Redes privadas virtuales

Departamento de Sistemas Telemáticos y Computación (GSyC)

http://gsyc.urjc.es

Septiembre de 2011

Contenidos

1. Introducción
   - Protocolos para VPN
2. VPN en la pila OSI
   - Nivel de implementación de la VPN
   - VPN ofreciendo nivel de red
   - VPN ofreciendo nivel de enlace
   - Combinación de nivel ofrecido y de implementación
3. SSL/TLS
   - Características de SSL/TLS
   - SSL/TLS Handshake
4. OpenVPN
5. Secure Shell: SSH
   - Usos de SSH
   - Implementaciones de SSH
6. IPsec
7. Referencias

VPN

VPN: *Virtual Private Network*, Red Privada Virtual
Conjunto de ordenadores interconectados a través de una red insegura, típicamente internet, sobre la que se añaden protocolos adicionales para conseguir seguridad

- Varios segmentos de red distantes se comportan como si fueran uno solo. P.e. sucursales de una empresa, trabajador desde su casa
- Tecnología relativamente reciente. Sustituye a las línea dedicadas, mucho más caras, habituales durante la década de 1990

Las VPN incorporan las exigencias de seguridad (confidencialidad, integridad, autenticación, no repudio) que fueron ignoradas en el diseño original de IPv4

Puede implementarse de muy diversas formas

- En distintos niveles de la pila OSI
- Tanto a nivel *software* en los *hosts* como en el *hardware* de red
- Protocolos abiertos y protocolos propietarios

Cada protocolo tendrá sus ventajas e inconvenientes

- Seguridad
- Facilidad de configuración
  - Por parte del administrador
  - Por parte del usuario
- Rendimiento
- Exigencias en la red
- Relación con NAT
- Relación con cortafuegos
- ...

Protocolos para VPN

Los principales protocolos con los que se pueden formar VPN son

- SSL/TLS: *Secure Sockets Layer/Transport Layer Security*
  Base de muchas comunicaciones seguras: https, VoIP, OpenVPN, SMTP (*Simple Mail Transfer Protocol*), OpenSSL, JSSE (*Java Secure Socket Extension*) y otros
- OpenSSH
- IPsec, estándar de Internet

- PPTP: *Point-to-Point Tunneling Protocol*. RFC 2637, año 1999
  Desarrollado por Microsoft, 3Com y otros
  Protocolo sencillo, habitual en MS Windows y otros. Muchos problemas de seguridad
- L2TP: *Layer 2 Tunneling Protocol*
  Mejora de PPTP. Configuración algo complicada en Windows (a partir de Windows Vista SP1), mal soporte en Linux
- DTLS: *Datagram Transport Layer Security*. Propietario de Cisco

VPN en la pila OSI

Las diversas tecnologías de VPN pueden implementarse

- En el nivel de transporte con TCP
- En el nivel de transporte con UDP
- En el nivel de red

¿Podría implementarse una VPN en el nivel de enlace?

