



DEPARTAMENTO DE COMPUTACION

Facultad de Ciencias Exactas y Naturales - UBA

NetworkDEVS

Una herramienta de estudio para modelar redes usando DEVS y Power-DEVS

16 de mayo de 2017

Teoría de las telecomunicaciones

Integrante	LU	Correo electrónico
Belloli, Laouen Mayal Louan	134/11	laouen.belloli@gmail.com

Instancia	Docente	Nota
Primera entrega		
Segunda entrega		



Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires
Ciudad Universitaria - (Pabellón I/Planta Baja)
Intendente Güiraldes 2160 - C1428EGA
Ciudad Autónoma de Buenos Aires - Rep. Argentina
Tel/Fax: (54 11) 4576-3359
<http://www.fcen.uba.ar>

Índice

1. Introducción y Motivación	2
2. Objetivos	2
2.1. Desarrollar un framework para modelos de redes	2
2.2. Modelar una red UDP/IP básica	3
2.3. Mapeo entre el modelo UDP/IP y el modelo OSI	3
3. Arquitectura	3
3.0.1. Conceptos básicos de la arquitectura	3
3.0.2. Arquitectura implementada	4
3.0.3. Arquitectura de las capas	5
4. Como usar el template	6
4.1. Como heredar el modelo layer	7
4.2. Como enviar y recibir mensajes entre las capas	9
4.3. Como implementar el protocolo de una capa (internal/external functions)	11
5. Como agregar una capa	11
6. Input/Output - Modelos generadores y vertederos	13
6.1. Input	13
6.2. La clase Parser	13
6.3. El modelo input stream	14
6.4. El modelo output stream	14
7. Logger	15
8. Caso de estudio	15
8.1. DNS	15
8.1.1. Estructuras de datos correspondientes	16
8.1.2. Funcionamiento del protocolo	17
8.2. UDP	19
8.2.1. Estructuras de datos correspondiente a UDP	19
8.2.2. Comportamiento del protocolo	20
8.3. IP	22
8.4. ARP	24
8.5. Datagrams	26
8.6. SWP	27
8.6.1. Remitente:	28
8.6.2. Destinatario:	28
8.7. Tests de unidad	29
8.7.1. Test de unidad para el módulo DNS	29
8.7.2. Test de unidad para el módulo UDP	32
8.7.3. Test de unidad para el módulo IP Host	38
8.7.4. Test de unidad para el módulo IP Router	41
8.7.5. Test de unidad para el módulo link	44
8.8. Test de integración - Escenario implementado	45
9. Resultados	47
10. References	99

1. Introducción y Motivación

El presente trabajo introduce un modelo DEVS de una red funcionando bajo el protocolo UDP/IP encuadrando lo mejor posible cada uno de sus módulos en su correspondiente capa respecto al modelo OSI presentado en la literatura oficial de la materia de teoría de la telecomunicaciones [3]. El trabajo fue desarrollado en el simulador PowerDEVS y pensado como herramienta de estudio para futuros alumnos de la materia.

Además del modelo, este trabajo aporta un framework que permite programar y modificar los protocolos de una red de telecomunicaciones para luego simular su comportamiento y obtener feedback instantáneo de los resultados. También, usando el simulador PowerDEVS, se consigue brindar una interfaz gráfica intuitiva que mapea de forma directa los dispositivos de la red con los módulos del modelo. Esta interfaz gráfica permite crear distintos escenarios topológicos sin tener que tocar ni una linea de código.

Por otro lado, esta interfaz gráfica permite modularizar de forma explícita las distintas capas del modelo y sus interacciones. De esta forma, se programa cada protocolo en su capa correspondiente, esto no es menor, ya que si bien, teóricamente, los distintos protocolos están separados en capas y las redes pueden ser pensadas así, en la realidad, estas capas no existen de forma explícita y hay un salto entre las teorías de redes y los modelos en capas respecto de sus distintas implementaciones prácticas. Este modelo, permite a los alumnos hacer trabajos prácticos implementando los protocolos y manteniendo explícitas estas capas, por ende, permite acercar la teoría y la práctica de las redes de telecomunicaciones.

Si bien, el modelo realizado es un modelo DEVS (Discrete Event System Specification), el mismo fue pensado para que no sea necesario más que una breve introducción al formalismo, por lo que mismo personas sin ninguna experiencia en DEVS deberían ser capaces de poder utilizar el simulador, el modelo y el framework general propuesto en este trabajo. De todas formas, una explicación detallada sobre DEVS y PowerDEVS puede ser encontrada en [1], un documento donde los autores de PowerDEVS explican con detalles, el formalismo DEVS y el simulador PowerDEVS.

2. Objetivos

Los objetivos de este trabajo se pueden dividir en varias partes:

2.1. Desarrollar un framework para modelos de redes

Como ya fue mencionado en la introducción, uno de los principales objetivos de este trabajo es desarrollar un framework, que permita fácilmente implementar protocolos de red, validarlos y experimentar con ellos mediante Simulación de Eventos Discretos (DES). Esta herramienta permite testear los protocolos implementados de forma sencilla y rápida. Obteniendo una herramienta útil, tanto para el estudio de las redes de telecomunicaciones como para la investigación en el área.

La creación de un modelo que sirva como framework, tiene por objetivo factorizar y estandarizar las partes comunes a todos los protocolos, estas partes comunes provienen de las propiedades inherentes a los dispositivos físicos en los cuales los protocolos están corriendo, y de los estándares actualmente utilizados que permiten obtener robustez y compatibilidad entre distintas implementaciones que puedan existir en distintas partes de una red. Estas abstracciones encapsuladas en el modelo framework presentado, permiten al modelador, concentrarse plenamente en su protocolo y permite la reutilización y compatibilidad de distintos modelos que pueden luego ser combinados en un único modelo de una red que funcione bajo distintos protocolos, permitiendo así el estudio de la compatibilidad, homogeneidad y efectividad de los protocolos implementados.

2.2. Modelar una red UDP/IP básica

Por otro lado, este trabajo ~~pretende~~ implementar un modelo básico de una red UDP/IP que permita cubrir los protocolos mínimos e indispensables para permitir enviar mensajes entre distintos dispositivos. Este modelo no tiene como intención implementar todas las partes, ni cubrir todos los protocolos, por lo que temáticas como la fragmentación de paquetes, congestión de tráfico, manejo de errores, Dynamic Host Configuration Protocol (DHCP) [3, p. 231], Routing Information Protocol (RIP) [3, p. 243] y Spanning tree protocol [3, p. 194] quedan fuera del alcance de este trabajo, siendo los mismos posibles trabajos futuros.

También se implementó un modelo de switch que utiliza el protocolo Datagramas [3, p. 172] utilizando una forwarding table para enviar los Frames por la interface correspondiente.

Para completar el modelo, se implementó el protocolo DNS correspondiente a la capa de aplicación y se simuló un escenario en el cual un host desea obtener la dirección ip de otro host, cuyo dominio es network.devs.com.

2.3. Mapeo entre el modelo UDP/IP y el modelo OSI

En este trabajo, también pretendemos mostrar como se pueden implementar los distintos protocolos del modelo UDP/IP encuadrándolos en un modelo en capas OSI. Para esto, varias decisiones tuvieron que tomarse, siendo tal vez la más difícil, decidir en que capa implementar el protocolo ARP [3, p. 228]. Para tomar estas decisiones, se tomaron siempre como referencia los textos del libro oficial de la materia [3]. Se intentó también separar lo mejor posible las tareas de cada capa de forma tal que cada una de ellas pueda ignorar lo más posible la existencia de la o las capas inferiores.

3. Arquitectura

3.0.1. Conceptos básicos de la arquitectura

La arquitectura general del modelo propuesto en este trabajo tiene por principal objetivo ofrecer un framework donde se pueda fácilmente desarrollar protocolos, para esto, se concentra en resolver varios puntos principales:

- Estandarizar la comunicación entre las capas de forma de conseguir consenso entre los distintos modeladores que usen esta herramienta, y que facilite una comprensión jerárquica y organizada del modelo entero para los lectores nuevos.
- Implementar todas aquellas partes generales a todos los modelos de protocolos de forma que solo sea necesario concentrarse en el protocolo a implementar.
- Minimizar lo más posible la necesidad de tener conocimientos avanzados sobre DEVS a la hora de utilizar la herramienta.

3.0.2. Arquitectura implementada

La arquitectura consiste en un modelo acoplado “dispositivo” de N capas, con $N > 0$ de forma que cada capa $i, i \in [1, \dots, N]$ se comunica con las capas $i + 1$ e $i - 1$ utilizando 8 canales de comunicación distintos:

- Output port 0: Envío de datos a la capa $i + 1$.
- Output port 1: Envío de controles a la capa $i + 1$.
- Output port 2: Envío de datos a la capa $i - 1$.
- Output port 3: Envío de controles a la capa $i - 1$.
- Input port 0: Recepción de datos de la capa $i + 1$.
- Input port 1: Recepción de controles de la capa $i + 1$.
- Input port 2: Recepción de datos de la capa $i - 1$.
- Input port 3: Recepción de controles de la capa $i - 1$.

Dado que no siempre sucede que en un dispositivo de la red, exista un solo módulo por capa, y dado que los puertos de salida y entradas están pensados para comunicarse con un solo modelo en la capa superior e inferior. Es necesario usar modelos demultiplexers que permitan redirigir los mensajes salientes por un puerto al modelo correspondiente de entre todas las posibles opciones. También, se puede usar modelos multiplexers para que los múltiples modelos de la capa superior o inferior envíen mensajes al modulo y quede guardado en el campo *interfaz* la identidad del modulo de origen del mensaje si fuera necesario.

Ejemplos de estos son:

- Un host que tiene varias aplicaciones enviando datos a través de distintos sockets: Hay más de un módulo en la capa de aplicación (capa siete del modelo OSI) de ese host.
En este caso debería haber también varias capas TCP (una por cada socket)
- Un router que tiene más de una interfaz: Hay una interfaz por cada sub-red a la cual esté conectado, contando con más de un módulo en la capa de enlace (capa dos del modelo OSI).

*Esto es común en los PCs, pero en los servidores suele haber más de 1 conexión física.
Los routers por ejemplo tiene muchos puertos físicos de entrada/salida*

Las capas 1 y N usan la misma arquitectura que el resto de las capas y tienen definidos los mismos puertos. Como existe solo un medio de comunicación físico entre dispositivos por donde datos y controles son enviados, el modelo acoplado “dispositivo” tiene solo un puerto de salida y un puerto de entrada destinado a comunicarse con otros dispositivos, y queda en el modelador decidir que puerto o puertos de salida de la capa 1 conectar con el puerto de salida del modelo acoplado y que puerto o puertos de entrada de la capa 1 comunicar con los el puerto de entrada. Por otro lado, el input requerido por la capa N (comunmente el input que indica los pedidos del usuario de enviar o recibir datos) se puede obtener mediante modelos generadores situados adentro del modelo acoplado. Lo mismo ocurre con el output generado por la capa N, este output puede ser recibido por modelos vertederos que lo guardan en archivos.

Distintos dispositivos de la red pueden tener distintas cantidades de capas, por lo cual, la cantidad de capas existente puede variar entre los distintos modelos de cada dispositivo dentro de una misma red. Los hosts por ejemplo, suelen tener hasta la capa siete, mientras que los routers comunes llegan a la capa tres y los switches a la capa dos.

La Figura 1 muestra la arquitectura general recién explicada en un modelo de una red con un host y un router, en el mismo se pueden ver: los multiplexers y demultiplexers, modelos con distintas cantidades de capas y las conexiones entre capas utilizando los distintos puertos.

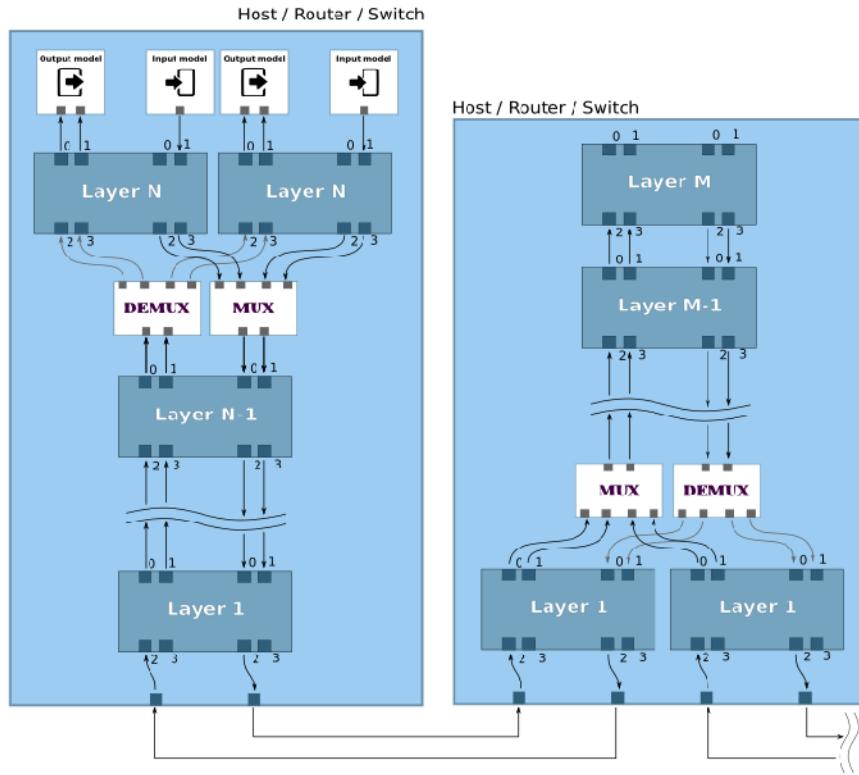
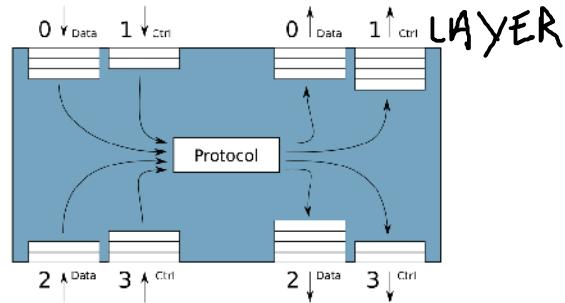


Figura 1: Arquitectura de un modelo en NetworkDEVS con dos dispositivos de N y M capas correspondientemente.

3.0.3. Arquitectura de las capas

La arquitectura de las capas está implementada en un **modelo template** llamado **Layer** del cual se puede realizar **herencia** para heredar su estructura. Esta estructura permite abstraer al protocolo de todo el modelo correspondiente al dispositivo, en otras palabras, esta estructura ya modela las propiedades de un dispositivo genérico, esto implica manejar el envío y recepción de mensajes a y desde las otras capas. Para manejar la entrada y salida de mensajes se utilizan colas FIFO; Cada vez que hay un mensaje entrante, el mismo es encolado en la cola de entrada correspondiente y el protocolo los va desencolando y atendiendo de uno o de muchos dependiendo su implementación, el modelo template ya está armado de forma que mientras no haya mensajes por procesar se mantiene pasivo (en estado IDLE) y mientras que hay mensajes por procesar se queda ciclando para ir atendiendo a todos hasta que no halla más mensajes a procesar, momento en el que el modelo se vuelve a pasivar. Por otro lado, cada vez que se quiere enviar un mensaje, el mismo solo debe ser encolado en la cola de mensajes salientes correspondiente dependiendo de si es un mensaje de datos o control a la capa superior o inferior, luego el simulador cuando el protocolo termina el procesamiento actual se encarga de ciclar entre las colas de salida para enviar todos los mensajes que hallan sido encolados. La Figura 2 muestra la arquitectura general de una capa cualquiera y la Figura 3 muestra el flujo de procesamiento generado por el modelo template cuando es extendido agregándole un protocolo.



unicamente 1 protocolo por LAYER?
Figura 2: Arquitectura de una capa en NetworkDEVS.

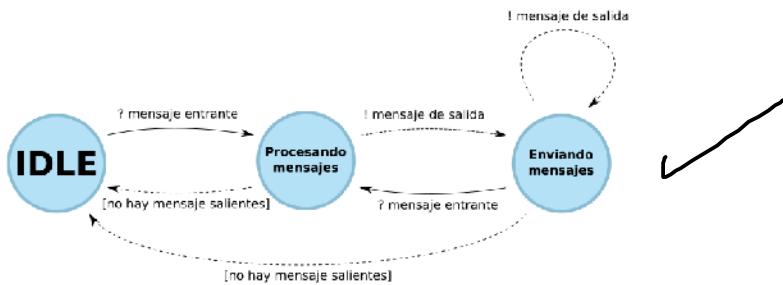
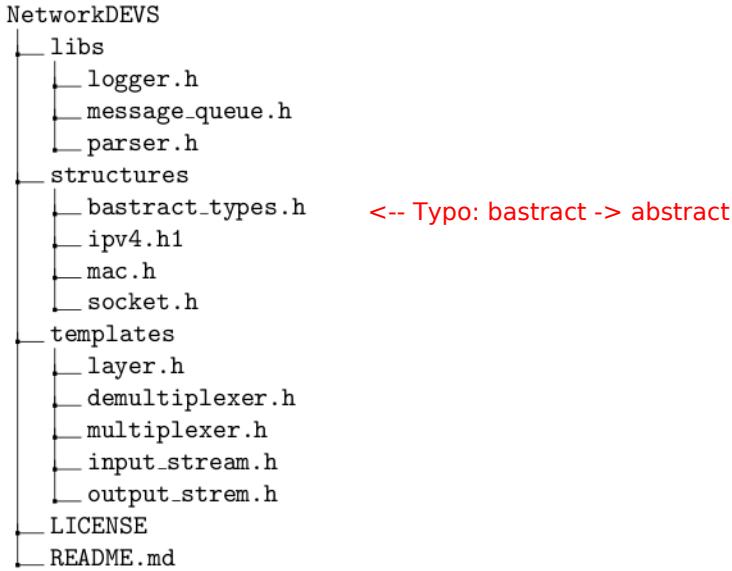


Figura 3: Flujo de procesamiento de una capa que extiende al modelo template Layer.
 Para que sea un DEVSGraph faltarian los ta (inf en IDLE y 0 en enviando)
 Si un protocolo quiere modelar la demora en el procesamiento, ¿puede cambiar el ta?

4. Como usar el template

Todo el modelo está implementado en C++ y la forma de usar el template es mediante la implementación de herencias. Para poder comprender como utilizarlo, primero introduciremos la estructura de archivos del framework.



La carpeta *structures* es la que contiene las definiciones de todos los tipos de datos que vienen por defecto con el framework, algunos de ellos fueron implementados exclusivamente para el modelo presentado en este trabajo, pero todos pueden ser utilizados y están incluidos por el modelo template *Layer* que introduciremos a continuación. Los tipos de datos están implementados usando structs y casi todos heredan de dos tipos abstractos definidos en *abstract_types.h* que sirven

como organizadores. De esta forma quedan ~~están~~ separados entre los que son Data y los que son Headers. Cada tipo de dato definido cuenta con documentación en el código en formato Doxygen.

Todos los tipos de datos están implementados en namespaces que sirven como organizadores, para cada capa existente debe existir un namespace y los tipos de datos inherentes a esa capa tienen que estar definidos dentro del namespace (por más que otras capas lo usen). Si se implementan nuevas capas, se deben crear sus correspondientes namespaces a menos que no halla tipos de datos correspondientes a la capa. De esta forma se evitan las ambigüedades que son muy recurrentes en el ámbito de las redes y se explica la interacción con los datos correspondientes a cada capa.

Nota: No están incluidos en esta estructura todos los archivos puntuales al modelo particular implementado, solo se encuentran aquellos archivos correspondientes a la arquitectura del framework.

4.1. Como heredar el modelo layer

El modelo *Layer* es el modelo template que contiene toda la implementación correspondiente a la arquitectura general de una capa. El mismo es un modelo que cuenta con 4 tipos de datos templates a instanciar y 4 tipos de datos template con instanciación por defecto si no se explicitaran otros tipos para ellos. Estos tipos a instanciar son los tipos de los mensajes que envían y reciben los puertos y los tipos con los cuales se definen las colas. **Por defecto se asume que los modelos intercambian los mismos tipos de datos, por ejemplo, si dos modelos intercambian datos de control, estos datos son del mismo tipo en ambas direcciones**, sin importar quien es el que los envia y quien es el que los recibe. Es por esto que hay solo 4 tipos template que deben ser instanciados obligatoriamente, mientras que los restantes cuatro se mapean por defecto con los primeros 4. De todas formas, **si se desea que los tipos de datos no sean iguales dependiendo quien envía y quien recibe, se pueden instanciar todos los tipos template.**

El modelo *Layer* tiene la siguiente forma:

```
template <typename DH, CH, DL, CL, DH2 = DH, CH2 = CH, DL2 = DL, CL2 = CL>
class Layer: public Simulator {

protected:
    // Logger
    Logger logger;

    // Colas de entrada
    std::queue<DH> higher_layer_data_in; // Input Port 0
    std::queue<DL> lower_layer_data_in; // Input Port 1
    std::queue<CH> higher_layer_ctrl_in; // Input Port 2
    std::queue<CL> lower_layer_ctrl_in; // Input Port 3

    // Colas de salida
    message::queue<DH2> higher_layer_data_out = 0; // Output Port 0
    message::queue<DL2> lower_layer_data_out = 1; // Output Port 1
    message::queue<CH2> higher_layer_ctrl_out = 3; // Output Port 2
    message::queue<CL2> lower_layer_ctrl_out = 3; // Output Port 3

    Event output;

    // Manipulacion del tiempo virtual de la simulacion
    double next_internal; → sigma
    double last_transition; → esto seria e?
    double infinity = std::numeric_limits<double>::max();
    bool queuedMsgs() const { ... }

public:
```

```

Layer(const char *n): Simulator(n) {};
double ta();
Event lambda(double) { .... }
void dint(double t) { .... }
void dext(Event x, double t) { .... }

virtual void dinternal(double t) {}
virtual void dexternal(double t) {}
};

```

PowerDEVS tiene un par de exigencias a la hora de declarar un nuevo modelo, por este motivo, a continuación se muestra un ejemplo de herencia que puede ser usado de base para implementar nuevos modelos de capas.

