

Migrando aplicaciones a IPv6

Estado actual y fundamentos de nuestro proyecto

Alonso Sandoval A. Hernán Vargas L.

Universidad Técnica Federico Santa María

`asandova@alumnos.inf.utfsm.cl`, `hvargas@alumnos.inf.utfsm.cl`

26 de septiembre de 2014

Índice

1 Contexto protocolo de internet

- Problemática actual
- IPv6 vs IPv4
- Principales implementaciones de IPv6

2 Proyecto aplicación Dual Stack

- Idea
- Entorno de desarrollo
- Sockets.h

Índice

1 Contexto protocolo de internet

- Problemática actual
- IPv6 vs IPv4
- Principales implementaciones de IPv6

2 Proyecto aplicación Dual Stack

- Idea
- Entorno de desarrollo
- Sockets.h

Índice

1 Contexto protocolo de internet

- Problemática actual
- IPv6 vs IPv4
- Principales implementaciones de IPv6

2 Proyecto aplicación Dual Stack

- Idea
- Entorno de desarrollo
- Sockets.h

Problemática direcciones IPv4

- La cantidad de direcciones IPv4 no dan a vasto: $4 \cdot 10^9$ app.
- La solución al problema consiste en la implementación de IPv6, con una cantidad de direcciones: $3,40 \cdot 10^{38}$ app.
- Lamentablemente, IPv6 e IPv4 no son compatibles.

Índice

1 Contexto protocolo de internet

- Problemática actual
- IPv6 vs IPv4
- Principales implementaciones de IPv6

2 Proyecto aplicación Dual Stack

- Idea
- Entorno de desarrollo
- Sockets.h

Comparación cabeceras

Version	IHL	TOS	Total Length	
Identification			Flags	Fragment Offset
Time To Live	Protocol	Header Checksum		
Source Address				
Destination Address				
Options				Padding

Figura : IPv4

- Version: Versión protocolo, 4.
- IHL: Tamaño encabezado IP
- TOS: Calidad de Servicio. Priorización paquetes.
- Total Length: Tamaño total datagrama, carga útil.
- Identification: Identificador para re-ensamblado
- Flags: Bits de control sobre la fragmentación
- Fragment Offset: Ubicación de este fragmento dentro del datagrama original
- TTL: Tiempo máximo vida datagrama.
- Protocol: Protocolo carga útil del paquete IP
- Header Checksum: Suma comprobación de paquetes
- Options: Campo opcionales, rara vez utilizado.
- Padding: Relleno con ceros, asegura que encabezado sea de tamaño múltiplo de 32 bits.

Comparación cabeceras

Cabe agregar que el campo opciones se deja como encabezado de extensión, con las posibles opciones:

- Hop-by-Hop Options: Opciones a revisar en cada salto.
- Routing: Especifica nodos intermedios a visitar antes de llegar a destino.
- Fragment: Utilizado para fragmentar paquetes con tamaño superior al MTU de algún enlace.
- Destination Options: Opciones revisadas por nodo de destino.
- Authentication: Para servicio de autenticación (IPsec)
- Encapsulating Security Payload: Diseñado para proveer confidencialidad, autenticación de origen, integridad, otros.

Algunas mejoras y diferencias de IPv6 con IPv4

- Formato Decimal a Hexadecimal: IPv4: 192.168.1.1, IPv6: 2001:0DB8:0:0:0:0:003D.
- Multicast Reemplaza a Broadcast: Mensajes múltiples a destinos específicos.
- Autoconfiguración de direcciones: Neighbor Discovery Protocol.
- Capa Seguridad: IPsec, uso en seguridad.
- Procesamiento simplificado: Eliminación de campos encabezado IPv6.
- Extensiones de Privacidad: Permite que direcciones auto-asignadas cambien con el tiempo.

Índice

1 Contexto protocolo de internet

- Problemática actual
- IPv6 vs IPv4
- Principales implementaciones de IPv6

2 Proyecto aplicación Dual Stack

- Idea
- Entorno de desarrollo
- Sockets.h

IPv6 sobre IPv4 (Máquinas IPv4 en internet IPv6)

- 6to4: A través de un túnel (Tunnel Broker), permite salida a internet IPv4 desde red IPv6.
- 6in4: Paso de paquetes IPv6 como payload de IPv4.
- 6over4: Requiere uso de multicast, no soportado ampliamente por IPv4.
- 6rd: similar a 6to4 solo que el ISP provee del Tunnel Broker.

IPv4 sobre IPv6 (No muy usado)

- Nat64/DNS64: DNS especial hace la traducción de registros A(IPv4) con quad A (IPv6).
- 4in6: Paquete IPv4 en en contenido de paquete IPv6.
- 4rd: Opuesto a 6rd

Dual Stack

- El método ideal ya que permite acceso nativo a ambos protocolos (Ipv4 e IPv6) de manera simultánea evitando costos en infraestructura(túneles por ejemplo) remitiendo estos a la adquisición de hardware.