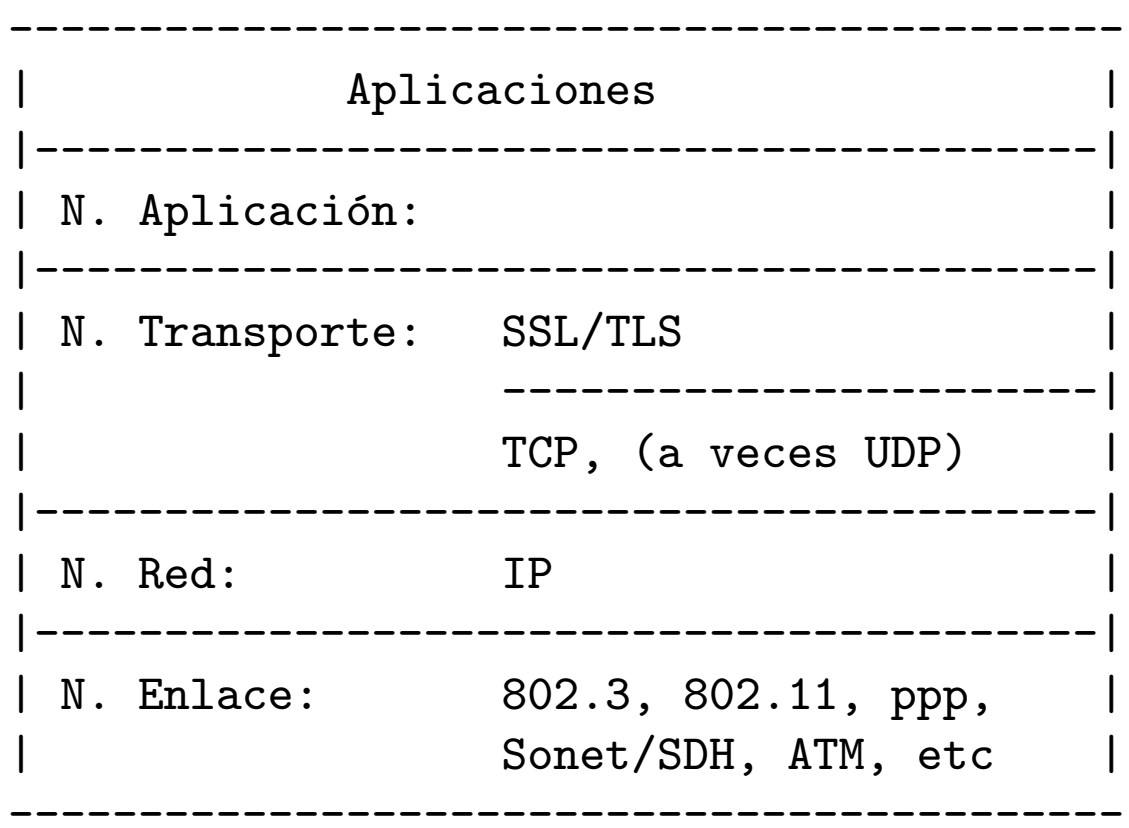
Nivel de implementación de la VPN

Las diferentes tecnologías pueden implentarse sobre distintos niveles

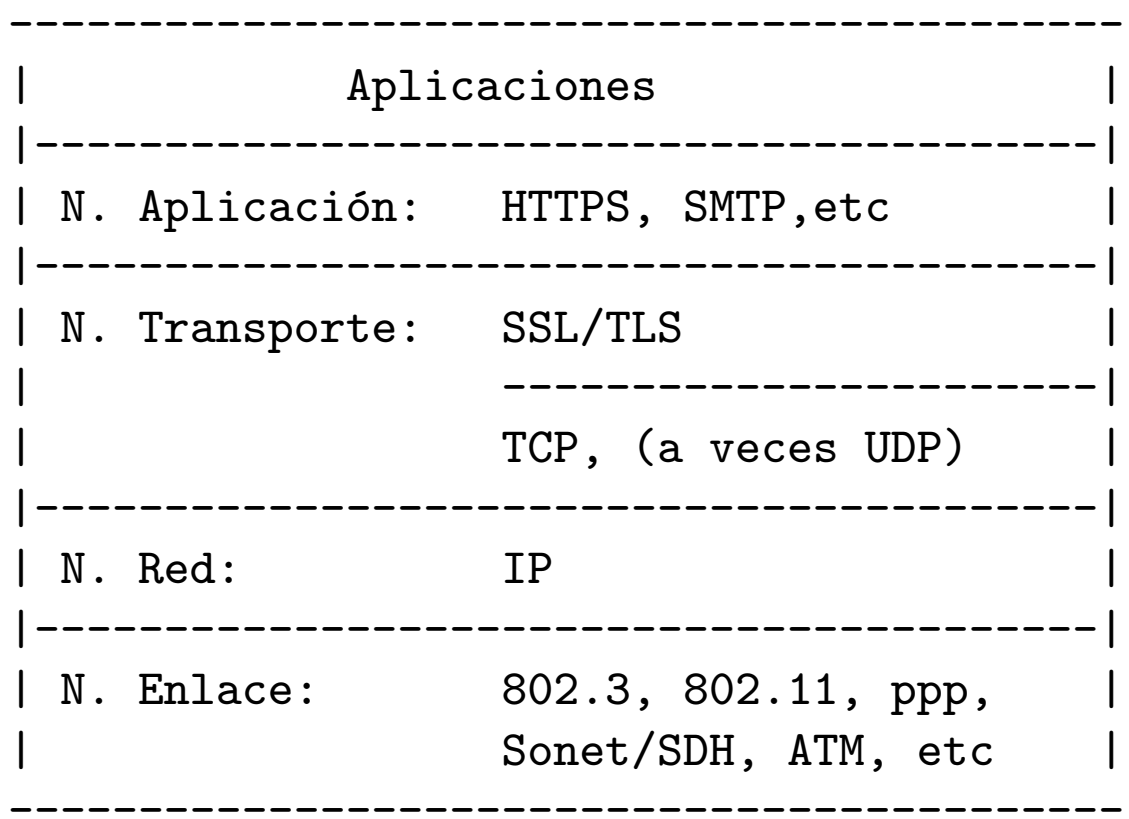
- SSL/TLS se implementa en el nivel de transporte, es la base p.e. para OpenVPN, que ofrece tanto redes de nivel 2 como de nivel 3 sobre UDP
- SSH se implementa en el nivel de aplicación y transporte, ofrece tanto redes de nivel 2 como de nivel 3, aunque siempre sobre TCP
- IPsec se implementa en el nivel de red, normalmente ofrece un nivel 3 aunque es posible configurarlo para ofrecer nivel 2

SSL/TLS en la pila

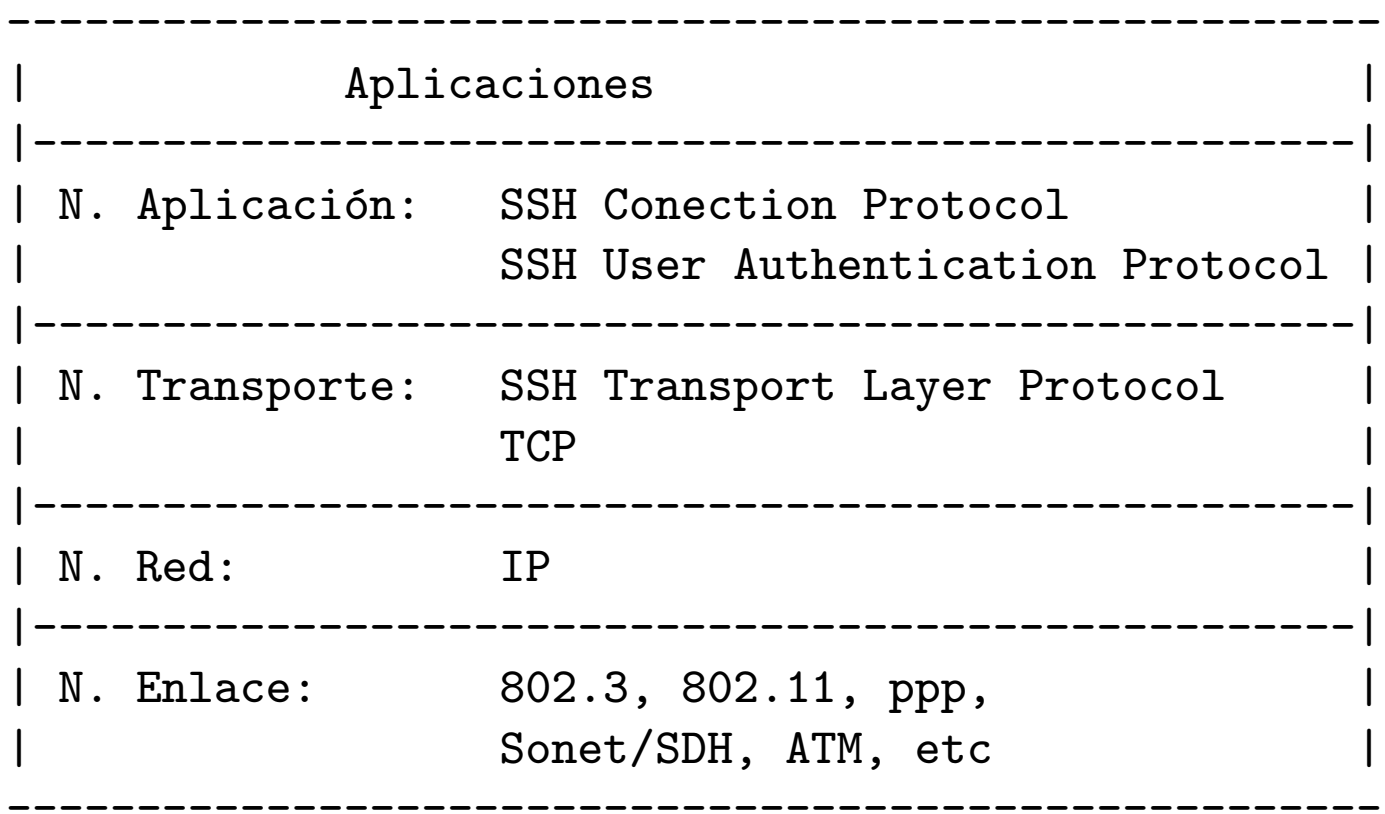
SSL/TLS se suele considerar ubicado en el nivel de transporte, las aplicaciones pueden invocarlo directamente