Protocol herada de layer
¿Esto quiere decir que un protocolo es una capa?

```

protocol_name.h

//CPP: path_to_cpp_files/protocol_name.cpp
#include "template/layer.h"

class protocol_name: public Layer<T1, T2, T3, T4 [ , T5, T6, T7, T8]> {

protected:

    // Variables privadas del modelo
    // Metodos privados del modelo

public:
    protocol_name(const char *n): Layer(n) {};
    void init(double, ...);
    void exit();

    virtual void dinternal(double);
    virtual void dexternal(double);

    // Variables publicas del modelo
    // Metodos publicos del modelo
};

```

Exigencias

- La primer linea es requerida por el compilador de PowerDEVS y tiene que tener correctamente seteado la dirección al archivo .cpp donde estén implementados todos los métodos del modelo.
- El constructor de la clase debe estar declarado como se muestra en el ejemplo y la función *init* es la utilizada por el simulador para inicializar el modelo. Esta función debe ser declarada como se muestra con los “...” como parámetro, el simulador llama a esta función pasándole como primer parámetro *double* la cantidad de parámetros a leer y luego el resto de los parámetros definidos para el modelo en la IDE PowerDEVS, los mismos se recuperan utilizando la librería *va_start* y *va_arg* [5] [4].
- El método *exit* también debe ser declarado y el mismo es llamado por el simulador al finalizar la simulación. Todos los destructores correspondientes deben ser llamados en esta función de forma de liberar correctamente la memoria (de ser necesario).
- El nombre de la clase debe estar escrito todo con minúsculas. **¿porque?**

Como se puede observar existen las funciones *dint* y *dext* que son las funciones exigidas por el simulador para ejecutar correctamente las transiciones interna y externa correspondientes al formalismo DEVS, el modelo Layer ejecuta en esas funciones todo el código relativo a la arquitectura presentada y llama a los métodos virtuales *dinternal* y *dexternal* que son sobre escritos por el modelador para ejecutar el protocolo implementado. De esta forma se consigue abstraer lo más posible al modelador del formalismo, convirtiendo esta herramienta en una herramienta

viable para el uso dentro del ámbito de estudio.

esto responde mi pregunta de antes sobre el ta :-)

Por otro lado, la variable `next_internal` definida en el template Layer es utilizada por el modelo para agendar la próxima transición interna. Si el protocolo no asigna ningún valor a esta variable, se asume que el protocolo tomó infinito tiempo en procesar y el modelo quedará pasivado hasta que llegue nuevos mensajes externos. Una vez que se cumple el `next_time`, el output correspondiente es inmediatamente enviado y luego el simulador queda pasivado a menos que haya más mensajes externos a procesar. El desarrollador del protocolo tiene que asignar en esta variable un modelo del tiempo de computo del protocolo para de esta forma modelar los tiempos reales de runtime¹. Esta variable es de tipo double y la representación del tiempo mediante doubles queda en manos del modelador y debe ser consensuada con el resto de los modeladores de los demás protocolos para poder unirlos en un solo modelo, un consenso común es asumir que una unidad de double representa un segundo y utilizarlo de forma similar a los timestamps.

4.2. Como enviar y recibir mensajes entre las capas

Todos los mensajes que llegan al modelo de una capa son recibidas por uno de los cuatro siguientes puertos dependiendo de donde provenga el mensaje:

- `std::queue< DH > higher_layer_data_in: Input Port 0`
- `std::queue< DL > lower_layer_data_in: Input Port 1`
- `std::queue< CH > higher_layer_ctrl_in: Input Port 2`
- `std::queue< CL > lower_layer_ctrl_in: Input Port 3`

La Figura 2 Muestra de que capa proviene y para que tipo de datos está pensado cada uno de estos puertos.

Como ya fue mencionado y como se puede observar en la definición de las colas, los tipos template instanciados son los tipos de los mensajes recibidos desde las capas superior e inferior. Es importante que los mensajes recibidos sean efectivamente de este tipo, porque el simulador PowerDEVS envía mensajes como `void *` y los mismos son casteados a su tipo correspondientes una vez recibidos por el modelo receptor, por lo que si los tipos no corresponden se generará una excepción en tiempo de ejecución.

Se procesan en dINTERNAL? Yo pensaría que los mensajes entreantes se procesan en dext...

Para procesar los mensajes, dentro del método `dinternal`, los mismos deben ser desencolados, es importante que se desencolen, ya que en caso contrario el mismo seguirá estando en la cola y el modelo volverá a producir una transición interna y llamar de nuevo al método `dinternal` procesando dos o más veces el mismo mensaje. Reprocesar un mensaje múltiple veces probablemente esté mal, pero tal vez es lo deseado por el protocolo, pero hay que tener en cuenta que una de las pocas exigencias del formalismo DEVS es que no ocurran infinitos eventos en un mismo momento del tiempo virtual, y como el modelo sigue generando transiciones internas mientras haya mensajes por procesar, si los mensajes no son nunca desencolados y la variable `next_internal` se mantiene en cero, se generan infinitas transiciones internas y eso produciría un modelo DEVS ilegítimo que cuelga al simulador.

¹Una opción es medir el tiempo real de procesamiento mediante librerías c++ para asignar el valor real de tiempo de computo.

Para enviar mensajes las colas a utilizar son las siguientes:

- message::queue<*DH2*> higher_layer_data_out: Output Port 0
- message::queue<*DL2*> lower_layer_data_out: Output Port 1
- message::queue<*CH2*> higher_layer_ctrl_out: Output Port 2

Podría ser "out"

- message::queue<*CL2*> lower_layer_ctrl_out: Output Port 3

Para encolar un mensaje en la cola de salida correspondiente se debe utilizar el método *message::queue::push(MSG mensaje)*. De la misma forma que el modelo Layer se encarga de encolar los mensajes en la cola de entrada correspondiente dependiendo el puerto por donde llegó el mensaje, el mismo se encarga de desencolar y enviar los mensajes encolados en las colas de salidas por el puerto correspondiente a cada cola y desencolar el mismo.

Nombre	Tipo	Descripción
Module_name	String	El nombre del módulo que utiliza el logger
Max_interface	Int	La cantidad máximas de interfaces de salida que puede demultiplexar.

Tabla 1: Parámetros del modelo demultiplexer

Esto quiere decir que un protocolo necesita saber si el mensaje que esta enviando va a ser demultiplexado o no? Y si es demultiplexado, sab :-(-er el ID del protocolo al que se envía.

Es decir, quien implemente un modelo atomico de protocolo debe saber como es la estructura del coupled del Nodo (interconexion entre layers) :-(- :-)

Por ejemplo: Si tengo implementado el protocolo IP, quiero que se pueda usar tanto por TCP y UDP. Se puede? Al implementar el protocolo (en C++) tengo que saber el indice de donde el protocolo fue puesto en el acoplado, no?

²Para setear parámetros en un modelo hay que hacer clic derecho -> parameters

³En el modelo implementado en este trabajo hay ejemplos de como utilizar el modelo demultiplexer con el tipo de dato message::multiplexed

⁴Un demultiplexer con relación 1 a *k* numera los módulos de salida de 0 a *k* - 1

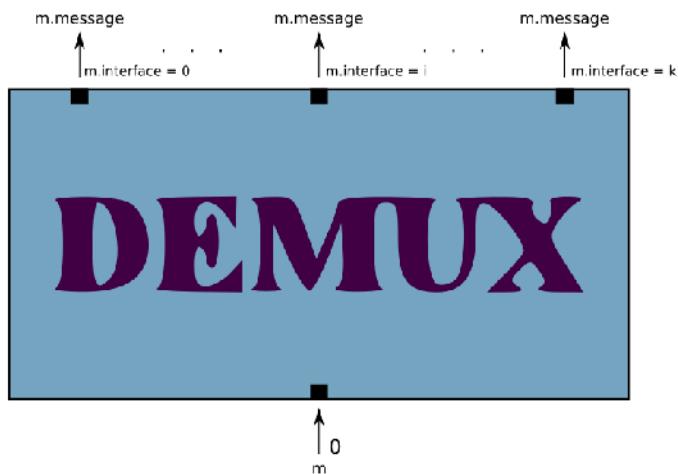


Figura 4: Puertos de entrada y salida de un demultiplexador, mapeo del campo interfaz del mensaje con el puerto de salida correspondiente.

4.3. Como implementar el protocolo de una capa (internal/external functions)

El template cuenta con dos métodos virtuales que están definidos e implementados como funciones vacías (sin comportamiento), estos métodos son los que deben ser sobre escritos para implementar el protocolo. **Siempre que haya mensajes por procesar en las colas de entradas, el modelo llama al método *dinternal*** en el cual se deben procesar dichos mensajes, el modelo Layer mira las colas de salidas para ver si hay mensajes a enviar y los envía, si al finalizar este proceso sigue habiendo mensajes por procesar en la cola de entrada se volverá a llamar a esta función respetando el formalismo DEVS de forma de seguir procesando mensajes. Este ciclo se repite indefinidamente hasta que no haya mas mensajes por procesar, momento en el cual el modelo se pasiva a la espera de nuevos eventos externos.

Que se considera un "evento externo" aca? por que puerto llegaria?

El método virtual *dexternal* existe por si se desea implementar código que se ejecute cada vez que ocurre un evento externo para modificar el estado de las variables internas del protocolo o por algún otro motivo. Esto no es recomendado para desarrolladores no familiarizados con el formalismo DEVS o con el simulador PowerDEVS. **Algun ejemplo de cuando/como se usa?**

Tal vez un diagrama de secuencia ayude a entender cuando Layer llama a *dinternal* y *dexternal*

5. Como agregar una capa

Hay dos tipos de modelos: los modelos atómicos y los modelos acoplados. Para implementar un modelo de capa, primero es necesario heredar del template Layer para obtener el modelo atómico capa como fue descripto en la sección anterior y luego el mismo debe ser insertado mediante la IDE PowerDEVS dentro del modelo acoplado de un dispositivo, luego en el *TOP model* se interconectan todos los modelos acoplados de los dispositivos.

Como se muestra en la arquitectura general, las capas deben ser insertadas dentro de los modelos acoplados de dispositivos (hosts, switches, routers, etc), por lo que antes de agregar la capa al modelo hay que abrir el modelo acoplado dispositivo en caso de existir, o crear el modelo acoplado dispositivo en caso de todavía no existir.

- **Abrir un modelo acoplado en PowerDEVS:** Seleccionar con el mouse el modelo acoplado a abrir y desplegar el menú de opciones del mismo dando click derecho sobre el ícono, luego elegir la opción “open model”.

Me suena que *dinternal* y *dexternal* tienen esos nombre para ser semejantes a las funciones de transicion de DEVS. Pero parecen estar intercambiadas:

en DEVS *dext* recibe mensajes externos. Aca pareceria que los mensajes se reciben por *dinternal*. *dexternal* no entiendo como se usaria.

Lo que estaria faltando es entonces la *dint* (cambios en el modelo por decision propia, sin estimulacion externa). Por ejemplo, en redes se me ocurre cuando TCP retransmite un paquete despues de 200ms. Es algo que decide el modelo por su estado interno sin que le llegue ningun mensaje. ¿En donde se implementaria eso (*dinternal* o *dexternal*) ?

- **Crear un modelo acoplado en PowerDEVS:** Abrir la pestaña “Basic Elements” del sector “library” de la IDE y arrastrar el icono “Coupled” desde el menú hasta el sector donde esta el dibujo del modelo.

La figura 5 muestra la IDE PowerDEVS e indica como crear un modelo acoplado dentro de la misma.

Se podria crear una libreria que ya tenga el modelo host, switch, router implementado

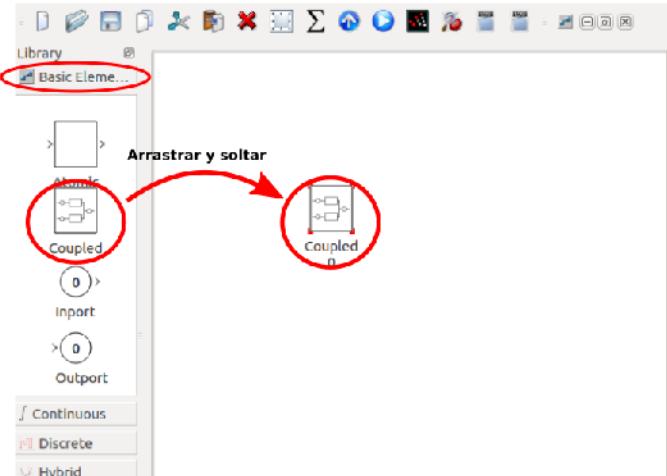


Figura 5: Screenshot de la IDE PowerDEVS con la pestaña “Basic Elements” abierta resaltando el icono de coupled model.

Para insertar un nuevo modelo de capa dentro de un modelo acoplado de dispositivo en PowerDEVS hay que seguir los siguientes pasos:

1. Crear el modelo acoplado de dispositivo (en caso de ser necesario) y abrirlo.
2. Crear un nuevo modelo atómico de capa:
 - a) De la solapa “Basic Elements” arrastrar al escenario el icono “atomic”.
 - b) Click derecho sobre el icono del nuevo modelo atómico en el escenario.
 - c) Elegir la opción “edit”.
 - d) Ir a la pestaña “code”, navegar por los modelos hasta encontrar el archivo .h del modelo capa creado y seleccionarlo.
 - e) Ir a la pestaña “parameters” y agregar todos los parámetros requeridos por el modelo, o sea aquellos que el simulador le va a pasar al método “init” del modelo para inicializarlo y que serán leídos usando *va_arg*⁵.
 - f) Ir a la pestaña “properties” y realizar las siguientes tareas: Escribir el nombre deseado para el nuevo modelo, setear cuatro puertos de entrada y cuatro puertos de salida y elegir el icono a mostrar en el modelo (Elegir un icono no es obligatorio)
3. Conectar las salidas de los modelos superiores e inferiores con las entradas del nuevo modelo según lo explicado anteriormente en la sección *Como enviar y recibir mensajes entre las capas*.
4. Conectar las salidas del nuevo modelo a las entradas correspondientes de los modelos de las capas superiores e inferiores según lo explicado anteriormente en la sección *Como enviar y recibir mensajes entre las capas*.

⁵Estos parámetros dependen de la implementación particular del protocolo.

Poner un screenshot de como deberia quedar

Información más detallada sobre como usar PowerDEVS puede ser encontrada en el documento [1], allí los autores describen como funciona el simulador y como crear un modelo desde el principio.

6. Input/Output - Modelos generadores y vertederos

Para poder alimentar al modelo con eventos, es necesario contar con unos modelos específicos que sepan leer archivos, parsearlos y generar eventos para enviar a los otros modelos. Estos modelos son llamados generadores o *sources* en inglés y si bien no forman parte del modelado del fenómeno, los mismos son una abstracción del mundo externo del modelo y permite modelar dicha interacción.

6.1. Input

Existen dos módulos responsables de leer el o los inputs del modelo, estos son: El modelo atómico template *input_stream* y la clase *Parser*. La idea de separar el input en dos módulos es la de atacar dos problemáticas distintas pero similares:

- Contar con un parser que sepa leer archivos utilizados para inicializar modelos que requieren parámetros muy grandes⁶. **Se pueden usar archivos scilab y leer de ahí.**
- Contar con un modelo generador que utilice el parser y se encargue de ir insertando los estímulos externos al modelo en el tiempo virtual correspondiente. **Pero tal vez esta implementación le sirve a MatiRe**

6.2. La clase Parser

Esta clase está diseñada como template y sirve para leer las líneas de un archivo y parsearlas con el fin de devolverlas en la estructura de datos correspondiente según la instanciación realizada. El *typename INPUT* indica el tipo del input a parsear y el mismo debe implementar el operador `std::istream operator>>(std::istream& INPUT& i)`. Es importante que este operador esté implementado de forma que lea todos los atributos necesarios sin saltos de linea ya que el parser lee un input por linea.

La clase parser cuenta con dos métodos distintos para leer los datos del archivo:

- *next_timed_input*: La linea del archivo a parsear debe comenzar con un valor *double* que será devuelto por el parser de forma separada al input y que representa el tiempo en que debe ser insertado el input en la simulación, luego debe haber un espacio y el resto de la sintaxis requerida por el operador `>>` del tipo de datos a parsear.
- *next_input*: No hace falta explicitar el tiempo como primer elemento de la linea y solo se devuelve el dato parseado por el operador `>>`.

Nota: Si bien el Parser tiene la opción para parsear el input junto con un tiempo asociado, el mismo no hace más que devolverlo en una tupla (tiempo, mensaje).

La clase Parser puede ser utilizada de forma independiente al modelo que lee input siempre que sea necesario leer y parsear las líneas de un archivo. La interfaz y documentación técnica del Parser está en formato Doxygen.

Poner un ejemplo de archivo que puede ser parseado!

⁶como por ejemplo las inicializaciones de las diferentes tablas utilizadas por el modelo (routing/forwarding tables)

6.3. El modelo input stream

El modelo input stream es un modelo handler que se encarga de manejar el parser de forma de ir leyendo el input e ir insertándolo en el momento correcto de la simulación. Para esto, el modelo está implementado como template de forma de poder funcionar con distintos tipos de inputs, el parámetro template es el mismo que se utiliza para iniciar el parser y tiene las mismas precondiciones.

Para utilizar este modelo, hay que crear una clase `input_stream_<tipo del input>` que herede de `input_stream`⁷ instanciando el parámetro template del tipo de datos del input. Dado que el simulador PowerDEVS requiere que los modelos tengan un archivo `.cpp` el mismo debe existir y puede simplemente estar vacío.

Una vez creada la herencia, el modelo atómico está listo para ser utilizado, el mismo no tiene ningún puerto de entrada y un solo puerto de salida por donde salen los inputs generados, este puerto de salida es el que debe ser conectado al modelo que precisa del input. Un ejemplo de esto se encuentra disponible en el modelo del caso de estudio presentado con el nombre `domain_name_source.h`.

Los parámetros a setear del mismo en la IDE de PowerDEVS son los que se muestran en la Tabla 2.



Nombre	Tipo	Descripción
Module name	String	El nombre del módulo que utiliza el logger
Input	String	El path al archivo donde está guardado el input a parsear

Tabla 2: Parámetros del modelo `input_stream`

Vi que en otros modelos tambien estaba este parametro
En el `.cpp` de un atomico se puede hacer `this->getName()` y eso devuelve el nombre del modelo (el que se puso en la IDE).
¿Eso no alcanza? como para no tener un parametro menos por modelo que adamas imagino tiene que ser unico

6.4. El modelo output stream

De la misma forma que es necesario contar con modelos generadores, también se necesita contar con modelos vertederos o *sink* en inglés. Estos modelos sirven para absorver todo el output generado por el modelo y guardarlos en un archivo. El mismo está implementado de la misma forma que el modelo input stream y se hereda tomando las mismas consideraciones. Las únicas diferencias son las siguientes:

- No cuenta con ningún puerto de salida y cuenta con un puerto de entrada.
- Tiene dos tipos de datos template, uno para el tipo de output que le llega por el puerto 0 y otro para el que le llega por el puerto 1. Esto es así ya que la arquitectura de capas introducida en este trabajo envía mensajes de tipo datos por el puerto 0 y mensajes de tipo control por el puerto 1. Para que el modelo vertedero pueda aceptar ambos mensajes se utilizó este mecanismo.

Un ejemplo de esto se encuentra disponible en el modelo del caso de estudio presentado con el nombre `domain_name_source.h`. Los parámetros a setear del mismo en la IDE de PowerDEVS son los que se muestran en la Tabla 3.

⁷El archivo header siempre tiene que ser de tipo `.h` y la implementación debe estar en un archivo `.cpp`

Nombre	Tipo	Descripción
Module name	String	El nombre del módulo que utiliza el logger
Output	String	El path al archivo donde está guardado el input a parsear

Tabla 3: Parámetros del modelo input_stream

7. Logger

La clase Logger sirve para ir creando logs de la simulación. Esta clase separa los logs en cuatro tipos:

- **LOG:** Logs que sirven para documentar eventos no relevantes de la simulación.
- **INFO:** Logs utilizados para documentar eventos relevantes de la simulación.
- **DEBUG:** Logs utilizados para documentar eventos que sirven de ayuda para tareas de debug pero que no tiene ningún otro interés.
- **ERROR:** Logs utilizados para documentar errores producidos en tiempo de ejecución.

Para habilitar y deshabilitar los distintos tipos de logs a guardar en una simulación solo hay que comentar y descomentar las líneas 4, 5, 6 y 7 del archivo *logger.h*, de esta forma en el código se logean todos los eventos correspondientes a estas cuatro categorías y luego antes de cada simulación se decide que mensajes mostrar y cuales no.

Lineas a descomentar y comentar para habilitar y deshabilitar los distintos tipos de logs correspondientemente:

```
#define show_log
#define show_info
#define show_debug
#define show_error
```

Para utilizar esta clase, cada modulo tiene que crear una variable de tipo *Logger* e inicializarla pasándole como parámetro un *string* con el nombre del módulo o modelo del cual el logger va a loggear eventos y luego utilizarlo siguiendo su interfaz. La inicialización de la variable puede ser en su constructor o mediante el método *Logger::setModuleName(std::string other_module_name)*. La documentación de la interfaz está en formato Doxygen.

La clase template *Layer*, ya cuenta con una variable de tipo *textitLogger* que debe ser inicializada en el método *init*.

Este logger tal vez tambien esta bueno para reutilizarlo en otros modelos

8. Caso de estudio

Además de implementar un framework para el estudio de las redes, en este trabajo se introduce un modelo sencillo del protocolo UDP/IP, el mismo cuenta con los protocolos necesarios para que la red funcione de forma que los mensajes van pasando a través de las distintas capas para llegar a ser enviados de un dispositivo a otro. La Figura 6 muestra un diagrama de los protocolos implementados.

