datos de una red a otra.

NFS: *Network File System*. Sistema de gestión de archivos de red usado frecuentemente en Unix/Linux.

NTP: *Network Time Protocol*. Protocolo de tiempo de red, empleado para sincronizar el reloj de un terminal con el de otro terminal.

PR-SCTP: *Partial Reliability SCTP*. Extensión de SCTP que dismimuye parcialmente la confiabilidad de los mensajes.

Proxy: Servicio de acceso a Internet y de cache de páginas web para un entorno de red.

PSTN: *Public Switched Telephone Networks*. 'Redes Públicas Telefónicas Conmutadas', denominación de la red telefónica pública.

RFC: *Request For Comments*. Solicitud de Comentarios.

RTO: *Retransmission TimeOut*. Tiempo que el transmisor espera para recibir una confirmación del receptor. Pasado ese tiempo, el paquete se considera perdido.

SCTP: *Stream Control Transmission Protocol*. Protocolo de Transmisión con Control de Flujo.

SIGTRAN: *SIGnaling TRANsport*. Transporte de Señalización.

Socket: En UNIX, un manejador de archivo que corresponde a una conexión de red.

SS7: *Signaling System #7*. Pila de red de conmutación de paquetes usada en telefonía.

Streaming: Se refiere a ver u oír un archivo directamente en una página web sin necesidad de descargarlo antes al ordenador o computador.

Tag: Etiqueta de identificación.

TCP: *Transmission Control Protocol*. Protocolo de control de transmisión.

TCP/IP: *Transmission Control Protocol / Internet Protocol*. Siglas de la pila de protocolos de Internet ampliamente conocida.

UDP: *User Datagram Protocol*. Protocolo de datagrama de usuario.

UNIX: Familia de sistemas operativos.

XML: *eXtensible Markup Language*. Lenguaje extensible de marcadores, usado como estándar de mensajes y documentos.

BIBLIOGRAFÍA

1. ARIAS-RODRIGUEZ, Iván. "Stream Control Transmission Protocol – The design of a new reliable transport protocol for IP networks". Helsinki University of Technology, Electrical and Communications Engineering Department. Networking Laboratory - 2002/02/12.

2. STEWART, Randall R. y XIE, Qiaobing. "Stream Control Transmission Protocol (SCTP)". Addison-Wesley, 2002.

3. PFÜTZENREUTER, Elvis. "Aplicabilidade e desempenho do protocolo de transporte SCTP". Universidade Federal de Santa Catarina, Programa de Pós-Graduação em Ciência da Computação, Dic. 2004.

REFERENCIAS

[1] International Engineering Consortium (IEC). On-line Education, Web-Proforum. Stream Control Transmission Protocol: http://www.iec.org/online/tutorials/sctp/. Archivo descargable al registrarse como usuario: sctp-tutorial.pdf, página 3.

[2] International Engineering Consortium (IEC). On-line Education, Web-Proforum. Stream Control Transmission Protocol: http://www.iec.org/online/tutorials/sctp/. Archivo descargable al registrarse como usuario: sctp-tutorial.pdf, página 6.

[3] International Engineering Consortium (IEC). On-line Education, Web-Proforum. Stream Control Transmission Protocol: http://www.iec.org/online/tutorials/sctp/. Archivo descargable al registrarse como usuario: sctp-tutorial.pdf, página 8.

[4] International Engineering Consortium (IEC). On-line Education, Web-Proforum. Stream Control Transmission Protocol: http://www.iec.org/online/tutorials/sctp/. Archivo descargable al registrarse como usuario: sctp-tutorial.pdf, página 9.

[5] International Engineering Consortium (IEC). On-line Education, Web-Proforum. Stream Control Transmission Protocol: http://www.iec.org/online/tutorials/sctp/. Archivo descargable al registrarse como usuario: sctp-tutorial.pdf, página 10.

[6] International Engineering Consortium (IEC). On-line Education, Web-Proforum. Stream Control Transmission Protocol: http://www.iec.org/online/tutorials/sctp/. Archivo descargable al registrarse como usuario: sctp-tutorial.pdf, página 11.