O bien SSL/TLS puede ofrecer sus servicios a un protocolo estándar del nivel de aplicación

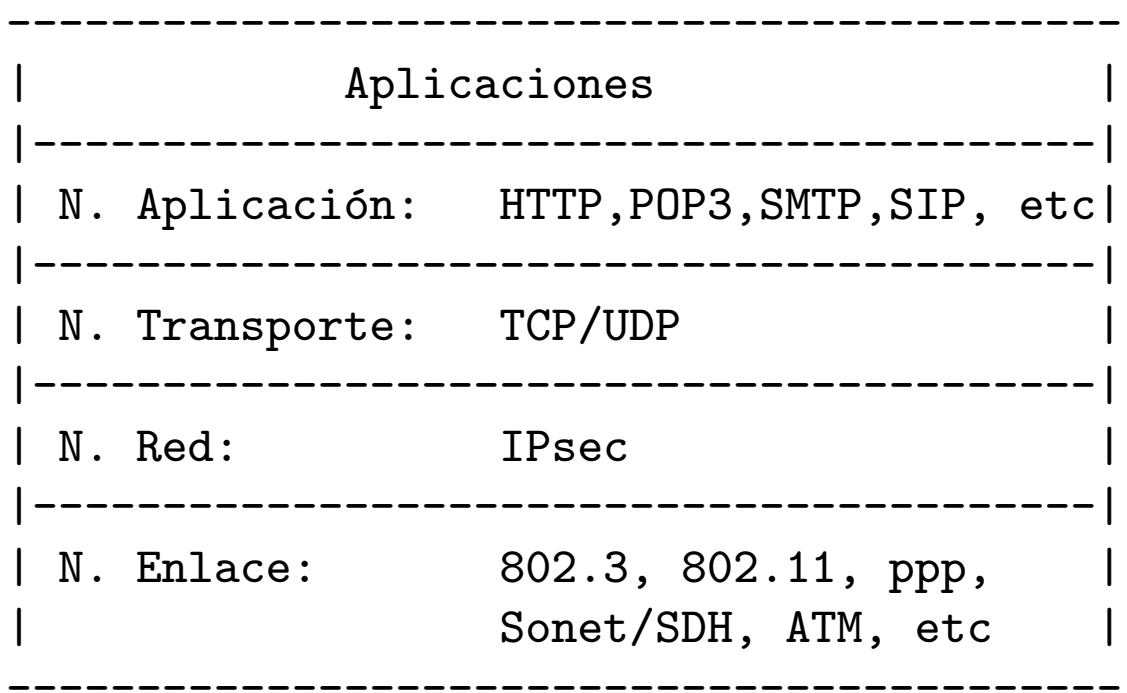


SSH en la pila



IPsec en la pila

IPsec reemplaza a IP



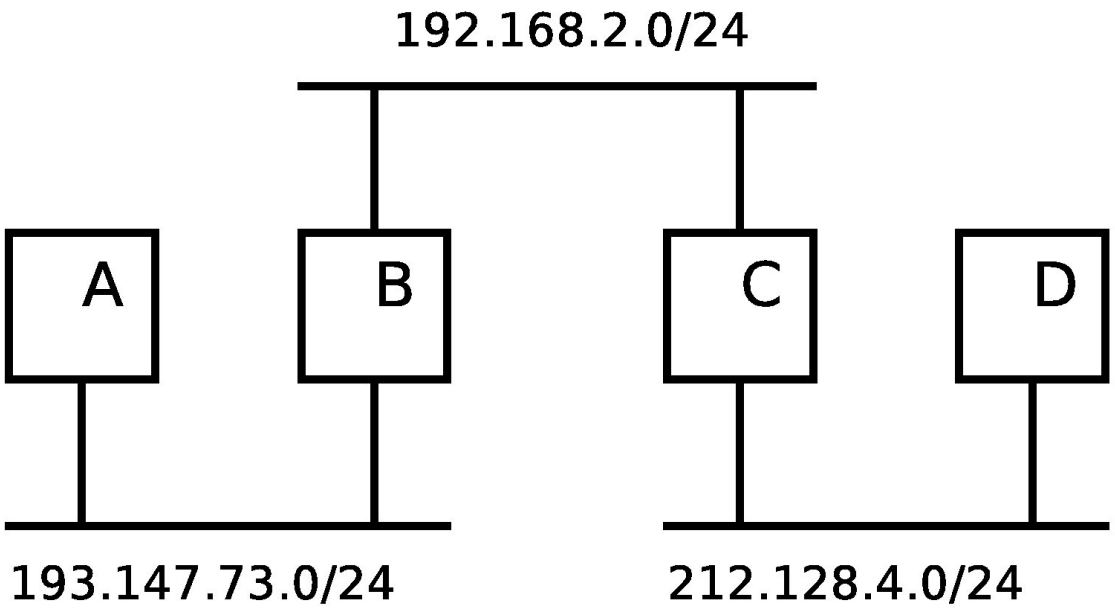
Nivel ofrecido por la VPN

Independientemente del nivel de implementación, la VPN puede ofrecer

- Un nivel de red. Segmento de red propio, conexión punto a punto, tráfico IP
  (Lo más habitual)
- Un nivel de enlace. Unir segmentos de red, tráfico ethernet