8.1. DNS

DNS (Domain Name System) es un sistema de indexación mediante el cual se puede transformar un nombre de dominio en una dirección *IP* del host correspondiente. Este sistema es muy utilizado ya que para los humanos es muy difícil recordar direcciones en formato *IP* pero no en formato *Domian Name*. Así como el sistema de indexación de una guia telefónica indexa por apellido

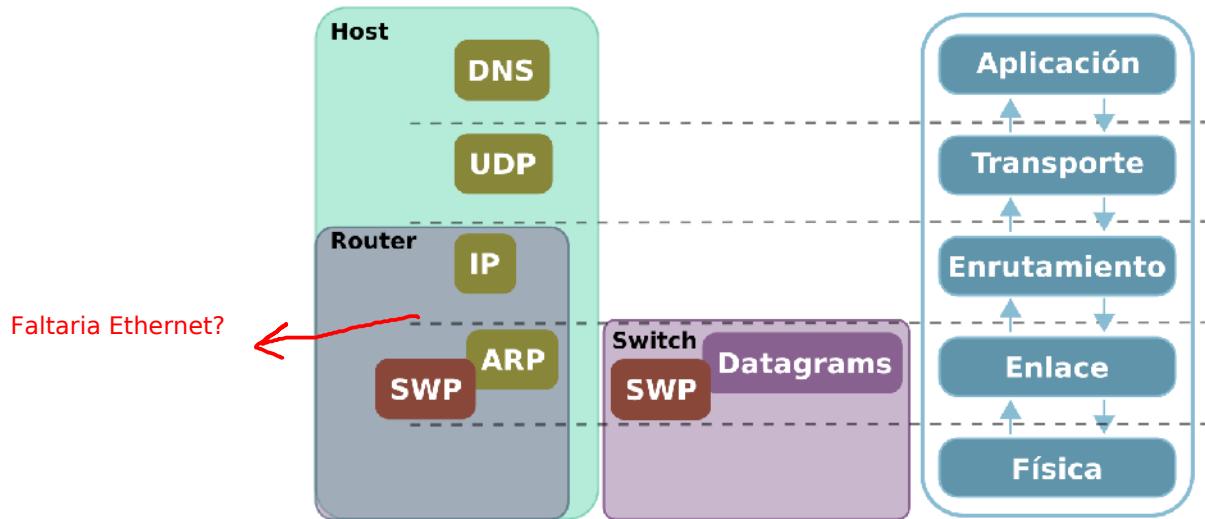


Figura 6: Protocolos implementados separados por capas y dispositivos que los utilizan.

y nombre para conseguir el número de teléfono de las persona, cuando se cuenta con un nombre de dominio se debe utilizar *DNS* para conseguir la dirección *IP* del host. Este sistema pertenece a la capa de aplicación y es utilizada por diversas aplicaciones para facilitar la comunicación con aplicaciones remotas. Ejemplos de esto son los browsers donde una persona escribe el nombre de dominio de un servidor y *DNS* lo transforma en *IP* para poder acceder al mismo donde se encuentra alojada la plataforma web.

¿Como se ve esto en PowerDEVS?

8.1.1. Estructuras de datos correspondientes

La estructura de datos utilizada para el sistema de *DNS* es *dns::Packet* y cuenta con los siguientes campos:

- **header:** Es de tipo *dns::Header*, cuenta entre otras cosas, con toda la información necesaria para saber cuantos *dns::ResourceRecord* hay en cada una de los siguientes campos. La Figura 7 muestra esta estructura de datos.
- **questions:** Una lista de *dns::ResourceRecord* que almacena el nombre de dominio del cual se desea obtener el *IP* asociado.
- **answers:** Una lista de *dns::ResourceRecord* que almacena la respuesta obtenida mediante *DNS*.
- **authoritatives:** Una lista de *dns::ResourceRecord* que almacena la dirección *IP* de servidores *DNS* mediante los cuales se puede obtener la dirección pedida.
- **additionals:** Una lista de *dns::ResourceRecord* donde se guarda la dirección ip del host que mandó el request. De esta forma no necesita pedirle a la capa *UDP* por el ip que mandó el *request*.

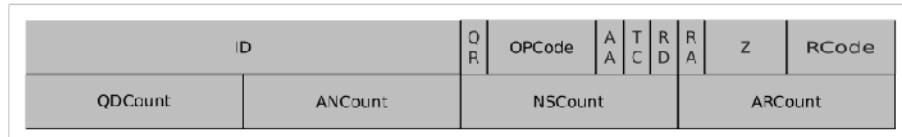


Figura 7: Estructura del tipo de datos *dns::Header* alineada a 4 bytes.

La Figura 7 muestra la estructura del header de un paquete dns, en el mismo se ven los campos ***QDCount***, ***ANCount***, ***NSCount*** y ***ARCount*** que indican la cantidad de dns::ResourceRecords en cada una de las secciones ***questions***, ***answers***, ***authoritatives***, ***aditionals*** respectivamente. No todos los campos son utilizados en la implementación realizada en este trabajo, la utilización de los mismos será explicada a continuación junto con la explicación del funcionamiento del protocolo. Información detallada sobre cada campo del header puede ser encontrada en [2]² decir que respeta el formato de headers del protocolo DNS, poniendo una referencia al RFC

dns::ResourceRecord es la estructura de datos encargada de guardar información de pares de pregunta/respuesta. Si bien la pregunta es siempre de tipo dns::DomainName, la respuesta puede no siempre ser de tipo *IP*, en algunos casos como en los renombrados de dominios, o en las consultas no recursivas, la respuesta puede ser de tipo dns::DomainName.

Los campos de la estructura dns::ResourceRecord son los siguientes:

- ***name***: Es de tipo dns::DomainName y guarda el nombre de dominio del cual se quiere conocer su *IP*.
- ***QType***: Indica el tipo de respuesta; A = 1 representa respuestas de servidores autoritativos y son de tipo *IP*, NS = 2 representa redirecciones a un servidor autoritativo para una determinada zona y son de tipo dns::DomainName.
- ***Avalue***: Si la respuesta es de tipo A, este campo guarda el *IP* correspondiente a la respuesta.
- ***NSValue***: Si la respuesta es de tipo NS, este campo guarda el dns::DomainName correspondiente a la respuesta.
- ***QClass***: Este campo indica el protocolo a utilizar; En este trabajo y en la mayoría de los casos, este campo se mantiene siempre con el valor IN = 1 = Internet protocol.
- ***TTL***: Es el tiempo de vida (generalmente en segundos) utilizado para saber cuánto tiempo se puede mantener un recurso obtenido por un pedido *DNS* en la memoria cache antes de considerar que es invalido y eliminarlo.

Por último dns::DomainName es una estructura que sirve para guardar nombres de dominios como por ejemplo “*network.devs.com*”. El mismo cuenta con métodos que permiten leer, escribir, imprimir y trabajar fácilmente con la jerarquía de zonas de los mismos. Por ejemplo, para saber si un nombre de dominio pertenece a la zona de otro nombre de dominio, se puede hacer lo siguiente:

```
dns::DomainName("com").isZoneFor("network.devs.com").
```

8.1.2. Funcionamiento del protocolo

cliente El protocolo implementado en este trabajo para el cliente es muy simple, las consultas comienzan con un dns::Packet que tiene un solo RR (dns::ResourceRecord) en la sección *questions* que contiene el nombre de dominio del cual se desea saber el *IP*. El Protocolo comienza buscando el nombre de dominio del RR en el cache y si lo encuentra devuelve el *IP* asociado y terminó la consulta. En caso de que no lo encuentre los siguientes pasos son realizados:

1. Le asigna un nuevo ID al dns::Header del dns::Packet de la consulta.
2. Le asigna el valor QR = 1 para indicar que es una query y no un answer.
3. Le asigna el valor RD = 1 para indicar que desea que la consulta se resuelva de forma recursiva, si el servidor no es recursivo este campo será ignorado y enviará una redirección a otro servidor.
4. Guarda el paquete en una tabla de consultas realizadas localmente (i.e. consultas realizadas por el host local).

5. Agrega un RR A la sección *additionals* que contiene el IP local para que el servidor sepa a quien devolver la respuesta.
6. Envía el paquete al servidor local para que el mismo resuelva el pedido.

Servidor recursivo Cuando un servidor recursivo recibe un dns::Packet con el campo QR = 1, el mismo intenta primero resolver la consulta buscando en la tabla de nombres de dominios para los cuales el es autoritativo. envía la respuesta realizando los siguientes pasos:

1. Al paquete entrante le cambia el campo QR = 0 para indicar que es una respuesta.
2. Asigna el valor AA = 1 para indicar que la respuesta proviene de un servidor autoritativo para ese nombre de dominio.
3. Le agrega un RR en la sección *answers* con *Qtype* = A, con el nombre de dominio de la consulta y en *AValue* el IP correspondiente al nombre de dominio.
4. Envía el dns::Packet de retorno al servidor que envió el *request*, el IP del mismo se encuentra en el RR de la sección *additional*.

En caso de no encontrar la respuesta en su lista personal, si el paquete dns tiene el campo RD = 1 el servidor realiza los siguientes pasos con la finalidad de conseguir la respuesta mediante otros servidores de forma recursiva:

1. Guarda el paquete en una tabla de consultas realizadas remotamente (i.e. consultas realizadas por un *host* remoto que llegó a través de paquetes dns).
2. Cambia el RR de la sección *additionals* para asignarle el *IP* local para que el servidor al cual reenvíe la consulta sepa como enviarle la respuesta.
3. Envía el paquete al servidor TLD que conoce.

Una vez reenviada la consulta se queda a la espera de una respuesta de del servidor. Si el servidor es autoritativo para ese nombre de dominio ó es un servidor recursivo, le enviará una respuesta final con el *IP* correspondiente que será reenviado al *host* del cual llegó la consulta. Si el servidor no es ni autoritativo para la consulta ni recursivo, entonces va a devolver una redirección a un servidor autoritativo para esa zona, cuando esto ocurre, en el paquete dns de respuesta vienen dos RRs, uno en la sección *answers* de tipo NS que contiene el nombre de dominio de dicho servidor y otro en la sección *authoritatives* de tipo A con el *IP* de dicho servidor. En este caso el paquete es reenviado a dicho servidor con la esperanza de que esté resuelva la consulta o vuelva a enviar una redirección.

Note: El *IP* de dicho *host* está en el paquete dns que se guardó anteriormente, y la forma de recuperarlo es utilizando el ID de la consulta que se mantiene igual durante todo el proceso.

Note: En caso de que el servidor envíe un paquete dns de error, el mismo es reenviado al *host* que realizo la consulta y el protocolo falló. Esto puede ocurrir porque no se conoce de ningún servidor autoritativo para la zona del nombre de dominio consultado.

Note: El paquete enviado como respuesta final es un paquete que llega desde un servidor al cual se le realizó la consulta, y por ende el campo AA que indica si la respuesta es autoritativa o no, depende de dicho servidor.

En caso de que el campo RD esté en 0, el servidor se comporta como un servidor iterativo

Servidor iterativo Cuando una consulta llega a un servidor iterativo, en caso de ser autoritativo para la consulta, envía el paquete previamente transformado en respuesta con el *IP* correspondiente a la consulta asignando $AA = 1$ y termina. En caso de no ser autoritativo, el servidor busca en la tabla de *DN servers* un servidor que sea autoritativo para la zona del nombre de dominio consultado, en caso de haber más de uno, elige el servidor más puntual y lo envía como respuesta de redirección. Para esto realiza los siguientes pasos modificando el paquete de la consulta para mantener el ID correcto.

1. Agrega un RR de tipo NS en la sección *answers* donde *name* contiene el nombre de dominio para el cual el servidor es responsable de todos sus sub-dominios (i.e. es responsable de esa zona), y en *NSValue* el nombre de dominio del servidor.
2. Agrega un RR de tipo A en la sección *authoritatives* donde *name* es igual al campo *NSValue* del RR mencionado en el punto anterior y en el campo *AValue* el *IP* de dicho servidor.
3. Asigna $AA = 0$ para marcar que el no es autoritativo para dicha consulta.
4. Utiliza el RR de la sección *additionals* para enviar el paquete con la respuesta al host que realizó la consulta.

note: un servidor *s1* es más puntual para una zona que otro servidor *s2* si y solo si *s1* pertenece a la zona de *s2*, por ejemplo “*devs.com*” es más puntual que “*com*” ya que “*devs.com*” pertenece a la zona “*com*”, con lo cual, si hay una consulta “*algo.devs.com*” el servidor que tiene la respuesta final es “*devs.com*” y no “*com*” quien va a devolver una redirección a “*devs.com*”.

No me queda claro como se usan los 2 tipos de puertos de las capas (de datos y de control).

¿Que mensajes van por cada uno?

8.2. UDP

8.2.1. Estructuras de datos correspondiente a UDP

El protocolo UDP implementado cuenta con los siguientes tipos de datos, los cuales tienen documentación en formato Doxygen en el código y que se encuentra adjunta en el apéndice de este documento:

referencia al RFC

- *udp::Segment*: Representa un segmento de datos para ser enviado por la red, el mismo cuenta con pseudoheader, header y payload como se muestra en la Figura 8.
- *udp::Control*: Es la estructura de datos utilizada para la comunicación con la capa de aplicación, en la misma se especifica que comando envía la aplicación a la capa UDP. Toda la información a enviar llega al protocolo UDP a través de esta estructura.

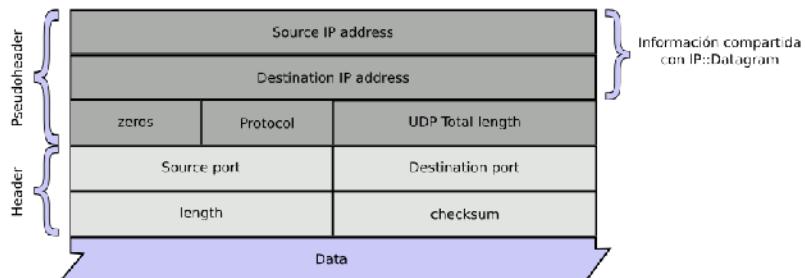


Figura 8: Estructura del tipo de datos *udp::Segment* alineada a 4 bytes.

Estaria bien un diagrama de como los mensajes van bajando y subiendo de capas en este caso (como se comunican entre los protocolos DNS, UDP, Datagram, etc del cliente y los servidores)

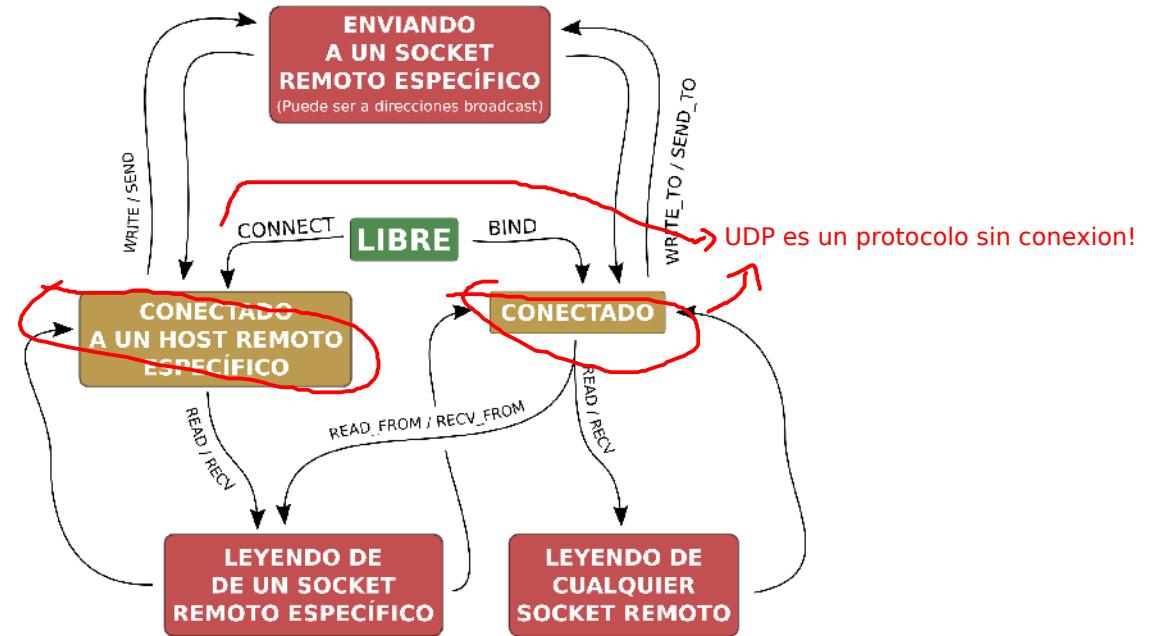


Figura 9: Máquina de estados del protocolo UDP para un socket determinado. Por prolíjidad, el comando *CLOSE* no aparece, el mismo corta cualquier estado y vuelve automáticamente a *LIBRE*.

8.2.2. Comportamiento del protocolo

El protocolo UDP no cuenta con ningún sistema de detección de fallas en la entrega de los mensajes ya que el mismo está pensado para funcionar de forma que optimice la velocidad de entrega, para esto se basa en un mecanismo de mejor esfuerzo. Esto lo convierte en un buen protocolo cuando la eficiencia es un atributo clave. Por otro lado, dado que no controla que los mensajes lleguen o no a su destino, es necesario contar con otro mecanismo que detecte mensajes faltantes (aquellos que no llegaron a destino) o de lo contrario, debe permitirse la perdida de los mismos. Si bien no se realiza ningún esfuerzo por detectar si el mensaje llega o no a destino, UDP implementa checksum para detectar errores en la codificación del header del segmento, esto es implementado con la finalidad de no entregar mensajes a destinatarios incorrectos. Cuando un error es detectado en el header, el segmento es directamente descartado⁸.

UDP funciona con sockets, los sockets son una combinación de IP y puerto que permiten enviar y entregar los mensajes a la aplicación correspondiente dentro del host de destino. De esta forma, para que un mensaje sea correctamente entregado, el host destino tiene que tener una aplicación que esté escuchando por el puerto al cual fue enviado el mensaje, si este no es el caso, el mensaje es simplemente descartado. Los sockets también cumplen el rol de evitar colisiones entre las aplicaciones ya que las mismas **no pueden conectarse a un puerto mediante un socket si el puerto ya está siendo utilizado por otra aplicación**, ya sea para enviar o recibir mensajes. La Figura 9 muestra la máquina de estados del protocolo UDP implementado en este trabajo.

Esto no es necesariamente cierto. Varias aplicaciones pueden mandar mensajes UDP por el mismo <IP, port> y varios servers pueden estar escuchando el mismo <IP, port>.

En TCP no se puede, pero en UDP (que no mantiene conexiones) si se puede.

⁸No se comprueba que el payload contenga errores en su codificación, por lo que el dato en sí, podría llegar a destino con errores y nunca serían detectados por UDP.

En el modelo implementado, los mensajes de Control contienen un campo *request* que indica que operación se desea efectuar (connect, bind, read, etc) también es necesario enviar el IP y puerto del socket y el id de la aplicación, esta información es siempre requerida, ya sea para crear un nuevo socket atado a la aplicación del id o para corroborar que el socket existe y que corresponde a la aplicación que envía el mensaje de control. Dependiendo de la operación, es preciso completar correctamente los campos restantes. A continuación se explica cada operación y que campos requiere cada una:

- **CONNECT:** *remote_ip, remote_port*. De esta forma UDP sabe que cada vez que esta aplicación envíe o reciba un paquete, el mismo será para o desde el socket con IP *remote_ip* y puerto *remote_port*. Esta operación falla si el socket ya existe.
- **BIND:** Esto permite a la aplicación enviar paquetes a distintos destinos, especificando siempre el socket del destino mediante los comandos WRITE_TO y SEND_TO y recibir mensajes de distintos orígenes mediante los comandos READ_FROM, RECV_FROM, REC o READ. Esta aplicación falla si el socket ya existe.
- **READ_FROM / RECV_FROM:** *remote_ip, remote_port*, estos campos son utilizados para saber que mensajes entrantes deben ser aceptados y cuales no. Cuando este comando es recibido por UDP, el socket correspondiente es marcado como “En espera de mensajes” y el mismo se mantiene en ese estado hasta que llega un mensaje del host remoto indicado, en ese momento el mensaje es entregado y el socket vuelve al estado anterior. Este comando solo puede ser utilizado en un socket que está en estado *BOUND*, o sea, un socket que fue previamente bindeado.
- **READ / RECV:** Es lo mismo que el item anterior pero con la diferencia de que no se especifica un socket remoto. Si el socket local fue bindeado se aceptan todos los mensajes entrantes a este socket y si fue conectado se aceptan solo los mensajes del socket remoto indicado en el momento de la conexión.
- **WRITE_TO / SEND_TO:** Lo mismo que READ_FROM / RECV_FROM pero para enviar mensajes, con lo cual el campo *packet* debe contener el paquete a enviar.
- **WRITE / SEND:** Lo mismo que READ / RECV pero para enviar mensajes, con lo cual el campo *packet* debe contener el paquete a enviar.
- **CLOSE:** Esta operación remueve el socket, dejando el IP y puerto libres para futuras aplicaciones que quieran conectarse o bindearse al mismo.

Estas no son las operaciones de UDP el RFC. (ver sección "User Interface" del RFC 768)

Cada vez que el protocolo UDP quiere enviar un paquete, crea un nuevo *udp::Segment*, asigna el paquete en el campo *payload* en formato *char ** de no más de 512 bytes de tamaño, completa los campos correspondientes del header y pseudoheader, calcula el checksum utilizando el header ya completado, guarda el valor obtenido en el campo *checksum* y envía el mensaje a la capa inferior por el canal de datos encolando el mismo en la cola *lower_layer_data_out*.

Cada vez que el protocolo UDP recibe un mensaje de la capa inferior se realizan un par de chequeos, en caso de que el mensaje pase los controles, el *payload* del segmento recibido es entregado a la aplicación correspondiente, en caso contrario el segmento es descartado. Un segmento pasa los chequeos si se cumplen las siguientes condiciones:

- El campo *checksum* debe coincidir con el cálculo de checksum que realiza nuevamente el protocolo sobre el header del segmento recibido.
- El socket destino del segmento debe ser un socket local existente.
- El socket local debe estar en un estado de “En espera de mensajes”.

- El socket local debe estar esperando mensajes del socket remoto o esperando mensajes de cualquier socket remoto.

La estructura de datos *Socket* es utilizada para manipular esta combinaciones de ip-puerto, la misma cuenta con los siguientes campos.