Aplicación	Soporte	Observaciones
Apache HTTPD	Completo	Desde versión 2.0.
Desarrollos internos	Parcial	Es necesario revisar caso a caso el soporte de IPv6.
MySQL	Completo	Desde versión 5.5.3.
PostgreSQL	Completo	Desde versión 7.4.
Oracle Database	Parcial	Desde versión 11g Release 2.
Zimbra	Completo	Modalidad Beta, desde versión 8.0.
Postfix	Completo	Desde versión 2.2.
Sendmail	Completo	Desde versión 8.10.
Exchange	Completo	Desde versión 2013.

Índice

1 Contexto protocolo de internet

- Problemática actual
- IPv6 vs IPv4
- Principales implementaciones de IPv6

2 Proyecto aplicación Dual Stack

- Idea
- Entorno de desarrollo
- Sockets.h

Índice

1 Contexto protocolo de internet

- Problemática actual
- IPv6 vs IPv4
- Principales implementaciones de IPv6

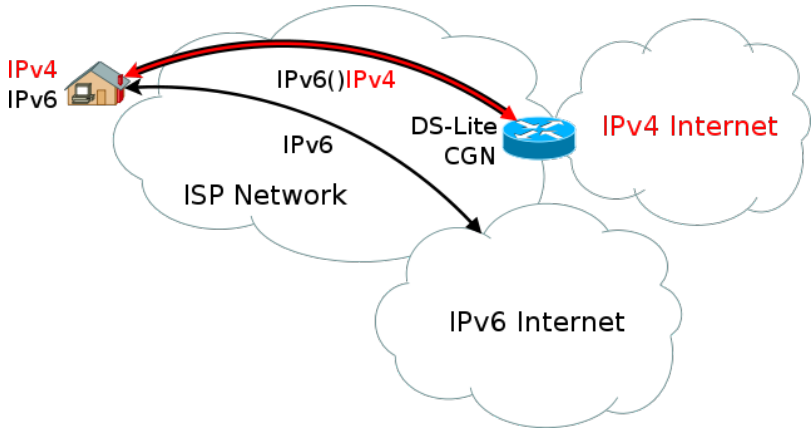
2 Proyecto aplicación Dual Stack

- Idea
- Entorno de desarrollo
- Sockets.h

Aplicación

- Dos aplicaciones conectadas por sockets, una para el cliente y otra para el servidor.
- El servidor es capaz de aplicar dual-stack, es decir, procesa los paquetes sin importar si vienen en IPv4 o IPv6.
- El cliente es capaz de enviar paquetes de ambos protocolos.
- Se pueden obtener estadísticas de las operaciones efectuadas (tiempos, tazas de perdida, entre otros).

Aplicación



Índice

1 Contexto protocolo de internet

- Problemática actual
- IPv6 vs IPv4
- Principales implementaciones de IPv6

2 Proyecto aplicación Dual Stack

- Idea
- Entorno de desarrollo
- Sockets.h

Internet IPv6

- La mayoría de los ISP no proveen de acceso a la red IPv6 en los hogares.
- El departamento de Informática de la UTFSM ofrece el acceso a la red local en IPv6
- Por ello, utilizaremos una máquina virtual en la red del DI como servidor con la cual nos comunicaremos en la misma red bajo IPv6.

Máquina virtual

- Utilizaremos una VM que nos disponga algún laboratorio en el departamento.
- Para la VM utilizaremos la distribución de linux CentOS que soporta IPv6 además de estar orientada a servidores.



Índice

1 Contexto protocolo de internet

- Problemática actual
- IPv6 vs IPv4
- Principales implementaciones de IPv6

2 Proyecto aplicación Dual Stack

- Idea
- Entorno de desarrollo
- Sockets.h

Sockets

- Un socket es un mecanismo por el cual se traspasan mensajes entre aplicaciones, generalmente en diferentes computadores.
- Está definido al menos por las direcciones IP, números de puerto y el protocolo de transporte.
- Los protocolos más utilizados son TCP y UDP.
- Implementan la arquitectura cliente-servidor, el programa que inicia la comunicación será el cliente, mientras que el que responde actúa como servidor.

socket.h

- API para el trabajo con sockets en c para linux.
- Soporta tanto IPv4 como IPv6.
- Incorpora funciones para la creación de sockets, convención entre formatos, y transmisión de datos.
- Define estructuras para el manejo de direcciones.

socket.h (continuación)

En general el procedimiento es el siguiente:

- Se crea la estructura para manejar la dirección. `sockaddr_in` para IPv4 y `sockaddr_in6` para IPv6.
- Se crea el socket y se le asigna un descriptor. `socket()` ;
- Se asocia el socket y la estructura de dirección. `bind()` ;
- Se pone al socket como “escucha” esperando conexiones. `accept()` ;
- Cuando una conexión es detectada se crea un nuevo socket, al que se le llama “socket conectado”.
- Se puede lanzar un subprocesso para continuar “escuchando” y trabajar con el “socket conectado”.

Migrando aplicaciones a IPv6

Estado actual y fundamentos de nuestro proyecto

Alonso Sandoval A. Hernán Vargas L.

Universidad Técnica Federico Santa María

`asandova@alumnos.inf.utfsm.cl`, `hvargas@alumnos.inf.utfsm.cl`

26 de septiembre de 2014