VPN ofreciendo nivel de red

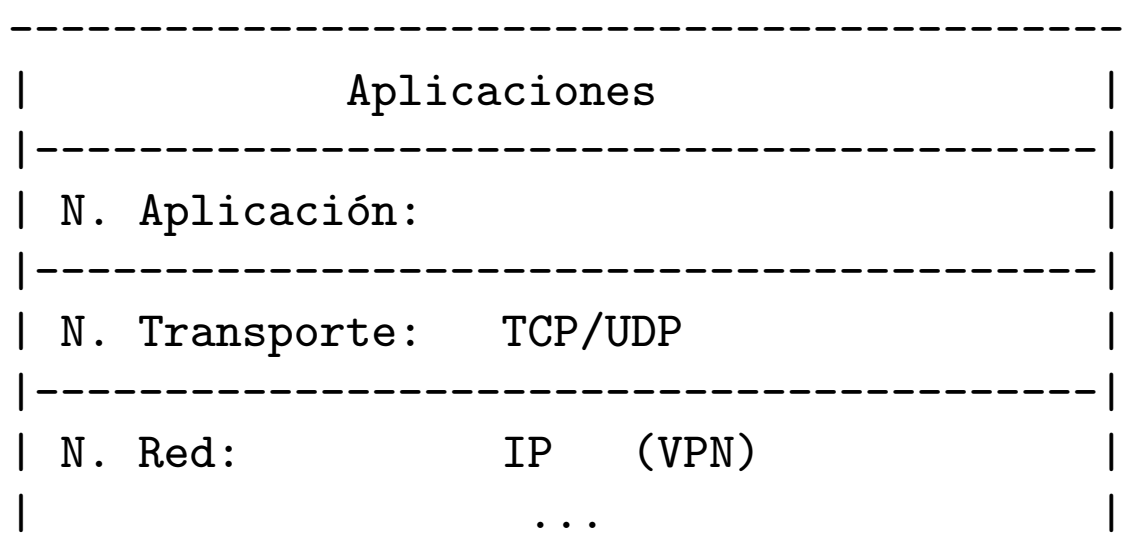
B y C están unidos por una VPN que ofrece un nivel 3 (red)



- Por la VPN se transmiten datagramas IP
- En el caso de Linux, esta funcionalidad la ofrece el dispositivo virtual *TUN* (*network TUNnel*)

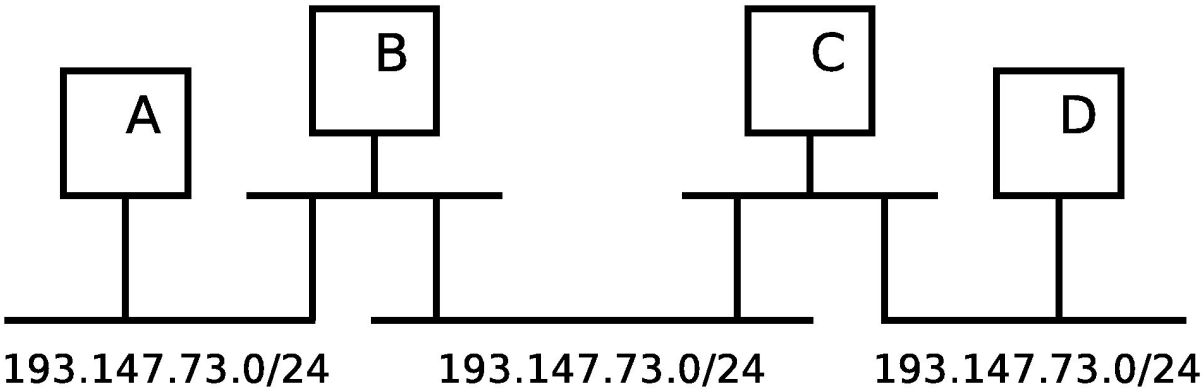
- La VPN crea una conexión punto a punto, formando su propio segmento de red
- A y D son accesibles
  - Encaminando adecuadamente
  - Empleando TCP/IP

Sobre el nivel IP que ofrece la VPN, se puede colocar cualquiera de los protocolos habituales de la familia TCP/IP



VPN ofreciendo nivel de enlace

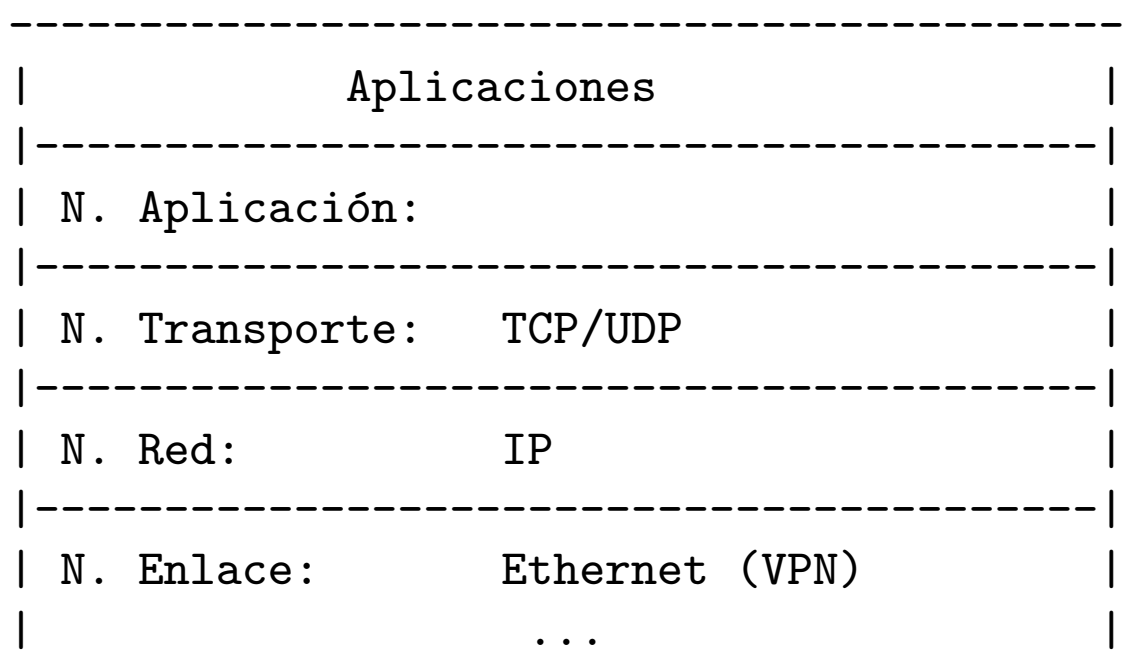
B y C están unidos por una VPN que ofrece un nivel 2 (enlace)