- **local_port:** Es de tipo *uint_16* y representa el puerto del host local al cual se conecta o bindea el socket.
- **remote_port:** Es de tipo *uint_16* y representa el puerto del host remoto al cual se desea enviar y/o recibir datos. Este campo es opcional, no siempre se especifica un host remoto, por ejemplo, cuando se desea leer datos de cualquier host remoto, este campo no debe especificarse.
- **local_ip:** Es de tipo *IPv4*⁹ y representa el ip del host local al cual se conecta o bindea el socket.
- **remote_ip:** Es de tipo *IPv4* y representa el ip del host remoto al cual se desea enviar y/o recibir datos. Este campo es opcional, no siempre se especifica un host remoto, por ejemplo, cuando se desea leer datos de cualquier host remoto, este campo no debe especificarse.
- **status:** Es de tipo enum e indica en que estado se encuentra el socket. Los estados son los mencionados previamente cuando se explicaron los comandos.

La Tabla 4 muestra los parámetros del modelo UDP en la IDE PowerDEVS.

Nombre	Tipo	Descripción
Module name	String	El nombre del módulo que utiliza el logger
ip_amount	Int	La cantidad de ip locales, generalmente este valor es 1
ip 1	IPv4	La primer ip local
:	:	:
ip n	IPv4	La última ip local

Tabla 4: Parámetros del modelo UDP

8.3. IP

El protocolo IP es el encargado del enrutamiento de los ip::Datagrams de un nodo (router, host) a otro con la finalidad de llegar al host destino. La estructura de datos ip::Datagram es la utilizada para encapsular los udp::Segments y cuenta con un campo *header* de tipo ip::Header y un campo *data* de tipo udp::Segment, la Figura 10 muestra la estructura del tipo de datos ip::Header.

El protocolo IP implementado en este trabajo **funciona con UDP** y se puede dividir en dos categorías:

- **Enrutamiento en un host:** Los hosts son los encargados de encapsular los udp::Segments provenientes de la capa superior (UDP) dentro de ip::Datagrams y enviarlos al router o host directamente conectado al mismo dependiendo de si el IP de destino corresponde a un host de la misma subnet o a un host en otra subnet. **Si un host recibe un ip::Datagram cuyo IP destino no corresponde con ningún IP del host**, el ip::Datagram no es forwardeado a nadie y es descartado, en cambio si el IP corresponde con algún IP del host, el udp::Segment del ip::Datagram recibido es desencapsulado y entregado a la capa superior.

⁹El tipo de datos *IPv4* está documentado en formato Doxygen en el código

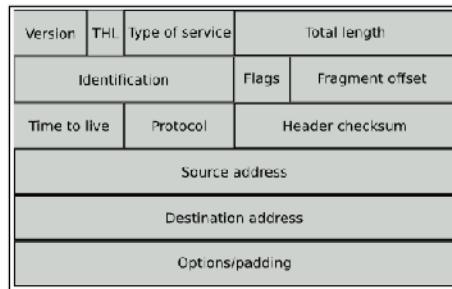


Figura 10: Estructura del tipo de datos de ip::Header alineada a 4 bytes.

- **Enrutamiento en un router:** Los routers por el contrario no encapsulan udp::Segments ya que no existe capa de transporte en un router, los mismos se encargan de forwardear ip::Datagrams que llegan.

Si bien hay diferencia entre los hosts y los routers, el proceso de forwardear ip::Datagrams es el mismo para ambos, este proceso consiste en buscar el mejor *nexthop* dentro de la “routing table” y enviar el ip::Datagram al mismo con la esperanza de que este sepa como entregar el mensaje al destino.

Como sabe el modelo que implementa IP si esta dentro de un host o un router/switch?

En este trabajo no se implementaron clases de direcciones IP A, B y C, por el contrario se implementó el mecanismo de *subnetting* mediante el uso de *netmasks*, esto significa que se utiliza una máscara que indica cuantos bits de la izquierda del IP corresponden al network. La máscara utilizada es una dirección IP que tiene en los bits de la izquierda unos y en los bits de la derecha ceros, la cantidad de unos de la máscara determinan cuantos bits del IP deben ser considerados como network.

referencia a routing tables? libro de la materia

La *routing table* contiene los siguientes campos:

- **Network:** IP del network asociado a esta entrada.
- **Netmask:** Máscara utilizada para saber que parte del IP network debe ser comparada con el IP destino para determinar si el IP destino corresponde a la misma red a la cual se puede llegar mediante este netxhope.
- **Nexthope:** IP del nodo al cual hay que enviarle el ip::Datagram de forma que el mismo sepa como entregar el ip::Datagram. El nexthope debe estar directamente conectado con el nodo actual. El mismo puede ser un host, en este caso, el destino es directamente alcanzable y el mensaje es enviado al mismo.
- **Metric:** Una métrica de a que distancia (en termino de nodos intermedios) se encuentra el destino del router o host actual.
- **Description:** Una descripción que ayude a comprender el rol del nexthope. Esto puede ser el nombre del mismo.

Para determinar cual *nexthope* es el mejor se utiliza el siguiente criterio: Se elige aquel *nexthope* con prefijo de network más pequeño, si hay más de uno con menor prefijo, se elige aquel con menor métrica, o sea, aquel para el cual se espera que el camino sea menor, si sigue siendo necesario desempatar, se elige el primero de la tabla de arriba para abajo.

Un dispositivo puede tener más de una interfaz, cada interfaz está conectada con una network particular y el *nexthope* elegido pertenece a alguna de esas networks, para saber por que interfaz enviar el ip::Datagram se utiliza la *forwarding table* la cual indica para cada *nexthope* su interfaz

asociada dependiendo de en que network se encuentre el *nexthope*.

Dado que en este trabajo **no se implementó ningún algoritmo dinámico para completar las *routing table* y *forwarding table***, si las mismas no son correctamente inicializadas a mano, puede suceder que un paquete nunca llegue a destino o que el mismo se quede dando vueltas en círculos. Para evitar que se generen ciclos infinitos que ocasionarían una gran congestión de la red, se utiliza el campo ***TTL* (Time To Live)** el cual arranca con el valor 255 cuando el host envía el paquete al primer *nexthope* y es decrementado en uno cada vez que pasa por un nuevo router, de esta forma, cuando el *TTL* llega a 0 el ip::Datagram es descartado en vez de ser forwardeado. Este tipo de problemas puede suceder también cuando las *routing tables* son completadas dinámicamente.

Para poder enviar un ip::Datagram al *nexthope* o al host en cuestión, **el protocolo IP le manda un mensaje de control a la capa de linko para pedirle que准备 la MAC address asociada al IP del *nexthope***, una vez que recibe el mensaje de la capa de linko confirmando que dicha MAC address está cacheada, el protocolo IP pasa el ip::Datagram a la capa de linko para que el mismo lo envíe. El mensaje de confirmación de que la *MAC address* existe no asegura que el mensaje sea enviado, en caso de que un error acontezca en el protocolo ARP, el mismo se lo comunica a IP que se queda esperando una nueva confirmación del protocolo ARP comunicando que el envío puede ser realizado.

En general ARP es usado por Ethernet. IP no sabe nada de direcciones MAC.

¿En esta implementacion, que se hace con las direcciones MAC? ¿Viajan en el paquete IP?

Los parámetros del modelo IP en la IDE de PowerDEVS son los que muestra la Tabla 5

Nombre	Tipo	Descripción
Module name	String	El nombre del módulo que utiliza el logger
ip_amount	Int	La cantidad de ip locales, generalmente este valor es 1
ip 1	IPv4	La primer ip del host
:	:	:
ip n	IPv4	La última ip del host
file_path_rt	String	Path al archivo que contiene las entradas de la routing table
file_path_ft	String	Path al archivo que contiene las entradas de la fowarding table

Tabla 5: Parámetros del modelo IP

8.4. ARP

Si bien los IPs cuentan con una estructura jerárquica que permite describir la topología de la red y de esa forma agrupar a los hosts por networks, los mismos son asignados dinámicamente dependiendo de en donde está conectado el host. Por el contrario, las *MAC addresses* son fijas, y cada dispositivo de red viene con una *MAC address* de fábrica, la misma se supone que es única. Para poder enviar datos de un dispositivo a otro, es necesario primero conseguir su *MAC address*, el protocolo ARP (*Address Resolution Protocol*) es el encargado en este modelo para realizar este trabajo.

Referencia al RFC de ARP

Los ip::Datagram que deben ser enviados entre dos nodos adyacentes de la red son encapsulados dentro de un link::Frame, el cual cuenta con la estructura mostrada en la Figura 11.



Figura 11: Estructura de datos de un link::Frame.

¿Que tiene que ver esto con el protocolo ARP? -> Esto pareceria ser mas bien Ethernet..

El campo payload es el que contiene al ip::Datagram, el mismo admite un tamaño máximo de 1500 bytes, por lo cual todos los paquetes enviados en este modelo deben tener un tamaño menor a 1500 bytes ¹⁰. El campo *MAC_destination* es el que contiene el *MAC address* del dispositivo de destino al cual se desea enviar el frame.

El protocolo ARP cuenta con una *ARP table*, la cual tiene las siguientes columnas:

- **ip:** El IP del dispositivo al cual se debe enviar el link::Frame, este IP es el que utiliza la capa IP.
- **mac:** La *MAC address* del dispositivo al cual se debe enviar el link::Frame.
- **timeout:** Es el periodo de tiempo de validez que tiene el dato. Cada vez que se agrega una fila a la *ARP table*, al mismo se le asigna un *timeout*, y cuando el mismo se acaba, la fila es eliminada de la tabla.

Cada vez que el protocolo IP envía un pedido para enviar un ip::Datagram, el protocolo ARP busca el IP en la *ARP Table*, si hay una entrada con ese IP, ARP manda un mensaje de control a la capa IP comunicando que el envío puede ser efectuado, si por el contrario, no hay una entrada con ese IP, ARP envía un link::arp::Packet con un pedido de *MAC* a todos los dispositivos de la red directamente conectados con el. Todo dispositivo que recibe un link::arp::Packet con pedido de *MAC* cuya dirección IP corresponda con la requerida, enviará una respuesta con su *MAC address*, luego, el dispositivo que envió el pedido, al recibir la respuesta agrega la nueva entrada a la *ARP table*. Una vez que se obtuvo la *MAC address*, ARP envía el mensaje de control a la capa IP comunicando que el envío puede ser efectuado. **Si ningún dispositivo de la red matchea con el IP solicitado, entonces el ip::Datagram nunca será enviado pues el protocolo ARP se quedará hasta el infinito esperando la respuesta de la red, esto no implica que el protocolo quede bloqueado y otros pedidos podrían ser procesados.**

Podría suceder que durante el tiempo que transcurre entre que ARP envía el mensaje a IP comunicándole que el envío puede realizarse y que el protocolo IP manda el ip::Datagram para ser enviado, el *timeout* de la *MAC address* requerida se cumpla y la misma sea eliminada de la tabla. Si este es el caso, se envía un mensaje al protocolo IP de falla en el envío del ip::Datagram y se vuelve a mandar un link::arp::Packet para volver a obtener el *MAC address*. La figura 12 muestra la comunicación entre ARP y IP para la obtención de la *MAC address* y la Figura 13 muestra la estructura de un link::arp::Packet.

No esta explicado. Es un modelo atomico o acoplado?

Es una capa, un protocolo..

Un dispositivo puede tener varias interfaces y **cada interfaz es un modelo *Link* que implementa ARP y SWL**. Este es el caso de los routers que cuentan con una interfaz por cada network al que están conectados. Cada interfaz cuanta con un IP y con una *MAC address* y los pedidos de *MAC address* que llegan a la interfaz son respondidos por el protocolo ARP dentro de la interfaz sin llegar a la capa de enrutamiento. De esta forma, un router puede ser accedido por los dispositivos de una red solo si ellos conocen el IP de la interfaz que está conectada a dicha red, en caso de que llegue un pedido de *MAC address* para un IP de otra interfaz, el mismo es ignorado ya que las interfaces son independientes entre si.

Tanto *ARP* como *SWP* (explicado más adelante) son protocolos de nivel 2 y es por eso que se encuentran ambos implementados dentro del modelo *link*. La tabla 6 muestra los parámetros del modelo de *link* en la IDE PowerDEVS.

¹⁰En este trabajo no está implementado el mecanismo de fragmentación

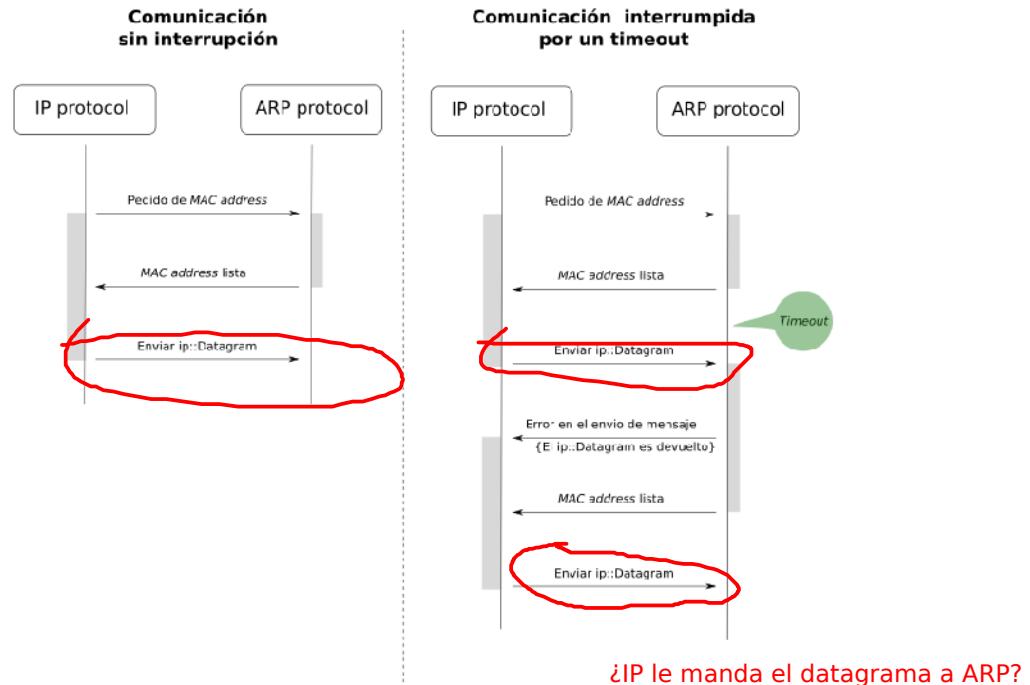


Figura 12: Flujo de comunicación entre los protocolos IP y ARP para enviar un ip::Datagram.

Hardware Type		Protocol Type
Hardware length	Protocol length	Operation
Sender hardware address		
Sender protocol address		
Target hardware address		
Target protocol address		

Figura 13: Estructura de datos de un link::arp::Packet alineada a 4 bytes. **Nota:** las direcciones de hardware (*MAC addresses*) son de 6 bytes y no de 4, esto no se muestra en la figura para mantener el orden de los campos.

[Referencia al RFC de ARP](#)

8.5. Datagrams

El [protocolo Datagrams](#) es utilizado por los switches para forwardear mensajes.

En el modelo implementado, el envío de frames sucede mediante cables ethernet, y para evitar colisiones, un dispositivo puede estar conectado a muchos otros mediante el uso de switches. Aunque en la realidad los switches evitan gran parte de las colisiones, los mismos no evitan todas, ya que siguen existiendo las colisiones generadas entre los frames que se envían de cada punta del cable. Este tipo de colisiones no están consideradas y el modelo implementado se abstrae de ellas, dejándolo como posible trabajo futuro.

El trabajo principal de un switch es el de forwardear frames que llegan por un puerto de entrada a través de un puerto de salida de forma de que el frame llegue al destinatario. Para realizar

Nombre	Tipo	Descripción
Module name	String	El nombre del módulo que utiliza el logger
Mac	MAC	la mac address de la interfaz
ip	IPv4	La ip de la interfaz
interface	Int	El número de la interfaz

Tabla 6: Parametros del modelo Link

esta tarea, el protocolo implementado es *Datagrams* explicado en el libro de cabecera de la cátedra [3, p. 172]. El mismo tiene un funcionamiento muy sencillo, utiliza una *forwarding table* que contiene las siguientes columnas:

- **mac:** La *MAC address* del destinatario.
- **interface:** La interfaz o puerto por donde debe forwardear el frame para llegar al destinatario.

Cada vez que un nuevo frame llega a un switch, el mismo busca la *MAC address* del destinatario en la *forwarding table* y envía el frame por el puerto correspondiente a esa *MAC address*. Si dicha *MAC address* no está en la *forwarding table*, el frame es descartado y se logea un mensaje de error.

Nuevamente, en este trabajo **no se implementó ningún algoritmo dinámico para completar las *forwarding tables* de los switches** y las mismas deben ser completadas a mano, si dichas tablas no están completas o están mal completadas los frames podrían nunca llegar a destino y/o quedarse dando vueltas entre los distintos switches infinitamente.

Tanto *Datagrams* como *SWP* (explicado más adelante) son protocolos de nivel 2 y es por eso que se encuentran ambos implementados dentro del modelo *switch*. La Tabla 7 muestra los parámetros del modelo *switch*.

Nombre	Tipo	Descripción
Module name	String	El nombre del módulo que utiliza el logger
Interface_amount	Int	La cantidad de interfaces por las cuales forwardear los Frames entrantes
forwarding_table	String	Path al archivo que contiene las entradas de la fowarding table

Tabla 7: Parámetros del modelo Switch

Sliding Window Protocol (SWP)

8.6. SWP

Este protocolo es el utilizado en la capa 2 y el mismo se encuentra tanto en el modelo *link* como en el modelo *switch*, de esta forma la comunicación de todos los dispositivos a nivel 2 es realizada mediante *SWP*.

En este trabajo también se implementó el protocolo *Sliding Window Protocol* de forma de garantizar que la comunicación entre dos dispositivos de la red se realiza sin perdida de frames y que los mismos llegan en orden, o sea, si se envía más de un paquete de un dispositivo a otro, los mismos son entregados en el mismo orden en que fueron enviados. Para no desaprovechar el ancho de banda enviando un solo frame por vez, el protocolo SWP utiliza una ventana que sirve para marcar cuantos frames pueden ser enviados en simultaneo de forma desordenada y a medida que los mismos van llegando, cuando el destinatario los ordena, la ventana se va corriendo. La Figura 14 ilustra la idea general del envío y recivo de mensajes mediante el protocolo SWP en una linea temporal.

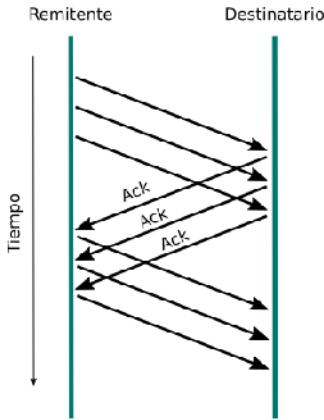


Figura 14: Idea general de la linea temporal del protocolo SWP.

El protocolo SWP está separado en dos partes, el remitente y el destinatario y funcionan de la siguiente manera:

8.6.1. Remitente:

Cuenta con una variable **SWS** (*Send Window Size*) que indica el tamaño de la ventana y una variable **LAR** (*Last Acknowledgment Received*) que indica el último frame enviado para el cual se recibió un acknowledgment. A medida que llegan frames a enviar, se les asigna un número de secuencia (*SeqNum*) correspondiente al orden de llegada y se procede a enviar todos aquellos que cumplen que $LAR < SeqNum \leq LAR + SWS$. Por otro lado, cada vez que recibe un ack, mira el *SeqNum* asociado al mismo y corre el inicio de la ventana (la variable **LAR**) hasta el mismo, esto automáticamente corre el final de la ventana y habilita por lo tanto nuevos frames a ser enviados. Este mecanismo funciona porque se sabe que el destinatario solo envía el ack de un frame cuando todos los frames anteriores al mismo también fueron correctamente recibidos. Cada vez que un frame es enviado, el mismo es guardado en un arreglo de frames enviados esperando que llegue un ack igual o más grande que el mismo, cuando el ack llega, el frame es descartado. Además, todo frame que es enviado se le asigna un timeout, si no se recibe un ack válido para dicho frame antes de que se termine el timeout, se considera que hubo una falla en el envío y el frame es enviado nuevamente.

8.6.2. Destinatario:

Cuenta con una variable **RWS** (*Receive Window Size*) que indica el tamaño de la ventana de cuáles *SeqNum* son aceptados y con una variable **LFR** (*Last Frame received*) que indica el último frame para el cual envió un acknowledgment y por ende el comienzo de la ventana. En caso de que un frame llegue y no cumpla que $LFR < SeqNum \leq LFR + RWS$, el mismo es descartado. Los frames que llegan y son aceptados se guardan en un arreglo a la espera de ser entregados a la capa superior. Cada vez que un frame es aceptado, se mira cual es el frame aceptado con *SeqNum* más grande tal que todos los frames con *SeqNum* menor a el ya fueron recibidos y se envía un ack de dicho frame. Una vez que se envió el ack de un frame, el y todos los frames anteriores son enviados a la capa superior. int? double?

Una cosa a tener en cuenta es que el *SeqNum* está representado con un unsigned char y cuando llega al máximo número representable la secuencia vuelve a empezar de cero. Para evitar problemas con esto, es necesario asegurarse que las ventanas **SWS** y **RWS** sean lo suficientemente chicas respecto al máximo *SeqNum* de forma de que cuando se llegue al mismo y se vuelva a empezar no se generen confusiones.

8.7. Tests de unidad

para tener una validación del comportamiento de cada uno de los módulos del modelo implementado, se realizaron distintos test de unidad, en los cuales se inventó el input de los modelos y se comparó el resultado obtenido con el esperado. De esta forma a continuación se listan todos los test unitarios implementados:

El único módulo que no cuenta con test de unidad, es el modelo *Switch* ya que el mismo se encuentra en un modelo acoplado de dispositivo donde es el único módulo, con lo cuál, el test de integración para este caso funciona como test de unidad también.