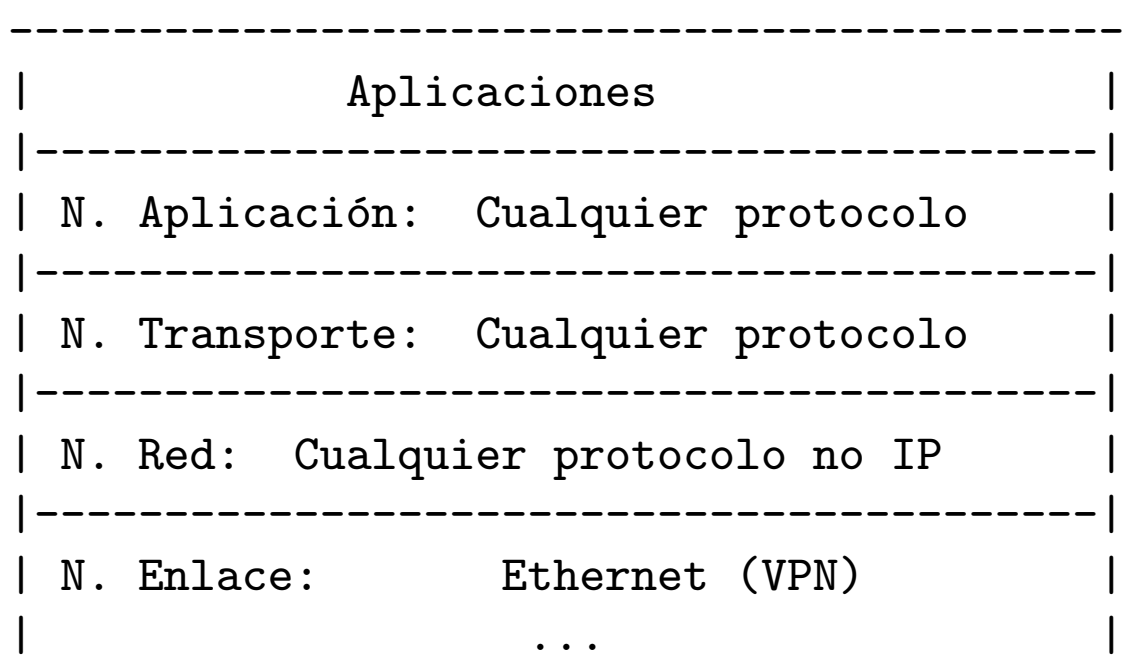
- Por la VPN se transmiten tramas ethernet
- En el caso de Linux, esta funcionalidad la ofrece el dispositivo virtual *TAP* (*network tap*)

- Las máquinas B y C están conectadas a un *bridge* virtual que une el interfaz TAP y el interfaz ethernet
- La VPN propaga todas las tramas, incluyendo Broadcast. A y D pasan a ser contiguas
- La red puede emplear cualquier protocolo de red, no solo IP
  - SMB/CIFS, protocolos de descubrimiento de servicios, streaming de audio y vídeo para LAN, juegos para LAN, etc

Sobre la red privada virtual ethernet puede trabajar IP



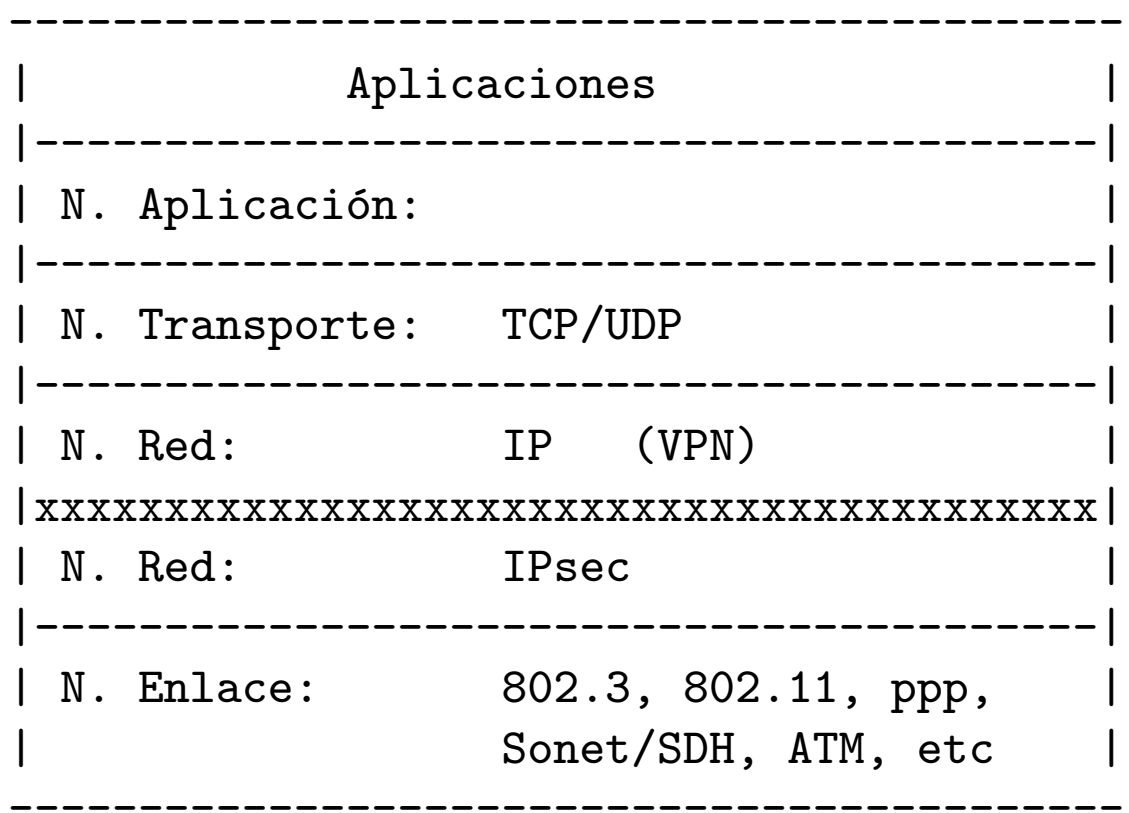
Pero sobre la red privada virtual ethernet también se puede colocar cualquier otro protocolo



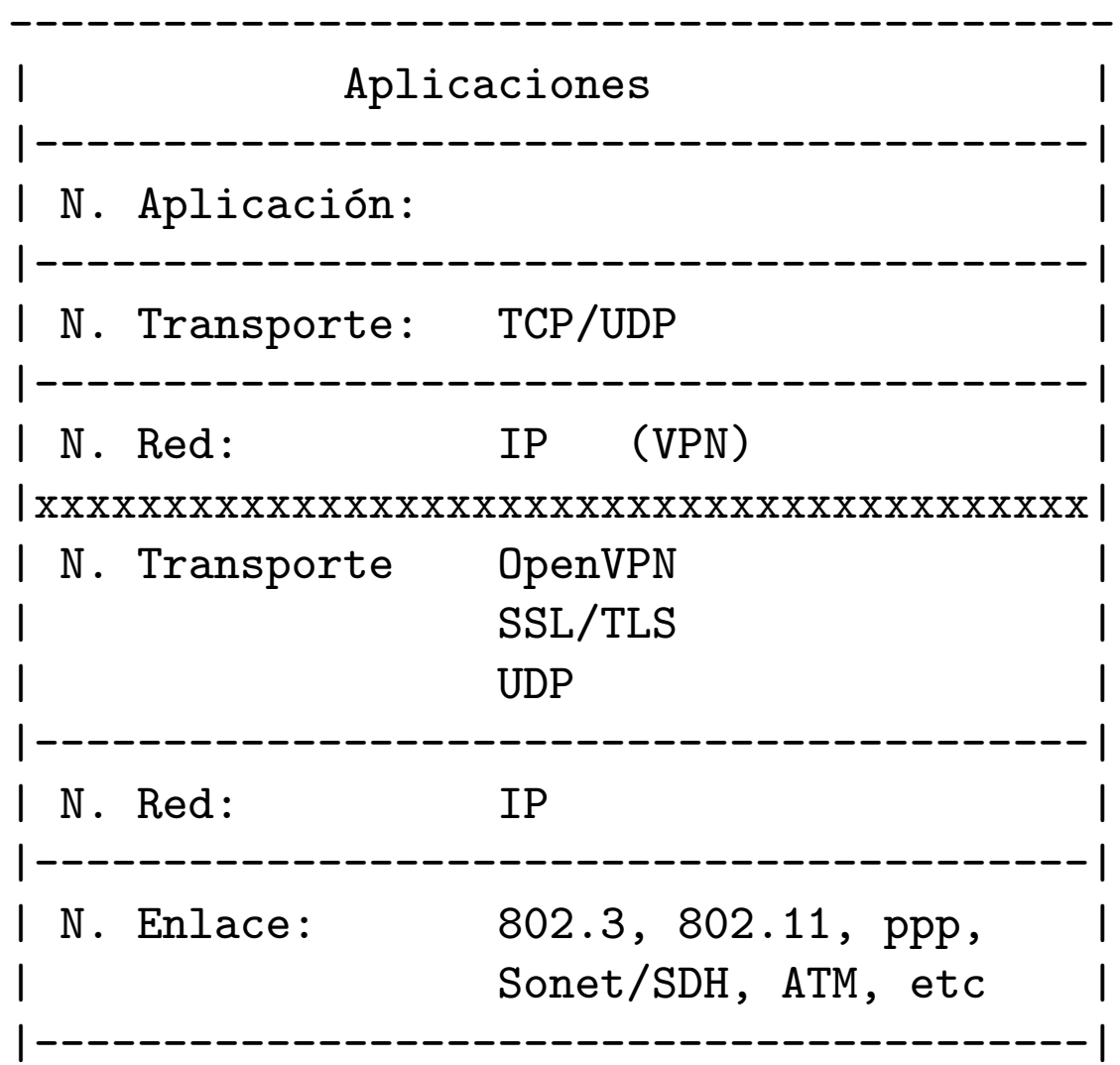
Combinación de nivel ofrecido y de implementación

El nivel ofrecido y el nivel de implementación pueden combinarse de diversas maneras