Para realizar test de unidad de cada módulo por separado (un módulo representa una capa o un protocolo en específico), dado que los mismos funcionan en su totalidad mediante la interacción con el resto de los modelos y ellos no van a estar en el test (por ser de unidad y no de integración), se implementaron múltiples modelos generadores y vertederos utilizando la técnica descripta en la Sección 6. De esta forma, por cada puerto de entrada por el cual el modelo recibe un input o emite un output desde y hacia otros modelos, se mockean los inputs y se guardan en archivos los output generados por el modelo consiguiendo de esta forma emular un simulación con interacciones aunque no estén todas las partes presentes.

Nota: En todos los escenarios realizados para los test de unidad, los paquetes utilizados, son paquetes vacíos.

8.7.1. Test de unidad para el módulo DNS

En el caso del modelo DNS, el test de unidad básico implementado recibe un pedido de resolución para el dominio “networks.devs.com” por el puerto de entrada 0 (puerto por el cual recibe datos de la capa superior) en el tiempo virtual 15, el módulo envía un pedido recursivo a su servidor local con ip conocido 1.0.0.2 por el puerto de salida 3 y luego de que el servidor resuelve la consulta de forma recursiva, se le envía un mensaje con la respuesta en formato de paquete dns por el puerto de entrada 2 (puerto por el cual recibe datos de la capa inferior), el mismo procesa la respuesta y la reenvía por el puerto de salida 0. La Figura 15 muestra el modelo implementado en PowerDEVS para el test de unidad del modelo dns con sus respectivas fuentes y vertederos.

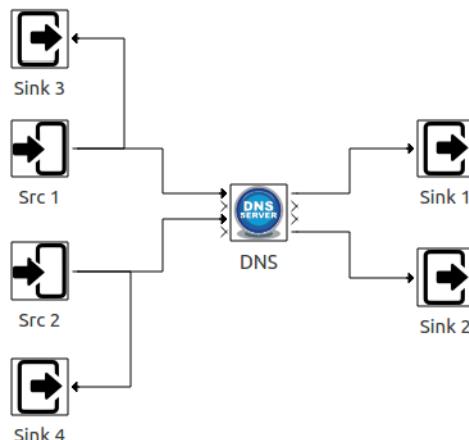


Figura 15: Src 1 genera input de tipo dns::DomainName, src 2 genera input de tipo dns::Packet. Sink 1 imprime output de tipo dns::Packet y Sink 2 imprime output de tipo udp::Control, Sinks 3 y 4 imprimen el input generado por Src 1 y Src 2.

A continuación se muestra el input y output del test:

Input port 0 (dns::DomainName)

```
time:15
network.devs.com
```

Input port 2 (dns::Packet)

```
time:18
----- dns packet -----
** HEADER **
id: 0
flags: QR_ANSWER
      Opcode_QUERY
      AA_NOT_AUTHORITATIVE_ANSWER
      TC_NOT_TRUNCATED
      RD_ITERATIVE
      RA_ITERATIVE
      RCode_NO_ERROR
QDCount: 1
ANCount: 1
NSCount: 0
ARCount: 1

** QUESTION SECTION **
network.devs.com A 0.0.0.0 IN 0

** ANSWER SECTION **
network.devs.com A 1.0.0.3 IN 65

** AUTHORITATIVES SECTION **

** ADDITIONAL SECTION **
network.devs.com A 1.0.0.1 IN 0
```

Output port 0 (dns::Packet)

```
time:18.002
----- dns packet -----
** HEADER **
id: 0
flags: QR_ANSWER
      Opcode_QUERY
      AA_NOT_AUTHORITATIVE_ANSWER
      TC_NOT_TRUNCATED
      RD_ITERATIVE
      RA_ITERATIVE
      RCode_NO_ERROR
QDCount: 1
ANCount: 1
NSCount: 0
ARCount: 1

** QUESTION SECTION **
network.devs.com A 0.0.0.0 IN 0

** ANSWER SECTION **
network.devs.com A 1.0.0.3 IN 65

** AUTHORITATIVES SECTION **

** ADDITIONAL SECTION **
network.devs.com A 1.0.0.1 IN 0
```

Output port 3 (udp::Control)

```
time:0.001
app_id: 0
Ctrol: BIND
local_port: 53
local_ip: 1.0.0.1

time:0.002
app_id: 0
Ctrol: READ
local_port: 53
local_ip: 1.0.0.1

time:15.002
app_id: 0
Ctrol: SEND_TO
local_port: 53
local_ip: 1.0.0.1
remote_port: 53
remote_ip: 1.0.0.2
packet:
----- dns packet -----
** HEADER **
id: 0
flags: QR.QUERY
        Opcode.QUERY
        AA.NOT.AUTHORITATIVE_ANSWER
        TC.NOT.TRUNCATED
        RD.RECURSIVE
        RA.ITERATIVE
        RCode.NO_ERROR
QDCount: 1
ANCount: 0
NSCount: 0
ARCount: 1

** QUESTION SECTION **
network.devs.com A 0.0.0.0 IN 0

** ANSWER SECTION **

** AUTHORITATIVES SECTION **

** ADDITIONAL SECTION **
network.devs.com A 1.0.0.1 IN 44242
----- dns packet -----
** HEADER **
id: 0
Ctrol: READ
local_port: 53
local_ip: 1.0.0.1
```

Output generado por el logger durante la simulación

```
[INFO] - [DNS Test] DNS local ip: 1.0.0.1
[INFO] - [DNS Test] DNS root server ip: 1.0.0.2
[INFO] - [DNS Test] No authoritative RR table file.
[INFO] - [DNS Test] No zone server table file.
[INFO] - [DNS Test] Recursive allowed: True
Simulation Initialized
[INFO] - [DNS Test] Process domain name network.devs.com
[INFO] - [DNS Test] Domain not in cache
[INFO] - [DNS Test] Send DNS Query to: (1.0.0.2, 53)
[INFO] - [DNS Test] Incoming packet:
[INFO] - [DNS Test] dns packet -----
** HEADER **
id: 0
flags_code: 0
QDCount: 1
ANCount: 1
NSCount: 0
ARCount: 1

** QUESTION SECTION **
network.devs.com A 0.0.0.0 IN 0

** ANSWER SECTION **
network.devs.com A 1.0.0.3 IN 65

** AUTHORITATIVES SECTION **

** ADDITIONAL SECTION **
network.devs.com A 1.0.0.1 IN 0
-----
[INFO] - [DNS Test] Answer packet
[INFO] - [DNS Test] Answer Type: A
Simulation Ended (0.000862 sec)
```

8.7.2. Test de unidad para el módulo UDP

Para testear el módulo UDP se utilizó la misma lógica que para DNS. En este caso, se implementaron generadores tanto para los controles de udp que envían pedidos de conexión, bindeo, etc. desde la capa superior, como para los udp::Segment que llegan desde la capa inferior. La figura 16 muestra el modelo implementado en PowerDEVS para el test de unidad del módulo UDP.

En este escenario, se simuló lo siguiente:

1. El módulo de capa de aplicación se conecta la aplicación con id 0 al socket local 1.0.0.1:80 y remoto 1.0.0.2:80.
2. El módulo de capa de aplicación realiza envía dos paquetes vacíos para la aplicación con id 0 utilizando el socket previamente conectado.
3. El módulo de capa de aplicación bindea la aplicación con id 1 al socket local 1.0.0.1:8080.
4. El módulo de capa de aplicación envía un paquete para la aplicación con id 1 especificando el destinatario con socket remoto 1.0.0.2:8080.
5. El módulo de capa de aplicación envía un paquete para la aplicación con id 0 por el socket local 1.0.0.1:53 inexistente, por lo que este pedido falla.
6. Una aplicación remota envía un paquete con socket remoto 1.0.0.2:80 y local 1.0.0.1:80. Si bien el socket es válido, el udp::Segment entrante falla en el checksum ya que el mismo es inválido y es descartado (esto no se ve en el output, pero se ve en el log de la simulación).

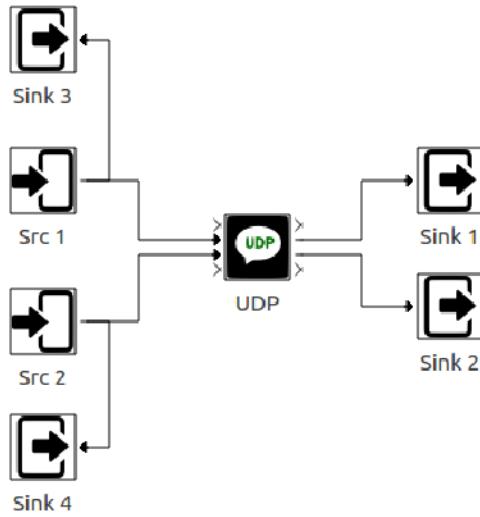


Figura 16: Src 1 genera input de tipo udp::control, src 2 genera input de tipo udp::Segment. Sink 1 imprime output de tipo udp::Control, Sinks 2 imprime output de tipo udp::Segment mientras que Sink 3 y 4 imprimen el input generado por Src 1 y Src 2.

A continuación se muestra el input, output y log del test:

Input port 1 (udp::Control)

```

time:15
app_id: 0
Ctrol: CONNECT
local_port: 80
local_ip: 1.0.0.1
remote_port: 80
remote_ip: 1.0.0.2

time:16
app_id: 0
Ctrol: SEND
local_port: 80
local_ip: 1.0.0.1
packet:
----- dns packet -----
** HEADER **
id: 0
flags: QRANSWER
Opcode_QUERY
AA_NOT_AUTHORITATIVE_ANSWER
TCNOTTRUNCATED
RD_ITERATIVE
RA_ITERATIVE
RCode_NO_ERROR
QDCount: 0
ANCount: 0
NSCount: 0
ARCount: 0

** QUESTION SECTION **

** ANSWER SECTION **

```

```
** AUTHORITATIVES SECTION **
```

```
** ADDITIONAL SECTION **
```

```
time:17
app_id: 0
Ctrol: SEND
local_port: 80
local_ip: 1.0.0.1
packet:
----- dns packet -----
** HEADER **
id: 0
flags: QR_ANSWER
    Opcode_QUERY
    AA_NOT_AUTHORITATIVE_ANSWER
    TC_NOT_TRUNCATED
    RD_ITERATIVE
    RA_ITERATIVE
    RCode_NO_ERROR
QDCount: 0
ANCount: 0
NSCount: 0
ARCount: 0

** QUESTION SECTION **

** ANSWER SECTION **

** AUTHORITATIVES SECTION **

** ADDITIONAL SECTION **
```

```
time:18
app_id: 1
Ctrol: BIND
local_port: 8080
local_ip: 1.0.0.1
```

```
time:19
app_id: 1
Ctrol: SEND_TO
local_port: 8080
local_ip: 1.0.0.1
remote_port: 8080
remote_ip: 1.0.0.2
packet:
----- dns packet -----
** HEADER **
id: 0
flags: QR_ANSWER
    Opcode_QUERY
    AA_NOT_AUTHORITATIVE_ANSWER
    TC_NOT_TRUNCATED
    RD_ITERATIVE
    RA_ITERATIVE
    RCode_NO_ERROR
QDCount: 0
ANCount: 0
NSCount: 0
```

```
ARCount: 0
** QUESTION SECTION **
** ANSWER SECTION **
** AUTHORITATIVES SECTION **
** ADDITIONAL SECTION **
```

```
time:20
app_id: 0
Ctrol: SEND
local_port: 53
local_ip: 1.0.0.1
packet:
----- dns packet -----
** HEADER **
id: 0
flags: QR_ANSWER
    Opcode_QUERY
    AA_NOT_AUTHORITATIVE_ANSWER
    TC_NOT_TRUNCATED
    RD_ITERATIVE
    RA_ITERATIVE
    RCode_NOERROR
QDCount: 0
ANCount: 0
NSCount: 0
ARCount: 0

** QUESTION SECTION **
** ANSWER SECTION **
** AUTHORITATIVES SECTION **
** ADDITIONAL SECTION **
```

Input port 2 (udp::Segment)

```
time:18
psd_header:
src_ip: 1.0.0.2
dest_ip: 1.0.0.1
zeros: 0
protocol: 0
udp_length: 0

header:
src_port: 80
dest_port: 80
length: 0
checksum: 0

payload:
----- dns packet -----
** HEADER **
id: 0
flags: QR_ANSWER
    Opcode_QUERY
    AA_NOT_AUTHORITATIVE_ANSWER
    TC_NOT_TRUNCATED
    RD_ITERATIVE
```

```

        RA_ITERATIVE
        RCode_NO_ERROR
QDCount: 0
ANCount: 0
NSCount: 0
ARCount: 0

** QUESTION SECTION **

** ANSWER SECTION **

** AUTHORITATIVES SECTION **

** ADDITIONAL SECTION **

```

Output port 1 (udp::Control)

```

time:15.011
app_id: 0
Ctrol: SUCCESS

```

```

time:18.011
app_id: 1
Ctrol: SUCCESS

```

```

time:20.011
app_id: 0
Ctrol: INVALID_SOCKET

```

Output port 2 (udp::Segment)

```

time:16.011
psd_header:
src_ip: 1.0.0.1
dest_ip: 1.0.0.2
zeros: 0
protocol: 0
udp_length: 20

```

```

header:
src_port: 80
dest_port: 80
length: 12
checksum: 41297

```

```

payload:
----- dns packet -----
** HEADER **
id: 0
flags: QR_ANSWER
        Opcode_QUERY
        AA_NOT_AUTHORITATIVE_ANSWER
        TC_NOT_TRUNCATED
        RD_ITERATIVE
        RA_ITERATIVE
        RCode_NO_ERROR

```

```

QDCount: 0
ANCount: 0
NSCount: 0
ARCount: 0

```

```

** QUESTION SECTION **

** ANSWER SECTION **

** AUTHORITATIVES SECTION **

```

```
** ADDITIONAL SECTION **
```

```
time:17.011
psd_header:
src_ip: 1.0.0.1
dest_ip: 1.0.0.2
zeros: 0
protocol: 0
udp_length: 20

header:
src_port: 80
dest_port: 80
length: 12
checksum: 41297

payload:
----- dns packet -----
** HEADER **
id: 0
flags: QR_ANSWER
    Opcode_QUERY
    AA_NOT_AUTHORITATIVE_ANSWER
    TC_NOT_TRUNCATED
    RD_ITERATIVE
    RA_ITERATIVE
    RCode_NO_ERROR
QDCount: 0
ANCount: 0
NSCount: 0
ARCount: 0

** QUESTION SECTION **

** ANSWER SECTION **

** AUTHORITATIVES SECTION **

** ADDITIONAL SECTION **
```

```
time:19.011
psd_header:
src_ip: 1.0.0.1
dest_ip: 1.0.0.2
zeros: 0
protocol: 0
udp_length: 20

header:
src_port: 8080
dest_port: 8080
length: 12
checksum: 8851

payload:
----- dns packet -----
** HEADER **
id: 0
flags: QR_ANSWER
    Opcode_QUERY
```

```

AA_NOT_AUTHORITATIVE_ANSWER
TC_NOT_TRUNCATED
RD_ITERATIVE
RA_ITERATIVE
RCode_NO_ERROR

QDCount: 0
ANCount: 0
NSCount: 0
ARCount: 0

** QUESTION SECTION **

** ANSWER SECTION **

** AUTHORITATIVES SECTION **

** ADDITIONAL SECTION **

```

Output generado por el logger durante la simulación

```

Simulation Initialized
[INFO] - [UDP Test] Connect
[INFO] - [UDP Test] SUCCESS
[INFO] - [UDP Test] Write/Send
[INFO] - [UDP Test] Write/Send
[INFO] - [UDP Test] Bind
[INFO] - [UDP Test] SUCCESS
[INFO] - [UDP Test] process Segment:
[INFO] - [UDP Test] local_port: 80
[INFO] - [UDP Test] remote_port: 80
[INFO] - [UDP Test] local_ip: 1.0.0.1
[INFO] - [UDP Test] remote_ip: 1.0.0.2
[INFO] - [UDP Test] valid checksum: False
[INFO] - [UDP Test] discarded segment.
[INFO] - [UDP Test] Write_to/Send_to
[INFO] - [UDP Test] Write/Send
[INFO] - [UDP Test] INVALID_SOCKET
Simulation Ended (0.00135 sec)

```

8.7.3. Test de unidad para el módulo IP Host

En este caso, el test implementado es para testear el comportamiento del protocolo IP para los hosts, la única diferencia entre este protocolo IP y el protocolo IP de los routers es que en este caso, se trabaja con la capa superior (UDP) y es necesario encapsular los udp::Segments en ip::Datagrams para forwardearlos por la red. Para este sencillo test se simula el proceso recién mencionado. La Figura 17 muestra es escenario para este test de unidad.

A continuación se muestra el input, output y log del test:

Input port 0 (udp::Segment)

```

time:15
psd_header:
src_ip: 1.0.0.1
dest_ip: 1.0.0.2
zeros: 0
protocol: 0
udp_lenght: 0

header:
src_port: 80
dest_port: 80
length: 0
checksum: 0

payload:

```

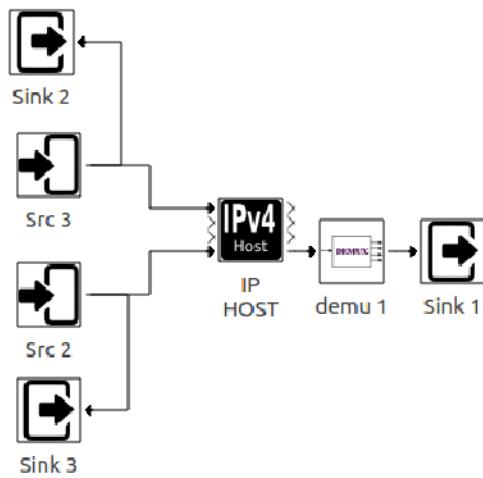


Figura 17: Src 1 genera input de tipo udp::Segment, src 2 genera input de tipo link::Control. Sink 1 imprime output de tipo link::Control mientras que Sink 2 y 3 imprimen el input generado por Src 1 y Src 2.

```

----- dns packet -----
** HEADER **
id : 0
flags : QR_ANSWER|Opcode_QUERY|AA_NOT_AUTHORITATIVE_ANSWER|TC_NOT_TRUNCATED|RD_ITERATIVE|RA_ITERATIVE
QDCount: 0
ANCount: 0
NSCount: 0
ARCount: 0

** QUESTION SECTION **

** ANSWER SECTION **

** AUTHORITATIVES SECTION **

** ADDITIONAL SECTION **
-----
```

Input port 3 (link::Control)

```
time:17
request: ARP_READY
interface: 0
ip: 1.0.0.2
```

Output port 3 (link::Control)

```
time:15.002
request: ARP_QUERY
interface: 0
ip: 1.0.0.2

time:17.002
request: SEND PACKET
interface: 0
ip: 1.0.0.2
Datagram:
Header:
vide: 17664
total_length: 28
identification: 0
ff: 0
ttlp: 65297
header_checksum: 35980
src_ip: 1.0.0.1
dest_ip: 1.0.0.2
```

```
data:
psd_header:
src_ip: 1.0.0.1
dest_ip: 1.0.0.2
zeros: 0
protocol: 0
udp_length: 0
```

```
header:
src_port: 80
dest_port: 80
length: 0
checksum: 0
```

payload:

```
----- dns packet -----
```

```
** HEADER **
```

```
id: 0
```

```
flags: QR ANSWER|Opcode QUERY|AA NOT AUTHORITATIVE ANSWER|TC NOT TRUNCATED|RD ITERATIVE|RA ITERATIVE
```

```
QDCount: 0
```

```
ANCount: 0
```

```
NSCount: 0
```

```
ARCount: 0
```

```
** QUESTION SECTION **
```

```
** ANSWER SECTION **
```

```
** AUTHORITATIVES SECTION **
```

```
** ADDITIONAL SECTION **
```

Output generado por el logger durante la simulación

```

[INFO] - [IPv4 Test] ip amount: 1
[INFO] - [IPv4 Test] ips:
[INFO] - [IPv4 Test] 1.0.0.1
[INFO] - [IPv4 Test] Routing table:
[INFO] - [IPv4 Test] | 0.0.0.0 | 0.0.0.0 | 1.0.0.0 | 1 | Default Route |
[INFO] - [IPv4 Test] | 1.0.0.2 | 255.255.255.255 | 1.0.0.2 | 1 | Host 2 |
[INFO] - [IPv4 Test] Forwarding table:
[INFO] - [IPv4 Test] | 0.0.0.0 | 0.0.0.0 | 0
[INFO] - [Demultiplexer Host 1] Initialized with 1 interfaces
Simulation Initialized
[INFO] - [IPv4 Test] Forwarding ip packet: 1.0.0.2
[INFO] - [IPv4 Test] Best route for packet with dest_ip: 1.0.0.2
is: | 1.0.0.2 | 255.255.255.255 | 1.0.0.2 | 1 | Host 2 |
[INFO] - [IPv4 Test] Added packet to wait ARP for nexthop: 1.0.0.2
[INFO] - [IPv4 Test] ARP query throw interface: NET 0
[INFO] - [IPv4 Test] link ::Ctrl::ARP_READY
Simulation Ended (0.000886 sec)

```

8.7.4. Test de unidad para el módulo IP Router

Para el caso del protocolo IP en routers se implementó un escenario donde a diferencia del anterior para IP en hosts, en este caso llega un ip::Datagram proveniente de la red y el mismo es forwardeado ya que un router no tiene capa 4 y todo ip::Datagram que llega es forwardeado para llegar a su destino o descartado si no pasa las verificaciones como checksum o su campo *TTL* es 0. La Figura 18 muestra el escenario para este test.

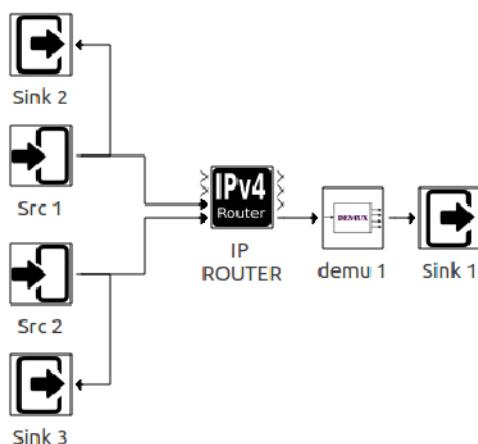


Figura 18: Src 1 genera input de tipo ip::Datagram, src 2 genera input de tipo link::Control. Sink 1 imprime output de tipo link::Control mientras que Sink 2 y 3 imprimen el input generado por Src 1 y Src 2.