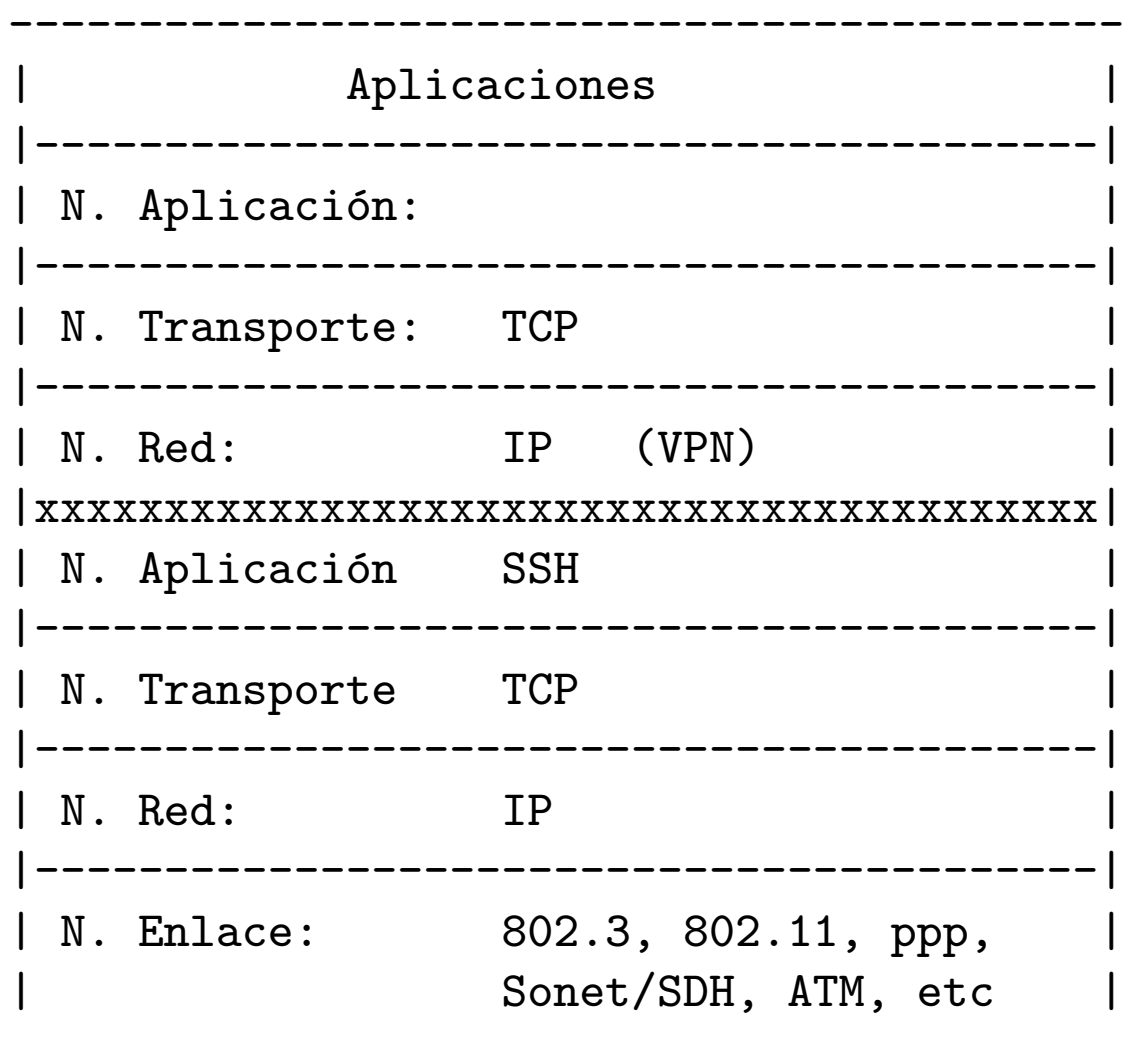
P.e. IPsec ofreciendo nivel 3, implementado sobre nivel 3



P.e. OpenVPN (SSL/TLS) ofreciendo nivel 3, implementado sobre nivel 4 con UDP



P.e. OpenSSH ofreciendo nivel 3, implementado sobre nivel 4 con TCP



Aunque prácticamente todas las combinaciones son posibles, no todas son igual de eficientes
TCP está pensado para trabajar sobre un protocolo no fiable, pero OpenSSH lo coloca sobre otro protocolo también fiable (TCP sobre TCP), lo que resulta una mala combinación

- Cuando el *timeout* de un segmento TCP vence y no hay asentimento, TCP supone que el nivel inferior, no fiable, lo ha perdido, y por tanto reenvía
- Pero el nivel inferior, fiable, no pierde segmentos. El ACK solamente se retrasado por la congestión en la red
- El reenvío es inecesario y además, empeora la congestión

SSL/TLS

Netscape desarrolla en 1995 el protocolo SSL: *Secure Sockets Layer*

- Cliente/servidor, diseñado originalmente para que servidor web y navegador intercambien tráfico de manera segura
- Sencillo, eficiente, muy extendido

Netscape prácticamente desaparece, el IETF desarrolla TLS: *Transport Layer Security*, protocolo estándar de internet, sucesor de SSL

- v1.0 RFC 2246 año 1998
  v1.2 RFC 5246 año 2008

SSL/TLS ofrece:

- Confidencialidad
- Integridad
- Autenticación

Se puede usar de dos maneras

- Invocado explícitamente por las aplicaciones
- Invocado por un protocolo estándar del nivel de aplicación

El tamaño de las claves es variable

- Originalmente, 40 bits
- Actualmente, 128 bits o más

- Normalmente trabaja sobre TCP aunque también puede hacerlo sobre UDP
- Cliente y servidor pueden negociar diversos protocolos de clave pública (RSA, OpenPGP, DSA, Diffie-Hellman) para el intercambio de claves privadas de sesión (DES, IDEA, AES...), autenticación mediante Kerberos, funciones hash mediante MD5, ...
- Normalmente solo está autenticado el servidor, opcionalmente también puede hacerlo el cliente
- Protege los datos en las comunicaciones, no en almacenamiento
- No específicamente diseñado para medios de pago, no incluye mecanismos como vincular un cliente a una tarjeta, o revocación de tarjetas

SSL/TLS Handshake

La parte más importante de SSL/TLS es la fase inicial, denominada *handshake*. Está basado en RSA
Interviene un CA (*Certificacion Authority*), que proporciona al servidor un certificado, compuesto por

- Clave pública RSA del servidor
- Firmada por la CA

1. Cliente envía a servidor
   - Versión de SSL/TLS que soporta
   - Qué algoritmos criptográficos prefiere
2. Servidor envía a cliente
   - Versión de SSL/TLS que soporta
   - Qué algoritmos criptográficos prefiere
   - Certificado
3. El cliente comprueba si la CA está en la lista de entidades que conoce previamente
   - En caso afirmativo, el cliente valida la clave RSA del servidor
   - En otro caso, advierte al usuario de que no puede garantizarse la identidad del servidor

4. El cliente genera una clave de sesión y la envía al servidor, cifrada con la clave pública del servidor
5. Cliente y servidor intercambian tráfico, cifrado con la clave de sesión