A continuación se muestra el input, output y log del test:

Input port 2 (ip::Datagram)

```
time:15
Header:
vide: 0
total_length: 0
identification: 0
ff: 0
ttl: 65535
header_checksum: 65278
src_ip: 1.0.0.1
dest_ip: 1.0.0.2

data:
psd_header:
src_ip: 1.0.0.1
dest_ip: 1.0.0.2
zeros: 0
protocol: 0
udp_length: 0

header:
src_port: 80
dest_port: 80
length: 0
checksum: 0

payload:
----- dns packet -----
** HEADER **
id: 0
flags: QR_ANSWER|Opcode_QUERY|AA_NOT_AUTHORITATIVE_ANSWER|TC_NOT_TRUNCATED|RD_ITERATIVE|RA_ITERATIVE
QDCount: 0
ANCount: 0
NSCount: 0
ARCount: 0

** QUESTION SECTION **

** ANSWER SECTION **

** AUTHORITATIVES SECTION **

** ADDITIONAL SECTION **
```

Input port 3 (link::Control)

```
time:19
request: ARP_READY
interface: 0
ip: 1.0.0.2
```

Output port 3 (link::Control)

```
time:15.002
request: ARP_QUERY
interface: 0
ip: 1.0.0.2

time:19.002
request: SEND_PACKET
interface: 0
ip: 1.0.0.2
Datagram:
Header:
```

```

vide: 0
total_length: 0
identification: 0
ff: 0
ttl: 65279
header_checksum: 0
src_ip: 1.0.0.1
dest_ip: 1.0.0.2

data:
psd_header:
src_ip: 1.0.0.1
dest_ip: 1.0.0.2
zeros: 0
protocol: 0
udp_length: 0

header:
src_port: 80
dest_port: 80
length: 0
checksum: 0

payload:
----- dns packet -----
** HEADER **
id: 0
flags: QR_ANSWER|Opcode_QUERY|AA_NOT_AUTHORITATIVE_ANSWER|TC_NOT_TRUNCATED|RD_ITERATIVE|RA_ITERATIVE
QDCount: 0
ANCount: 0
NSCount: 0
ARCount: 0

** QUESTION SECTION **

** ANSWER SECTION **

** AUTHORITATIVES SECTION **

** ADITIONAL SECTION **
-----
```

Output generado por el logger durante la simulación

```

[INFO] - [IPv4 Test] ip amount: 1
[INFO] - [IPv4 Test] ips:
[INFO] - [IPv4 Test] 1.0.0.1
[INFO] - [IPv4 Test] Routing table:
[INFO] - [IPv4 Test] | 0.0.0.0 | 0.0.0.0 | 1.0.0.0 | 1 | Default Route |
[INFO] - [IPv4 Test] | 1.0.0.2 | 255.255.255.255 | 1.0.0.2 | 1 | Host 2 |
[INFO] - [IPv4 Test] Forwarding table:
[INFO] - [IPv4 Test] | 0.0.0.0 | 0.0.0.0 | 0 |
[INFO] - [Demultiplexer Host 1] Initialized with 1 interfaces
Simulation Initialized
[INFO] - [IPv4 Test] Process Ip Datagram
[INFO] - [IPv4 Test] Valid checksum: true
[INFO] - [IPv4 Test] Forwarding ip packet: 1.0.0.2
[INFO] - [IPv4 Test] Best route for packet with dest_ip
is: 1.0.0.2 is: | 1.0.0.2 | 255.255.255.255 | 1.0.0.2 | 1 | Host 2 |
[INFO] - [IPv4 Test] Added packet to wait ARP for nexthop: 1.0.0.2
[INFO] - [IPv4 Test] ARP query throw interface: NET 0
[INFO] - [IPv4 Test] link::Ctrl::ARPREADY
Simulation Ended (0.000795 sec)
```

8.7.5. Test de unidad para el módulo link

Por último, el test de unidad para el módulo link, el cual implementa *ARP* y *SWP*, fue testeado con un escenario en el cual le llega un `link::Frame` de un nodo de la red que es aceptado. El frame entrante, es un frame de datos que contiene un `ip::Datagram` que es desencapsulado y entregado a la capa superior. Luego de haber entregado el `ip::Datagram`, el protocolo *SWP* se encarga de enviar un *Ack* al nodo que envió el frame para notificarle que llegó correctamente y que este pueda actualizar su ventana *SWS* y seguir enviando más frames en un futuro. La Figura 19 Muestra el escenario en la IDE PowerDEVS para este test.

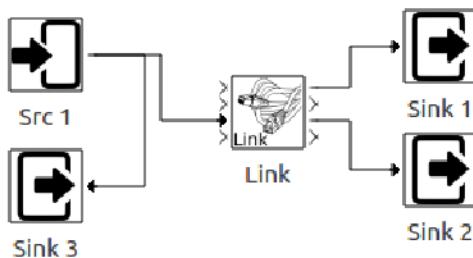


Figura 19: Src 1 genera input de tipo `link::Frame`, Sink 1 imprime output de tipo `ip::Datagram`, Sink 2 imprime output de tipo `link::Frame` mientras que Sink 3 imprime el input generado por Src 1.

A continuación se muestra el input, output y log del test:

Input port 2 (link::Frame)

```

time:15
preamble: 10200000
MAC_destination: 0:0:0:0:0:1
MAC_source: 0:0:0:0:0:2
EtherType: 0
CRC: 0
  
```

Input port 0 (ip::Datagram)

```

time:15.005
Header:
vide: 11840
total_length: 24539
identification: 32766
ff: 0
ttlp: 9
header_checksum: 0
src_ip: 0.0.0.0
dest_ip: 0.0.0.0

data:
psd_header:
src_ip: 0.0.0.0
dest_ip: 0.0.0.0
zeros: 0
protocol: 0
udp_length: 0

header:
src_port: 0
dest_port: 0
length: 0
checksum: 0

payload:
  
```

```

----- dns packet -----
** HEADER **
id : 0
flags : QR_ANSWER|Opcode_QUERY|AA_NOT_AUTHORITATIVE_ANSWER|TC_NOT_TRUNCATED|RD_ITERATIVE|RA_ITERATIVE
QDCount: 0
ANCount: 0
NSCount: 0
ARCount: 0

** QUESTION SECTION **

** ANSWER SECTION **

** AUTHORITATIVES SECTION **

** ADITIONAL SECTION **

```

Output port 2 (link::Frame)

```

time:15.003
preamble: 1100000
MAC_destination: 0:0:0:0:0:2
MAC_source: 0:0:0:0:0:1
EtherType: 0
CRC: 0

```

Output generado por el logger durante la simulación

```

[INFO] - [Link Test] MAC: 0:0:0:0:0:1
[INFO] - [Link Test] IP: 1.0.0.2
[INFO] - [Link Test] Interface: 0
Simulation Initialized
[INFO] - [SWP Test] swp_protocol::processFrame
[INFO] - [SWP Test] Deliver frame with SeqNum: 1
[INFO] - [SWP Test] send ACK LFR: 1
Simulation Ended (0.000283 sec)

```

8.8. Test de integración - Escenario implementado

Para finalizar y a modo de test de integración, se implementó un modelo de una red y se simuló un escenario donde todos los módulos funcionan en conjuntos para enviar mensajes entre los *host* de dicha red. El escenario implementado en este trabajo está pensado para cubrir de forma minimal los eventos relevantes en el envío de mensajes a través de la red tales como: Pasar por múltiples *routers*, enviar mensajes a otro host de la misma red y enviar mensajes a un *host* de una red donde es el único *host*. Dado que se implementó *DNS*, el escenario también busca mostrar un ejemplo donde se ejecuten tanto pedidos recursivos como pedidos iterativos a los distintos servidores *DNS*. Para conseguir esto, se planteó el escenario mostrado en la Figura 20, el mismo cuenta con cinco *subnets* con diferentes cantidades de *switches* y *hosts*, las *subnets* 2 y 3 no cuentan con dispositivos extras a los *routers* a los cuales están conectadas ya que no aportaría nada al test. Las Figuras 21, 22 y 23 muestran los modelos acoplados de dispositivos hosts routers y switches correspondientes al escenario planteado.

En este escenario, el host 2 es el servidor local para la *subnet* 1, el host 5 es el host TLD mientras que el host 4 es el host responsable de la zona “.com” y el host 3 es el host responsable para la zona “.devs.com” con lo cual, el es autoritativo para el dominio “networks.devs.com”. En esta simulación el host 1 le pide a su servidor local (host 2) que resuelva el dominio “networks.devs.com” de forma recursiva, el mismo al no tener esa entrada en su cache, envía el pedido al servidor TLD (host 5) quien no tiene la respuesta pero sabe que pertenece a la zona “.com” y le envía un mensaje diciéndole que el host 4 es el responsable de esa zona, también le envía el *IP* del host 4 para que sepa como llegar a el, el host 2 envía entonces la misma consulta al host 4 y repitiéndose lo mismo que antes, el host 4 al no tener la respuesta pero saber que el host 3 es responsable de la zona

más puntual “.devs.com”, le envía el *IP* del host 3 y repitiendo el proceso, el host 2 le envía el mismo pedido al host 3 quien si tiene la respuesta. Una vez con la respuesta, el host 2 la agrega a su cache, para que se mantenga ahí hasta que se venza su *TTL* y se la envía al host 1.

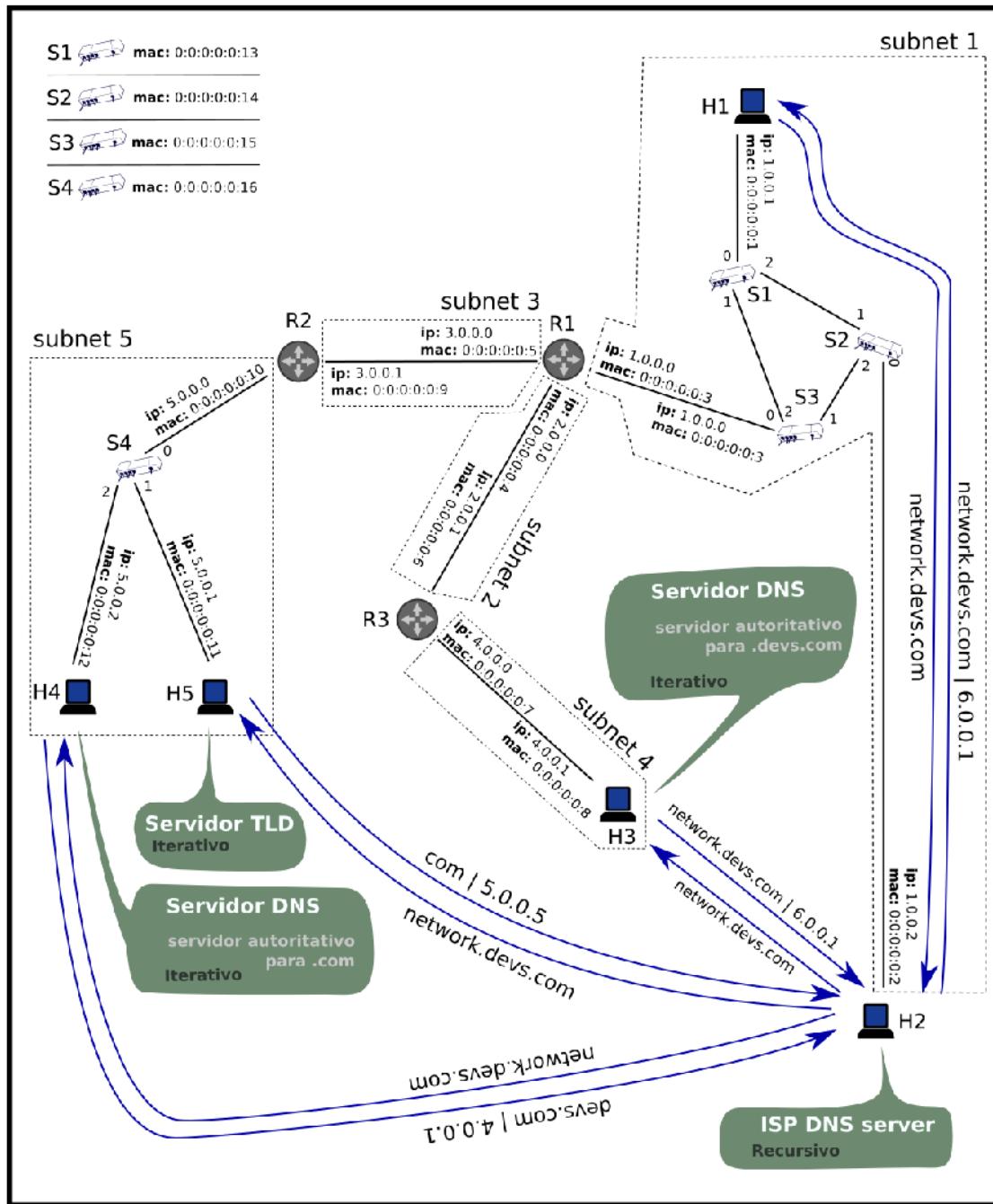


Figura 20: Escenario implementado: Las líneas azules muestran el intercambio de paquetes *DNS* entre los hosts mientras que las líneas negras muestran las conexiones de la red por donde efectivamente se envían los paquetes, teniendo que pasar por múltiples switches y routers para ir de un host a otro. Los números al lado de los switches muestran el mapeo de puertos para el protocolo *Datagram*.

Como se ve el modelos completo en PowerDEVS

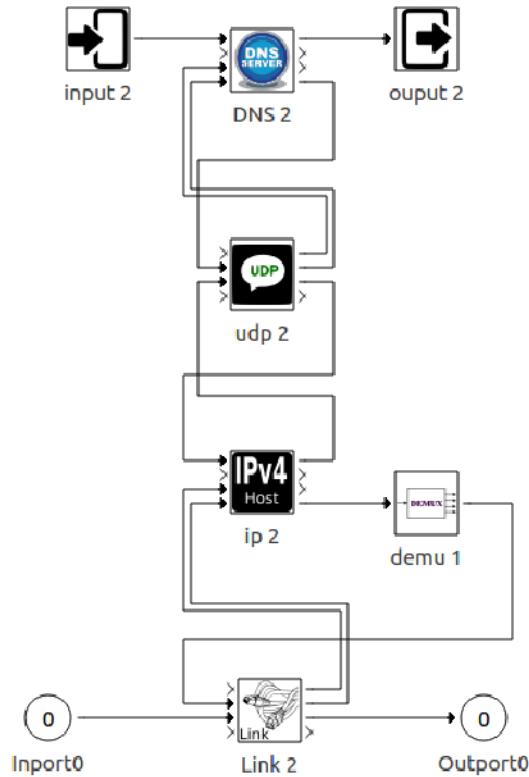
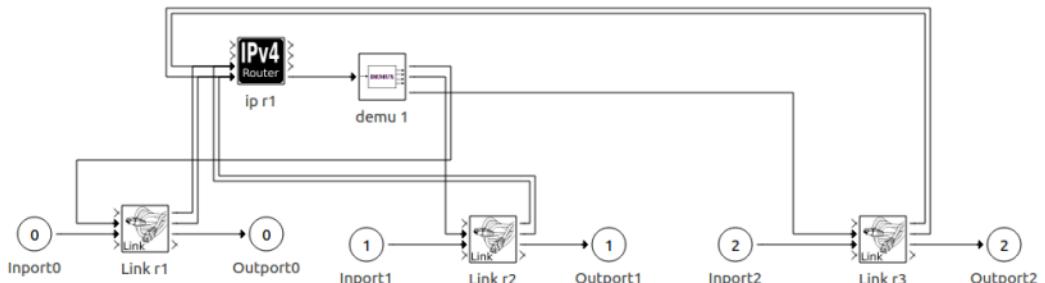


Figura 21: Modelo PowerDEVS de un dispositivo Host con las capas *DNS*, *UDP*, *IPv4* y *Link*; Se pueden ver las conexiones realizadas entre cada una de las capas del modelo UDP/IP así como el generador y el vertedero correspondientes al input y output de este modelo. Los puertos externos (Import0 y Outport0) permiten conectar el modelo dispositivo a otro dispositivo de la red para enviar los mensajes.



Estos modelos habria que agregarlos a una libreria para que sea facil reutilizarlos.

Figura 22: Modelo PowerDEVS de un dispositivo Router con las capas *IPv4* y *Link*; El router cuenta con un modelo link por cada *subnet* a la cual está conectado, para enviar un mensaje, el router debe elegir mediante la *forwarding table* por cual interfaz (modelo link) enviar el mensaje dependiendo a que *subnet* quiera alcanzar. Para poder enviar mensajes a los distintos módulos *Link* se utiliza un demultiplexador.

9. Resultados

A continuación se presentan los resultados del escenario con el mismo formato que se presentaron para los test de unidad. Primero se presenta el input y output de la simulación en el cual se detalla para cada tiempo virtual, el mensaje entrante o saliente del modelo. La documentación respecto al formato de los mensajes se puede ver en la misma documentación Doxygen del código.

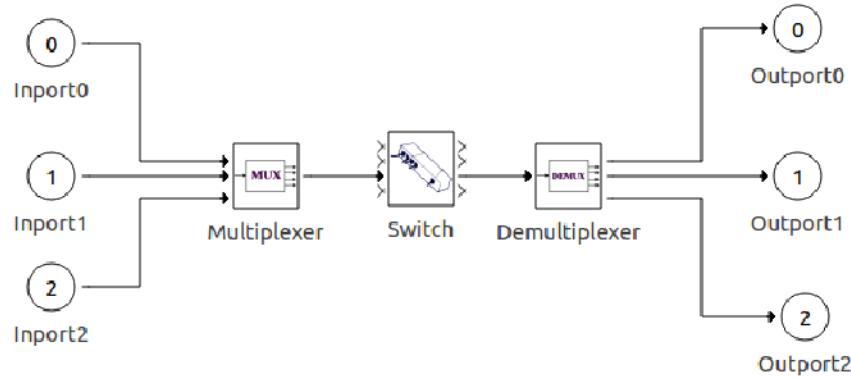


Figura 23: Modelo PowerDEVS de un dispositivo *Switch* que cuenta con una sola capa en la cual implementa el protocolo *Datagram* y *SWP*. Utiliza un multiplexor y demultiplexor para saber de cuál interfaz llegó el mensaje y para elegir por cual o cuales interfaces enviar el mensaje.

Luego se muestra el log generado por el logger, el mismo se muestra primero separado por capas (*link*, *ip*, *udp*, etc.) permitiendo ver el funcionamiento horizontal de cada capa, en donde se desconoce lo máximo posible la existencia de capas inferiores y los mensajes son enviados a la misma capa de otro nodo de la red, y luego se muestra el log completo para poder ver el funcionamiento entre las capas y la relación temporal de los eventos ocurridos en la simulación.

Host 1 - DNS - Input port 0 (dns::DomainName)

```
time:15
network.devs.com
```

Host 1 - DNS - Output port 0 (dns::Packet)

```
time:27915.7
----- dns packet -----
** HEADER **
id: 0
flags: QR_ANSWER|Opcode_QUERY|AA_AUTHORITATIVE_ANSWER|TC_NOT_TRUNCATED|RD_RECURSIVE|RA_ITERATIVE|RC
QDCount: 1
ANCount: 1
NSCount: 0
ARCount: 1

** QUESTION SECTION **
network.devs.com A 0.0.0.0 IN 0

** ANSWER SECTION **
network.devs.com A 6.0.0.1 IN 65535

** AUTHORITATIVES SECTION **

** ADDITIONAL SECTION **
network.devs.com A 1.0.0.2 IN 9227
```

Como se puede ver, primero llega el mensaje entrante con el nombre de dominio para el cual se quiere resolver su *IP* y luego, una vez resueltas todas las consultas recursivas e iterativas a los distintos servidores, se consigue la respuesta indicando que el *IP* vinculado al dominio “networks.devs.com” es 6.0.0.1. A continuación se muestran los logs de la simulación que muestra todo lo ocurrido para conseguir resolver el pedido, involucrando a todas las capas y dispositivos de la red.

Output de la capa DNS generado por el logger durante la simulación

```

[INFO] - [DNS Host 1] DNS local ip: 1.0.0.1
[INFO] - [DNS Host 1] DNS root server ip: 1.0.0.2
[INFO] - [DNS Host 1] No authoritative RR table file.
[INFO] - [DNS Host 1] No zone server table file.
[INFO] - [DNS Host 1] Recursive allowed: True
[INFO] - [DNS Host 3] DNS local ip: 4.0.0.1
[INFO] - [DNS Host 3] DNS root server ip: 5.0.0.1
[INFO] - [DNS Host 3] Authoritative RR table:
[INFO] - [DNS Host 3] network.devs.com A 6.0.0.1 IN 65535
[INFO] - [DNS Host 3] No zone server table file.
[INFO] - [DNS Host 3] Recursive allowed: True
[INFO] - [DNS Host 4] DNS local ip: 5.0.0.2
[INFO] - [DNS Host 4] DNS root server ip: 5.0.0.1
[INFO] - [DNS Host 4] No authoritative RR table file.
[INFO] - [DNS Host 4]
Zone _____
** NS **
devs.com NS ns1.devs.com IN 65535
** A **
ns1.dev.com A 4.0.0.1 IN 65535
Zone _____
[INFO] - [DNS Host 4] Zone servers table:
[INFO] - [DNS Host 4]
Zone _____
** NS **
devs.com NS ns1.devs.com IN 65535
** A **
ns1.dev.com A 4.0.0.1 IN 65535
Zone _____
[INFO] - [DNS Host 4] Recursive allowed: False
[INFO] - [DNS Host 5] DNS local ip: 5.0.0.1
[INFO] - [DNS Host 5] DNS root server ip: 5.0.0.1
[INFO] - [DNS Host 5] No authoritative RR table file.
[INFO] - [DNS Host 5]
Zone _____
** NS **
com NS ns1.com IN 65535
** A **
ns1.com A 5.0.0.2 IN 65535
Zone _____
[INFO] - [DNS Host 5] Zone servers table:
[INFO] - [DNS Host 5]
Zone _____
** NS **
com NS ns1.com IN 65535
** A **
ns1.com A 5.0.0.2 IN 65535
Zone _____
[INFO] - [DNS Host 5] Recursive allowed: False
[INFO] - [DNS Host 2] DNS local ip: 1.0.0.2
[INFO] - [DNS Host 2] DNS root server ip: 5.0.0.1
[INFO] - [DNS Host 2] No authoritative RR table file.
[INFO] - [DNS Host 2] No zone server table file.
[INFO] - [DNS Host 2] Recursive allowed: True
Simulation Initialized
[INFO] - [DNS Host 1] Process domain name network.devs.com
[INFO] - [DNS Host 1] Domain not in cache
[INFO] - [DNS Host 1] Send DNS Query to: (1.0.0.2, 53)
[INFO] - [DNS Host 2] Incoming packet:
[INFO] - [DNS Host 2]
dns packet _____
** HEADER **
id: 0
flags_code: 33024
QDCount: 1
ANCount: 0
NSCount: 0

```

```

ARCount: 1

** QUESTION SECTION **
network.devs.com A 0.0.0.0 IN 0

** ANSWER SECTION **

** AUTHORITATIVES SECTION **

** ADDITIONAL SECTION **
network.devs.com A 1.0.0.1 IN 9227
_____
[INFO] - [DNS Host 2] Query packet
[INFO] - [DNS Host 2] Recursive query for domain network.devs.com
[INFO] - [DNS Host 2] Send DNS Query to: (5.0.0.1, 53)
[INFO] - [DNS Host 5] Incoming packet:
[INFO] - [DNS Host 5]
_____  

dns packet

** HEADER **
id: 0
flags_code: 33024
QDCount: 1
ANCount: 0
NSCount: 0
ARCount: 1

** QUESTION SECTION **
network.devs.com A 0.0.0.0 IN 0

** ANSWER SECTION **

** AUTHORITATIVES SECTION **

** ADDITIONAL SECTION **
network.devs.com A 1.0.0.2 IN 9227
_____
[INFO] - [DNS Host 5] Query packet
[INFO] - [DNS Host 5] No recursive query for domain network.devs.com
[INFO] - [DNS Host 5] Zone found:
_____  

Zone

** NS **
com NS ns1.com IN 65535
** A **
ns1.com A 5.0.0.2 IN 65535
_____
[INFO] - [DNS Host 5] Send DNS Query to: (1.0.0.2, 53)
[INFO] - [DNS Host 2] Incoming packet:
[INFO] - [DNS Host 2]
_____  

dns packet

** HEADER **
id: 0
flags_code: 256
QDCount: 1
ANCount: 1
NSCount: 1
ARCount: 1

** QUESTION SECTION **
network.devs.com A 0.0.0.0 IN 0

** ANSWER SECTION **
com NS ns1.com IN 65535

** AUTHORITATIVES SECTION **
ns1.com A 5.0.0.2 IN 65535

** ADDITIONAL SECTION **

```

```

network.devs.com A 5.0.0.1 IN 9227
[INFO] - [DNS Host 2] Answer packet
[INFO] - [DNS Host 2] Answer Type: NS
[INFO] - [DNS Host 2] Send DNS Query to: (5.0.0.2, 53)
[INFO] - [DNS Host 4] Incoming packet:
[INFO] - [DNS Host 4]
----- dns packet -----
** HEADER **
id: 0
flags_code: 33024
QDCount: 1
ANCount: 0
NSCount: 0
ARCount: 1

** QUESTION SECTION **
network.devs.com A 0.0.0.0 IN 0

** ANSWER SECTION **

** AUTHORITATIVES SECTION **

** ADDITIONAL SECTION **
network.devs.com A 1.0.0.2 IN 9227
[INFO] - [DNS Host 4] Query packet
[INFO] - [DNS Host 4] No recursive query for domain network.devs.com
[INFO] - [DNS Host 4] Zone found:
----- Zone -----
** NS **
devs.com NS ns1.devs.com IN 65535
** A **
ns1.dev.com A 4.0.0.1 IN 65535
----- dns packet -----
[INFO] - [DNS Host 4] Send DNS Query to: (1.0.0.2, 53)
[INFO] - [DNS Host 2] Incoming packet:
[INFO] - [DNS Host 2]
----- dns packet -----
** HEADER **
id: 0
flags_code: 256
QDCount: 1
ANCount: 1
NSCount: 1
ARCount: 1

** QUESTION SECTION **
network.devs.com A 0.0.0.0 IN 0

** ANSWER SECTION **
devs.com NS ns1.devs.com IN 65535

** AUTHORITATIVES SECTION **
ns1.dev.com A 4.0.0.1 IN 65535

** ADDITIONAL SECTION **
network.devs.com A 5.0.0.2 IN 9227
[INFO] - [DNS Host 2] Answer packet
[INFO] - [DNS Host 2] Answer Type: NS
[INFO] - [DNS Host 2] Send DNS Query to: (4.0.0.1, 53)
[INFO] - [DNS Host 3] Incoming packet:
[INFO] - [DNS Host 3]
----- dns packet -----
** HEADER **
id: 0

```

```

flags_code: 33024
QDCount: 1
ANCount: 0
NSCount: 0
ARCount: 1

** QUESTION SECTION **
network.devs.com A 0.0.0.0 IN 0

** ANSWER SECTION **

** AUTHORITATIVES SECTION **

** ADDITIONAL SECTION **
network.devs.com A 1.0.0.2 IN 9227
[INFO] - [DNS Host 3] Query packet
[INFO] - [DNS Host 3] Resource for domain network.devs.com exist
[INFO] - [DNS Host 3] Send DNS Query to: (1.0.0.2, 53)
[INFO] - [DNS Host 2] Incoming packet:
[INFO] - [DNS Host 2] dns packet ----

** HEADER **
id: 0
flags_code: 1280
QDCount: 1
ANCount: 1
NSCount: 0
ARCount: 1

** QUESTION SECTION **
network.devs.com A 0.0.0.0 IN 0

** ANSWER SECTION **
network.devs.com A 6.0.0.1 IN 65535

** AUTHORITATIVES SECTION **

** ADDITIONAL SECTION **
network.devs.com A 4.0.0.1 IN 9227
[INFO] - [DNS Host 2] Answer packet
[INFO] - [DNS Host 2] Answer Type: A
[INFO] - [DNS Host 2] Caching authoritative answer
[INFO] - [DNS Host 2] Send DNS Query to: (1.0.0.1, 53)
[INFO] - [DNS Host 1] Incoming packet:
[INFO] - [DNS Host 1] dns packet ----

** HEADER **
id: 0
flags_code: 1280
QDCount: 1
ANCount: 1
NSCount: 0
ARCount: 1

** QUESTION SECTION **
network.devs.com A 0.0.0.0 IN 0

** ANSWER SECTION **
network.devs.com A 6.0.0.1 IN 65535

** AUTHORITATIVES SECTION **

** ADDITIONAL SECTION **
network.devs.com A 1.0.0.2 IN 9227

```

```
[INFO] - [DNS Host 1] Answer packet
[INFO] - [DNS Host 1] Answer Type: A
[INFO] - [DNS Host 1] Caching authoritative answer
```

Output de la capa UDP generado por el logger durante la simulación

```
Simulation Initialized
[INFO] - [UDP Host 1] Bind
[INFO] - [UDP Host 1] SUCCESS
[INFO] - [UDP Host 3] Bind
[INFO] - [UDP Host 3] SUCCESS
[INFO] - [UDP Host 4] Bind
[INFO] - [UDP Host 4] SUCCESS
[INFO] - [UDP Host 5] Bind
[INFO] - [UDP Host 5] SUCCESS
[INFO] - [UDP Host 2] Bind
[INFO] - [UDP Host 2] SUCCESS
[INFO] - [UDP Host 1] Read/Recv
[INFO] - [UDP Host 3] Read/Recv
[INFO] - [UDP Host 4] Read/Recv
[INFO] - [UDP Host 5] Read/Recv
[INFO] - [UDP Host 2] Read/Recv
[INFO] - [UDP Host 1] Write_to/Send_to
[INFO] - [UDP Host 2] process Segment:
[INFO] - [UDP Host 2] local_port: 53
[INFO] - [UDP Host 2] remote_port: 53
[INFO] - [UDP Host 2] local_ip: 1.0.0.2
[INFO] - [UDP Host 2] remote_ip: 1.0.0.1
[INFO] - [UDP Host 2] valid checksum: true
[INFO] - [UDP Host 2] socket local_ip: 1.0.0.2
[INFO] - [UDP Host 2] socket local_ip: 53
[INFO] - [UDP Host 2] Deliver dns packet
[INFO] - [UDP Host 2] Write_to/Send_to
[INFO] - [UDP Host 2] Read/Recv
[INFO] - [UDP Host 5] process Segment:
[INFO] - [UDP Host 5] local_port: 53
[INFO] - [UDP Host 5] remote_port: 53
[INFO] - [UDP Host 5] local_ip: 5.0.0.1
[INFO] - [UDP Host 5] remote_ip: 1.0.0.2
[INFO] - [UDP Host 5] valid checksum: true
[INFO] - [UDP Host 5] socket local_ip: 5.0.0.1
[INFO] - [UDP Host 5] socket local_ip: 53
[INFO] - [UDP Host 5] Deliver dns packet
[INFO] - [UDP Host 5] Write_to/Send_to
[INFO] - [UDP Host 5] Read/Recv
[INFO] - [UDP Host 2] process Segment:
[INFO] - [UDP Host 2] local_port: 53
[INFO] - [UDP Host 2] remote_port: 53
[INFO] - [UDP Host 2] local_ip: 1.0.0.2
[INFO] - [UDP Host 2] remote_ip: 5.0.0.1
[INFO] - [UDP Host 2] valid checksum: true
[INFO] - [UDP Host 2] socket local_ip: 1.0.0.2
[INFO] - [UDP Host 2] socket local_ip: 53
[INFO] - [UDP Host 2] Deliver dns packet
[INFO] - [UDP Host 2] Write_to/Send_to
[INFO] - [UDP Host 2] Read/Recv
[INFO] - [UDP Host 4] process Segment:
[INFO] - [UDP Host 4] local_port: 53
[INFO] - [UDP Host 4] remote_port: 53
[INFO] - [UDP Host 4] local_ip: 5.0.0.2
[INFO] - [UDP Host 4] remote_ip: 1.0.0.2
[INFO] - [UDP Host 4] valid checksum: true
[INFO] - [UDP Host 4] socket local_ip: 5.0.0.2
[INFO] - [UDP Host 4] socket local_ip: 53
[INFO] - [UDP Host 4] Deliver dns packet
[INFO] - [UDP Host 4] Write_to/Send_to
[INFO] - [UDP Host 4] Read/Recv
[INFO] - [UDP Host 2] process Segment:
```

```

[INFO] - [UDP Host 2] local_port: 53
[INFO] - [UDP Host 2] remote_port: 53
[INFO] - [UDP Host 2] local_ip: 1.0.0.2
[INFO] - [UDP Host 2] remote_ip: 5.0.0.2
[INFO] - [UDP Host 2] valid checksum: true
[INFO] - [UDP Host 2] socket local_ip: 1.0.0.2
[INFO] - [UDP Host 2] socket local_ip: 53
[INFO] - [UDP Host 2] Deliver dns packet
[INFO] - [UDP Host 2] Write_to/Send_to
[INFO] - [UDP Host 2] Read/Recv
[INFO] - [UDP Host 3] process Segment:
[INFO] - [UDP Host 3] local_port: 53
[INFO] - [UDP Host 3] remote_port: 53
[INFO] - [UDP Host 3] local_ip: 4.0.0.1
[INFO] - [UDP Host 3] remote_ip: 1.0.0.2
[INFO] - [UDP Host 3] valid checksum: true
[INFO] - [UDP Host 3] socket local_ip: 4.0.0.1
[INFO] - [UDP Host 3] socket local_ip: 53
[INFO] - [UDP Host 3] Deliver dns packet
[INFO] - [UDP Host 3] Write_to/Send_to
[INFO] - [UDP Host 3] Read/Recv
[INFO] - [UDP Host 2] process Segment:
[INFO] - [UDP Host 2] local_port: 53
[INFO] - [UDP Host 2] remote_port: 53
[INFO] - [UDP Host 2] local_ip: 1.0.0.2
[INFO] - [UDP Host 2] remote_ip: 4.0.0.1
[INFO] - [UDP Host 2] valid checksum: true
[INFO] - [UDP Host 2] socket local_ip: 1.0.0.2
[INFO] - [UDP Host 2] socket local_ip: 53
[INFO] - [UDP Host 2] Deliver dns packet
[INFO] - [UDP Host 2] Write_to/Send_to
[INFO] - [UDP Host 2] Read/Recv
[INFO] - [UDP Host 1] process Segment:
[INFO] - [UDP Host 1] local_port: 53
[INFO] - [UDP Host 1] remote_port: 53
[INFO] - [UDP Host 1] local_ip: 1.0.0.1
[INFO] - [UDP Host 1] remote_ip: 1.0.0.2
[INFO] - [UDP Host 1] valid checksum: true
[INFO] - [UDP Host 1] socket local_ip: 1.0.0.1
[INFO] - [UDP Host 1] socket local_ip: 53
[INFO] - [UDP Host 1] Deliver dns packet
[INFO] - [UDP Host 1] Read/Recv

```

Output de la capa IP generado por el logger durante la simulación

```

[INFO] - [IPv4 Host 1] ip amount: 2
[INFO] - [IPv4 Host 1] ips:
[INFO] - [IPv4 Host 1] 1.0.0.1
[INFO] - [IPv4 Host 1] 127.0.0.1
[INFO] - [IPv4 Host 1] Routing table:
[INFO] - [IPv4 Host 1] | 0.0.0.0 | 0.0.0.0 | 1.0.0.3 | 1 | Default Route |
[INFO] - [IPv4 Host 1] | 1.0.0.2 | 255.255.255.255 | 1.0.0.2 | 1 | Host 2 |
[INFO] - [IPv4 Host 1] Forwarding table:
[INFO] - [IPv4 Host 1] | 0.0.0.0 | 0.0.0.0 | 0 |
[INFO] - [IPv4 Host 3] ip amount: 2
[INFO] - [IPv4 Host 3] ips:
[INFO] - [IPv4 Host 3] 4.0.0.1
[INFO] - [IPv4 Host 3] 127.0.0.1
[INFO] - [IPv4 Host 3] Routing table:
[INFO] - [IPv4 Host 3] | 0.0.0.0 | 0.0.0.0 | 4.0.0.0 | 1 | Default Route |
[INFO] - [IPv4 Host 3] Forwarding table:
[INFO] - [IPv4 Host 3] | 0.0.0.0 | 0.0.0.0 | 0 |
[INFO] - [IPv4 Host 4] ip amount: 2
[INFO] - [IPv4 Host 4] ips:
[INFO] - [IPv4 Host 4] 5.0.0.2
[INFO] - [IPv4 Host 4] 127.0.0.1
[INFO] - [IPv4 Host 4] Routing table:
[INFO] - [IPv4 Host 4] | 0.0.0.0 | 0.0.0.0 | 5.0.0.0 | 1 | Default Route |

```

```

[INFO] - [IPv4 Host 4] | 5.0.0.1 | 255.255.255.255 | 5.0.0.1 | 1 | Host 5 |
[INFO] - [IPv4 Host 4] Forwarding table:
[INFO] - [IPv4 Host 4] | 0.0.0.0 | 0.0.0.0 | 0 |
[INFO] - [IPv4 Host 5] ip amount: 2
[INFO] - [IPv4 Host 5] ips:
[INFO] - [IPv4 Host 5] 5.0.0.1
[INFO] - [IPv4 Host 5] 127.0.0.1
[INFO] - [IPv4 Host 5] Routing table:
[INFO] - [IPv4 Host 5] | 0.0.0.0 | 0.0.0.0 | 5.0.0.0 | 1 | Default Route |
[INFO] - [IPv4 Host 5] | 5.0.0.2 | 255.255.255.255 | 5.0.0.2 | 1 | Host 4 |
[INFO] - [IPv4 Host 5] Forwarding table:
[INFO] - [IPv4 Host 5] | 0.0.0.0 | 0.0.0.0 | 0 |
[INFO] - [IPv4 Router 1] ip amount: 3
[INFO] - [IPv4 Router 1] ips:
[INFO] - [IPv4 Router 1] 1.0.0.0
[INFO] - [IPv4 Router 1] 2.0.0.0
[INFO] - [IPv4 Router 1] 3.0.0.0
[INFO] - [IPv4 Router 1] Routing table:
[INFO] - [IPv4 Router 1] | 0.0.0.0 | 0.0.0.0 | 3.0.0.1 | 1 | Default gateway |
[INFO] - [IPv4 Router 1] | 5.0.0.0 | 255.0.0.0 | 3.0.0.1 | 2 | Router 2 |
[INFO] - [IPv4 Router 1] | 4.0.0.0 | 255.0.0.0 | 2.0.0.1 | 2 | Router 3 |
[INFO] - [IPv4 Router 1] | 1.0.0.1 | 255.255.255.255 | 1.0.0.1 | 1 | Host 1 |
[INFO] - [IPv4 Router 1] | 1.0.0.2 | 255.255.255.255 | 1.0.0.2 | 1 | Host 2 |
[INFO] - [IPv4 Router 1] Forwarding table:
[INFO] - [IPv4 Router 1] | 1.0.0.0 | 255.0.0.0 | 0 |
[INFO] - [IPv4 Router 1] | 3.0.0.0 | 255.0.0.0 | 1 |
[INFO] - [IPv4 Router 1] | 2.0.0.0 | 255.0.0.0 | 2 |
[INFO] - [IPv4 Router 2] ip amount: 2
[INFO] - [IPv4 Router 2] ips:
[INFO] - [IPv4 Router 2] 3.0.0.1
[INFO] - [IPv4 Router 2] 5.0.0.0
[INFO] - [IPv4 Router 2] Routing table:
[INFO] - [IPv4 Router 2] | 0.0.0.0 | 0.0.0.0 | 3.0.0.0 | 1 | Default gateway |
[INFO] - [IPv4 Router 2] | 1.0.0.0 | 255.0.0.0 | 3.0.0.0 | 2 | Router 1 |
[INFO] - [IPv4 Router 2] | 2.0.0.0 | 255.0.0.0 | 3.0.0.0 | 2 | Router 1 |
[INFO] - [IPv4 Router 2] | 4.0.0.0 | 255.0.0.0 | 3.0.0.0 | 3 | Router 1 |
[INFO] - [IPv4 Router 2] | 5.0.0.1 | 255.255.255.255 | 5.0.0.1 | 1 | Host 5 |
[INFO] - [IPv4 Router 2] | 5.0.0.2 | 255.255.255.255 | 5.0.0.2 | 1 | Host 4 |
[INFO] - [IPv4 Router 2] Forwarding table:
[INFO] - [IPv4 Router 2] | 3.0.0.0 | 255.0.0.0 | 0 |
[INFO] - [IPv4 Router 2] | 5.0.0.0 | 255.0.0.0 | 1 |
[INFO] - [IPv4 Router 3] ip amount: 2
[INFO] - [IPv4 Router 3] ips:
[INFO] - [IPv4 Router 3] 2.0.0.1
[INFO] - [IPv4 Router 3] 4.0.0.0
[INFO] - [IPv4 Router 3] Routing table:
[INFO] - [IPv4 Router 3] | 0.0.0.0 | 0.0.0.0 | 2.0.0.0 | 1 | Default gateway |
[INFO] - [IPv4 Router 3] | 1.0.0.0 | 255.0.0.0 | 2.0.0.0 | 2 | Router 1 |
[INFO] - [IPv4 Router 3] | 3.0.0.0 | 255.0.0.0 | 2.0.0.0 | 2 | Router 1 |
[INFO] - [IPv4 Router 3] | 5.0.0.0 | 255.0.0.0 | 2.0.0.0 | 3 | Router 1 |
[INFO] - [IPv4 Router 3] | 4.0.0.1 | 255.255.255.255 | 4.0.0.1 | 1 | Host 3 |
[INFO] - [IPv4 Router 3] Forwarding table:
[INFO] - [IPv4 Router 3] | 2.0.0.0 | 255.0.0.0 | 0 |
[INFO] - [IPv4 Router 3] | 4.0.0.0 | 255.0.0.0 | 1 |
[INFO] - [IPv4 Host 2] ip amount: 2
[INFO] - [IPv4 Host 2] ips:
[INFO] - [IPv4 Host 2] 1.0.0.2
[INFO] - [IPv4 Host 2] 127.0.0.1
[INFO] - [IPv4 Host 2] Routing table:
[INFO] - [IPv4 Host 2] | 0.0.0.0 | 0.0.0.0 | 1.0.0.0 | 1 | Default Route |
[INFO] - [IPv4 Host 2] | 1.0.0.1 | 255.255.255.255 | 1.0.0.1 | 1 | Host 1 |
[INFO] - [IPv4 Host 2] Forwarding table:
[INFO] - [IPv4 Host 2] | 0.0.0.0 | 0.0.0.0 | 0 |
Simulation Initialized
[INFO] - [IPv4 Host 1] Forwarding ip packet: 1.0.0.2
[INFO] - [IPv4 Host 1] Best route for packet with dest_ip: 1.0.0.2
is: | 1.0.0.2 | 255.255.255.255 | 1.0.0.2 | 1 | Host 2 |