En Septiembre de 2011 aparece el *BEAST attack* contra SSLv3 / TLSv1. Aunque en la práctica es poco viable

OpenVPN

OpenVPN es una aplicación que permite implementar VPN

- Herramienta muy extendida, creado por James Yonan en 2001. Mantenida actualmente por OpenVPN project y OpenVPN Technologies
- Licencia GNU GPL
- Multiplataforma: Windows, Linux, MacOS, *BSD, Solaris, Android
  - Incompatible con IPsec o soluciones *hardware*
- Basado en SSL/TLS
  - SSL/TLS permite leer y escribir en un socket de forma segura, pero no es una verdadera VPN, necesita una capa adicional de *software*
- Ofrece redes de nivel 2 y nivel 3 (En Windows solo de nivel 2)
- Emplea módulos TUN/TAP

- Trabaja en espacio de usuario
- Normalmente UDP, también TCP
- Emplea un único puerto: el 1194
- Facil integración con NAT y cortafuegos
- Compatible con direcciones IP dinámicas (incluso mantiene la conexión si cambia la IP)
- Mucho más sencillo que IPsec, si bien ofrece peor rendimiento

Secure Shell: SSH

Creado en el año 1995 por Tatu Ylönen, investigador de la Universidad de Helsinki, Finlandia
Pasa a ser un estándar del IETF, RFC 4253 (Año 2006)

- Diseñado originalmente como alternativa segura para Telnet en entornos tipo Unix
- Puede emplear contraseñas ordinarias, RSA, DSA, Kerberos...
- Incluye tres protocolos apilados, en los niveles de transporte y aplicación, *SSH Conection Protocol, SSH User Authentication Protocol, SSH Transport Layer Protocol*

Usos de SSH

SSH se vuelve muy popular y se usa también para

- Transferencia segura de ficheros (SCP, SFTP)
- sshfs: sistema de ficheros en red
  Alternativa a NFS, el cliente no necesita que el servidor exporte un directorio por NFS, le basta con tener acceso por SSH
- *Port forwarding*. Permite comunicación segura sobre red insegura, muy similar a una VPN, pero trabajando con un único puerto
- OpenSSH desde la versión 4.3 (febrero de 2006) permite hacer verdaderas VPN (todos los puertos), tanto en nivel de enlace como en nivel de red

Implementaciones de SSH

Existen diversas implementaciones

- ssh original, con licencia *freeware*
- OpenSSH, desarrollado originalmente para OpenBSD, portada a muchos otros SO, es la versión más empleada
- Dropbear, versión reducida, habitual en SO empotrados como OpenWRT
- En MS Windows se puede usar el servidor CopSSH (OpenSSH compilado para Cygwin), o el cliente PuTTY

SSH podría trabajar sobre SSL/TLS pero no lo hace

- ¿Por qué?
-

IPsec

- La preocupación por la seguridad en internet no aparece hasta la RFC 1636, año 1994
- Diseñado originalmente para IPv6, donde es obligatorio soportarlo. Opcional en IPv4
- Conjunto bastante complejo de protocolos
  Originalmente RFC 1825, RFC 1820, año 1995
  Actualmente RFC 4301, RFC 4309, año 2005
- Proporciona confidencialidad, autenticación, integridad y no repudio de la comunicación

- Opera en el nivel de red
  - Niveles superiores (incluyendo aplicaciones) pueden despreocuparse. No es necesario modificar las aplicaciones
  - El usuario puede olvidar su existencia. Sin problemas de calidad de contraseñas, revocación de contraseñas, sesiones, etc
- Puede incluirse en los routers en el perímetro de una LAN, de forma que el tráfico interno no sufre sobrecarga
- Diversas implementaciones, Openswan es la más habitual en Linux

Inconvenientes

- Administración de cierta complejidad (lo que siempre supone un riesgo)
- Disponibilidad limitada en IPv4. Es necesario que toda la infraestructura de red soporte IPsec. No es tráfico tcp ni udp, un proveedor puede no permitirlo (o facturarlo adicionalmente)

Protocolos de IPsec

IPsec es un conjunto de protocolos

- Protocolos para la gestión de la confianza (*trust relationship management*, aka *control plane*)
  Normalmente en espacio de usuario
  IKE protocol (*Internet Key Exchange*)
- Protocolos para el manejo de paquetes (*packet handling*,aka *forwarding plane*
  Normalmente implementados a bajo nivel en el núcleo del S.O.
  1. AH, *Authentication Header*. Integridad, autenticación y no repudio, sin confidencialidad
  2. ESP, *Encapsulating Security Payload*. Confidencialidad y, posteriormente, integridad, autenticación y no repudio

  AH existe por motivos históricos, como limitaciones en la exportación de criptografía. Lo normal es emplear ESP

Security Associations

Antes de intercambiar tráfico, ambos extremos se autentican mediante IKE (*Internet Key Exchange*), con las técnicas habituales de criptografía asimétrica

- Forman una *Security Association* (SA): una relación segura, unidireccional, entre un emisor y un receptor
- Cada datagrama IP lleva cabeceras que lo asocia a un SA, y las estaciones conservan estado para cada SA (números de secuencia, tipo de protocolo, claves,...)
- Esto implica que se crea una conexión

El datagrama IPsec es distinto e incompatible con el datagrama IP

- Se le añade una cabecera de autenticación, que garantiza la integridad del datagrama
- El formato de los datos del datagrama también se modifica para permitir la confidencialidad

Modos de conexión en IPsec

- Modo túnel. Un datagrama IPsec que contiende un datagrama IP ordinario (con su propia cabecera y carga)
- Modo transporte. Un datagrama IPsec con una cabecera IPsec y dentro, un segmento TCP o un datagrama UDP
  Poco empleado, realmente no es una VPN

Referencias

- Data and computer communications
  W. Stallings. Ed. Prentice Hall
- Computer networking
  J.F. Kurose. Ed. Pearson
- Beginning OpenVPN 2.0.9
  M. Feilner, N. Graf. Ed. Packt Publishing
- Building and Integrating Virtual Private Networks with Openswan
  P. Wouters, K. Bantoft. Ed. Packt Publishing