```

```

[INFO] - [IPv4 Host 1] Added packet to wait ARP for nexthop: 1.0.0.2
[INFO] - [IPv4 Host 1] ARP query throw interface: NET 0
[INFO] - [IPv4 Host 1] link::Ctrl::ARP_READY
[INFO] - [IPv4 Host 2] Processing arrived ip packet: 1.0.0.2
[INFO] - [IPv4 Host 2] Valid checksum: true
[INFO] - [IPv4 Host 2] Forwarding ip packet: 5.0.0.1
[INFO] - [IPv4 Host 2] Best route for packet with dest_ip: 5.0.0.1
is: | 0.0.0.0 | 0.0.0.0 | 1.0.0.0 | 1 | Default Route |
[INFO] - [IPv4 Host 2] Added packet to wait ARP for nexthop: 1.0.0.0
[INFO] - [IPv4 Host 2] ARP query throw interface: NET 0
[INFO] - [IPv4 Host 2] link::Ctrl::ARP_READY
[INFO] - [IPv4 Router 1] Process Ip Datagram
[INFO] - [IPv4 Router 1] Valid checksum: true
[INFO] - [IPv4 Router 1] Forwarding ip packet: 5.0.0.1
[INFO] - [IPv4 Router 1] Best route for packet with dest_ip: 5.0.0.1
is: | 5.0.0.0 | 255.0.0.0 | 3.0.0.1 | 2 | Router 2 |
[INFO] - [IPv4 Router 1] Added packet to wait ARP for nexthop: 3.0.0.1
[INFO] - [IPv4 Router 1] ARP query throw interface: NET 1
[INFO] - [IPv4 Router 1] link::Ctrl::ARP_READY
[INFO] - [IPv4 Router 2] Process Ip Datagram
[INFO] - [IPv4 Router 2] Valid checksum: true
[INFO] - [IPv4 Router 2] Forwarding ip packet: 5.0.0.1
[INFO] - [IPv4 Router 2] Best route for packet with dest_ip: 5.0.0.1
is: | 5.0.0.1 | 255.255.255.255 | 5.0.0.1 | 1 | Host 5 |
[INFO] - [IPv4 Router 2] Added packet to wait ARP for nexthop: 5.0.0.1
[INFO] - [IPv4 Router 2] ARP query throw interface: NET 1
[INFO] - [IPv4 Router 2] link::Ctrl::ARP_READY
[INFO] - [IPv4 Host 5] Processing arrived ip packet: 5.0.0.1
[INFO] - [IPv4 Host 5] Valid checksum: true
[INFO] - [IPv4 Host 5] Forwarding ip packet: 1.0.0.2
[INFO] - [IPv4 Host 5] Best route for packet with dest_ip: 1.0.0.2
is: | 0.0.0.0 | 0.0.0.0 | 5.0.0.0 | 1 | Default Route |
[INFO] - [IPv4 Host 5] Added packet to wait ARP for nexthop: 5.0.0.0
[INFO] - [IPv4 Host 5] ARP query throw interface: NET 0
[INFO] - [IPv4 Host 5] link::Ctrl::ARP_READY
[INFO] - [IPv4 Router 2] Process Ip Datagram
[INFO] - [IPv4 Router 2] Valid checksum: true
[INFO] - [IPv4 Router 2] Forwarding ip packet: 1.0.0.2
[INFO] - [IPv4 Router 2] Best route for packet with dest_ip: 1.0.0.2
is: | 1.0.0.0 | 255.0.0.0 | 3.0.0.0 | 2 | Router 1 |
[INFO] - [IPv4 Router 2] Added packet to wait ARP for nexthop: 3.0.0.0
[INFO] - [IPv4 Router 2] ARP query throw interface: NET 0
[INFO] - [IPv4 Router 2] link::Ctrl::ARP_READY
[INFO] - [IPv4 Router 1] Process Ip Datagram
[INFO] - [IPv4 Router 1] Valid checksum: true
[INFO] - [IPv4 Router 1] Forwarding ip packet: 1.0.0.2
[INFO] - [IPv4 Router 1] Best route for packet with dest_ip: 1.0.0.2
is: | 1.0.0.2 | 255.255.255.255 | 1.0.0.2 | 1 | Host 2 |
[INFO] - [IPv4 Router 1] Added packet to wait ARP for nexthop: 1.0.0.2
[INFO] - [IPv4 Router 1] ARP query throw interface: NET 0
[INFO] - [IPv4 Router 1] link::Ctrl::ARP_READY
[INFO] - [IPv4 Host 2] Processing arrived ip packet: 1.0.0.2
[INFO] - [IPv4 Host 2] Valid checksum: true
[INFO] - [IPv4 Host 2] Forwarding ip packet: 5.0.0.2
[INFO] - [IPv4 Host 2] Best route for packet with dest_ip: 5.0.0.2
is: | 0.0.0.0 | 0.0.0.0 | 1.0.0.0 | 1 | Default Route |
[INFO] - [IPv4 Host 2] Added packet to wait ARP for nexthop: 1.0.0.0
[INFO] - [IPv4 Host 2] ARP query throw interface: NET 0
[INFO] - [IPv4 Host 2] link::Ctrl::ARP_READY
[INFO] - [IPv4 Router 1] Process Ip Datagram
[INFO] - [IPv4 Router 1] Valid checksum: true
[INFO] - [IPv4 Router 1] Forwarding ip packet: 5.0.0.2
[INFO] - [IPv4 Router 1] Best route for packet with dest_ip: 5.0.0.2
is: | 5.0.0.0 | 255.0.0.0 | 3.0.0.1 | 2 | Router 2 |
[INFO] - [IPv4 Router 1] Added packet to wait ARP for nexthop: 3.0.0.1
[INFO] - [IPv4 Router 1] ARP query throw interface: NET 1
[INFO] - [IPv4 Router 1] link::Ctrl::ARP_READY

```

```

[INFO] - [IPv4 Router 2] Process Ip Datagram
[INFO] - [IPv4 Router 2] Valid checksum: true
[INFO] - [IPv4 Router 2] Forwarding ip packet: 5.0.0.2
[INFO] - [IPv4 Router 2] Best route for packet with dest_ip: 5.0.0.2
is: | 5.0.0.2 | 255.255.255.255 | 5.0.0.2 | 1 | Host 4 |
[INFO] - [IPv4 Router 2] Added packet to wait ARP for nexthop: 5.0.0.2
[INFO] - [IPv4 Router 2] ARP query throw interface: NET 1
[INFO] - [IPv4 Router 2] link :: Ctrl :: ARP_READY
[INFO] - [IPv4 Host 4] Processing arrived ip packet: 5.0.0.2
[INFO] - [IPv4 Host 4] Valid checksum: true
[INFO] - [IPv4 Host 4] Forwarding ip packet: 1.0.0.2
[INFO] - [IPv4 Host 4] Best route for packet with dest_ip: 1.0.0.2
is: | 0.0.0.0 | 0.0.0.0 | 5.0.0.0 | 1 | Default Route |
[INFO] - [IPv4 Host 4] Added packet to wait ARP for nexthop: 5.0.0.0
[INFO] - [IPv4 Host 4] ARP query throw interface: NET 0
[INFO] - [IPv4 Host 4] link :: Ctrl :: ARP_READY
[INFO] - [IPv4 Router 2] Process Ip Datagram
[INFO] - [IPv4 Router 2] Valid checksum: true
[INFO] - [IPv4 Router 2] Forwarding ip packet: 1.0.0.2
[INFO] - [IPv4 Router 2] Best route for packet with dest_ip: 1.0.0.2
is: | 1.0.0.0 | 255.0.0.0 | 3.0.0.0 | 2 | Router 1 |
[INFO] - [IPv4 Router 2] Added packet to wait ARP for nexthop: 3.0.0.0
[INFO] - [IPv4 Router 2] ARP query throw interface: NET 0
[INFO] - [IPv4 Router 2] link :: Ctrl :: ARP_READY
[INFO] - [IPv4 Router 1] Process Ip Datagram
[INFO] - [IPv4 Router 1] Valid checksum: true
[INFO] - [IPv4 Router 1] Forwarding ip packet: 1.0.0.2
[INFO] - [IPv4 Router 1] Best route for packet with dest_ip: 1.0.0.2
is: | 1.0.0.2 | 255.255.255.255 | 1.0.0.2 | 1 | Host 2 |
[INFO] - [IPv4 Router 1] Added packet to wait ARP for nexthop: 1.0.0.2
[INFO] - [IPv4 Router 1] ARP query throw interface: NET 0
[INFO] - [IPv4 Router 1] link :: Ctrl :: ARP_READY
[INFO] - [IPv4 Host 2] Processing arrived ip packet: 1.0.0.2
[INFO] - [IPv4 Host 2] Valid checksum: true
[INFO] - [IPv4 Host 2] Forwarding ip packet: 4.0.0.1
[INFO] - [IPv4 Host 2] Best route for packet with dest_ip: 4.0.0.1
is: | 0.0.0.0 | 0.0.0.0 | 1.0.0.0 | 1 | Default Route |
[INFO] - [IPv4 Host 2] Added packet to wait ARP for nexthop: 1.0.0.0
[INFO] - [IPv4 Host 2] ARP query throw interface: NET 0
[INFO] - [IPv4 Host 2] link :: Ctrl :: ARP_READY
[INFO] - [IPv4 Router 1] Process Ip Datagram
[INFO] - [IPv4 Router 1] Valid checksum: true
[INFO] - [IPv4 Router 1] Forwarding ip packet: 4.0.0.1
[INFO] - [IPv4 Router 1] Best route for packet with dest_ip: 4.0.0.1
is: | 4.0.0.0 | 255.0.0.0 | 2.0.0.1 | 2 | Router 3 |
[INFO] - [IPv4 Router 1] Added packet to wait ARP for nexthop: 2.0.0.1
[INFO] - [IPv4 Router 1] ARP query throw interface: NET 2
[INFO] - [IPv4 Router 1] link :: Ctrl :: ARP_READY
[INFO] - [IPv4 Router 3] Process Ip Datagram
[INFO] - [IPv4 Router 3] Valid checksum: true
[INFO] - [IPv4 Router 3] Forwarding ip packet: 4.0.0.1
[INFO] - [IPv4 Router 3] Best route for packet with dest_ip: 4.0.0.1
is: | 4.0.0.1 | 255.255.255.255 | 4.0.0.1 | 1 | Host 3 |
[INFO] - [IPv4 Router 3] Added packet to wait ARP for nexthop: 4.0.0.1
[INFO] - [IPv4 Router 3] ARP query throw interface: NET 1
[INFO] - [IPv4 Router 3] link :: Ctrl :: ARP_READY
[INFO] - [IPv4 Host 3] Processing arrived ip packet: 4.0.0.1
[INFO] - [IPv4 Host 3] Valid checksum: true
[INFO] - [IPv4 Host 3] Forwarding ip packet: 1.0.0.2
[INFO] - [IPv4 Host 3] Best route for packet with dest_ip: 1.0.0.2
is: | 0.0.0.0 | 0.0.0.0 | 4.0.0.0 | 1 | Default Route |
[INFO] - [IPv4 Host 3] Added packet to wait ARP for nexthop: 4.0.0.0
[INFO] - [IPv4 Host 3] ARP query throw interface: NET 0
[INFO] - [IPv4 Host 3] link :: Ctrl :: ARP_READY
[INFO] - [IPv4 Router 3] Process Ip Datagram
[INFO] - [IPv4 Router 3] Valid checksum: true
[INFO] - [IPv4 Router 3] Forwarding ip packet: 1.0.0.2

```

```

[INFO] - [IPv4 Router 3] Best route for packet with dest_ip: 1.0.0.2
is : | 1.0.0.0 | 255.0.0.0 | 2.0.0.0 | 2 | Router 1 |
[INFO] - [IPv4 Router 3] Added packet to wait ARP for nexthop: 2.0.0.0
[INFO] - [IPv4 Router 3] ARP query throw interface: NET 0
[INFO] - [IPv4 Router 3] link :: Ctrl :: ARP_READY
[INFO] - [IPv4 Router 1] Process Ip Datagram
[INFO] - [IPv4 Router 1] Valid checksum: true
[INFO] - [IPv4 Router 1] Forwarding ip packet: 1.0.0.2
[INFO] - [IPv4 Router 1] Best route for packet with dest_ip: 1.0.0.2
is : | 1.0.0.2 | 255.255.255.255 | 1.0.0.2 | 1 | Host 2 |
[INFO] - [IPv4 Router 1] Added packet to wait ARP for nexthop: 1.0.0.2
[INFO] - [IPv4 Router 1] ARP query throw interface: NET 0
[INFO] - [IPv4 Router 1] link :: Ctrl :: ARP_READY
[INFO] - [IPv4 Host 2] Processing arrived ip packet: 1.0.0.2
[INFO] - [IPv4 Host 2] Valid checksum: true
[INFO] - [IPv4 Host 2] Forwarding ip packet: 1.0.0.1
[INFO] - [IPv4 Host 2] Best route for packet with dest_ip: 1.0.0.1
is : | 1.0.0.1 | 255.255.255.255 | 1.0.0.1 | 1 | Host 1 |
[INFO] - [IPv4 Host 2] Added packet to wait ARP for nexthop: 1.0.0.1
[INFO] - [IPv4 Host 2] ARP query throw interface: NET 0
[INFO] - [IPv4 Host 2] link :: Ctrl :: ARP_READY
[INFO] - [IPv4 Host 1] Processing arrived ip packet: 1.0.0.1
[INFO] - [IPv4 Host 1] Valid checksum: true

```

Output del módulo LINK generado por el logger durante la simulación

```

[INFO] - [Link Host 1 NET 0] MAC: 0:0:0:0:0:1
[INFO] - [Link Host 1 NET 0] IP: 1.0.0.1
[INFO] - [Link Host 1 NET 0] Interface: 0
[INFO] - [Link Host 3 NET 0] MAC: 0:0:0:0:0:8
[INFO] - [Link Host 3 NET 0] IP: 4.0.0.1
[INFO] - [Link Host 3 NET 0] Interface: 0
[INFO] - [Link Host 4 NET 0] MAC: 0:0:0:0:0:12
[INFO] - [Link Host 4 NET 0] IP: 5.0.0.2
[INFO] - [Link Host 4 NET 0] Interface: 0
[INFO] - [Link Host 5 NET 0] MAC: 0:0:0:0:0:11
[INFO] - [Link Host 5 NET 0] IP: 5.0.0.1
[INFO] - [Link Host 5 NET 0] Interface: 0
[INFO] - [Link Router 1 NET 0] MAC: 0:0:0:0:0:3
[INFO] - [Link Router 1 NET 0] IP: 1.0.0.0
[INFO] - [Link Router 1 NET 0] Interface: 0
[INFO] - [Link Router 1 NET 1] MAC: 0:0:0:0:0:5
[INFO] - [Link Router 1 NET 1] IP: 3.0.0.0
[INFO] - [Link Router 1 NET 1] Interface: 1
[INFO] - [Link Router 1 NET 2] MAC: 0:0:0:0:0:4
[INFO] - [Link Router 1 NET 2] IP: 2.0.0.0
[INFO] - [Link Router 1 NET 2] Interface: 2
[INFO] - [Link Router 2 NET 0] MAC: 0:0:0:0:0:9
[INFO] - [Link Router 2 NET 0] IP: 3.0.0.1
[INFO] - [Link Router 2 NET 0] Interface: 0
[INFO] - [Link Router 2 NET 1] MAC: 0:0:0:0:0:10
[INFO] - [Link Router 2 NET 1] IP: 5.0.0.0
[INFO] - [Link Router 2 NET 1] Interface: 1
[INFO] - [Link Router 3 NET 0] MAC: 0:0:0:0:0:6
[INFO] - [Link Router 3 NET 0] IP: 2.0.0.1
[INFO] - [Link Router 3 NET 0] Interface: 0
[INFO] - [Link Router 3 NET 1] MAC: 0:0:0:0:0:7
[INFO] - [Link Router 3 NET 1] IP: 4.0.0.0
[INFO] - [Link Router 3 NET 1] Interface: 1
[INFO] - [Link Host 2 NET 0] MAC: 0:0:0:0:0:2
[INFO] - [Link Host 2 NET 0] IP: 1.0.0.2
[INFO] - [Link Host 2 NET 0] Interface: 0
Simulation Initialized
[INFO] - [Link Host 1 NET 0] ARP_QUERY
[INFO] - [Link Host 1 NET 0] ARP ip 1.0.0.2 not cached.
[INFO] - [Link Host 1 NET 0] Sending ARP query for ip: 1.0.0.2
[INFO] - [Link Host 2 NET 0] Process ARP packet.
[INFO] - [Link Router 1 NET 0] Process ARP packet.

```

```

[INFO] - [Link Host 1 NET 0] Process ARP packet.
[INFO] - [Link Host 1 NET 0] Send control ARP_READY
[INFO] - [Link Host 1 NET 0] SEND_PACKET
[INFO] - [Link Host 1 NET 0] ARP ip 1.0.0.2 is in cache.
[INFO] - [Link Host 2 NET 0] ARP_QUERY
[INFO] - [Link Host 2 NET 0] ARP ip 1.0.0.0 not cached.
[INFO] - [Link Host 2 NET 0] Sending ARP query for ip: 1.0.0.0
[INFO] - [Link Host 1 NET 0] Process ARP packet.
[INFO] - [Link Router 1 NET 0] Process ARP packet.
[INFO] - [Link Host 2 NET 0] Process ARP packet.
[INFO] - [Link Host 2 NET 0] Send control ARP_READY
[INFO] - [Link Host 2 NET 0] SEND_PACKET
[INFO] - [Link Host 2 NET 0] ARP ip 1.0.0.0 is in cache.
[INFO] - [Link Router 1 NET 1] ARP_QUERY
[INFO] - [Link Router 1 NET 1] ARP ip 3.0.0.1 not cached.
[INFO] - [Link Router 1 NET 1] Sending ARP query for ip: 3.0.0.1
[INFO] - [Link Router 2 NET 0] Process ARP packet.
[INFO] - [Link Router 1 NET 1] Process ARP packet.
[INFO] - [Link Router 1 NET 1] Send control ARP_READY
[INFO] - [Link Router 1 NET 1] SEND_PACKET
[INFO] - [Link Router 1 NET 1] ARP ip 3.0.0.1 is in cache.
[INFO] - [Link Router 2 NET 1] ARP_QUERY
[INFO] - [Link Router 2 NET 1] ARP ip 5.0.0.1 not cached.
[INFO] - [Link Router 2 NET 1] Sending ARP query for ip: 5.0.0.1
[INFO] - [Link Host 5 NET 0] Process ARP packet.
[INFO] - [Link Host 4 NET 0] Process ARP packet.
[INFO] - [Link Router 2 NET 1] Process ARP packet.
[INFO] - [Link Router 2 NET 1] Send control ARP_READY
[INFO] - [Link Router 2 NET 1] SEND_PACKET
[INFO] - [Link Router 2 NET 1] ARP ip 5.0.0.1 is in cache.
[INFO] - [Link Host 5 NET 0] ARP_QUERY
[INFO] - [Link Host 5 NET 0] ARP ip 5.0.0.0 not cached.
[INFO] - [Link Host 5 NET 0] Sending ARP query for ip: 5.0.0.0
[INFO] - [Link Router 2 NET 1] Process ARP packet.
[INFO] - [Link Host 4 NET 0] Process ARP packet.
[INFO] - [Link Host 5 NET 0] Process ARP packet.
[INFO] - [Link Host 5 NET 0] Send control ARP_READY
[INFO] - [Link Host 5 NET 0] SEND_PACKET
[INFO] - [Link Host 5 NET 0] ARP ip 5.0.0.0 is in cache.
[INFO] - [Link Router 2 NET 0] ARP_QUERY
[INFO] - [Link Router 2 NET 0] ARP ip 3.0.0.0 not cached.
[INFO] - [Link Router 2 NET 0] Sending ARP query for ip: 3.0.0.0
[INFO] - [Link Router 1 NET 1] Process ARP packet.
[INFO] - [Link Router 2 NET 0] Process ARP packet.
[INFO] - [Link Router 2 NET 0] Send control ARP_READY
[INFO] - [Link Router 2 NET 0] SEND_PACKET
[INFO] - [Link Router 2 NET 0] ARP ip 3.0.0.0 is in cache.
[INFO] - [Link Router 1 NET 0] ARP_QUERY
[INFO] - [Link Router 1 NET 0] ARP ip 1.0.0.2 not cached.
[INFO] - [Link Router 1 NET 0] Sending ARP query for ip: 1.0.0.2
[INFO] - [Link Host 2 NET 0] Process ARP packet.
[INFO] - [Link Host 1 NET 0] Process ARP packet.
[INFO] - [Link Router 1 NET 0] Process ARP packet.
[INFO] - [Link Router 1 NET 0] Send control ARP_READY
[INFO] - [Link Router 1 NET 0] SEND_PACKET
[INFO] - [Link Router 1 NET 0] ARP ip 1.0.0.2 is in cache.
[INFO] - [Link Host 2 NET 0] ARP_QUERY
[INFO] - [Link Host 2 NET 0] ARP ip 1.0.0.0 not cached.
[INFO] - [Link Host 2 NET 0] Sending ARP query for ip: 1.0.0.0
[INFO] - [Link Host 1 NET 0] Process ARP packet.
[INFO] - [Link Router 1 NET 0] Process ARP packet.
[INFO] - [Link Host 2 NET 0] Process ARP packet.
[INFO] - [Link Host 2 NET 0] Send control ARP_READY
[INFO] - [Link Host 2 NET 0] SEND_PACKET
[INFO] - [Link Host 2 NET 0] ARP ip 1.0.0.0 is in cache.
[INFO] - [Link Router 1 NET 1] ARP_QUERY
[INFO] - [Link Router 1 NET 1] ARP ip 3.0.0.1 not cached.

```

```

[INFO] - [Link Router 1 NET 1] Sending ARP query for ip: 3.0.0.1
[INFO] - [Link Router 2 NET 0] Process ARP packet.
[INFO] - [Link Router 1 NET 1] Process ARP packet.
[INFO] - [Link Router 1 NET 1] Send control ARP_READY
[INFO] - [Link Router 1 NET 1] SEND_PACKET
[INFO] - [Link Router 1 NET 1] ARP ip 3.0.0.1 is in cache.
[INFO] - [Link Router 2 NET 1] ARP_QUERY
[INFO] - [Link Router 2 NET 1] ARP ip 5.0.0.2 not cached.
[INFO] - [Link Router 2 NET 1] Sending ARP query for ip: 5.0.0.2
[INFO] - [Link Host 5 NET 0] Process ARP packet.
[INFO] - [Link Host 4 NET 0] Process ARP packet.
[INFO] - [Link Router 2 NET 1] Process ARP packet.
[INFO] - [Link Router 2 NET 1] Send control ARP_READY
[INFO] - [Link Router 2 NET 1] SEND_PACKET
[INFO] - [Link Router 2 NET 1] ARP ip 5.0.0.2 is in cache.
[INFO] - [Link Host 4 NET 0] ARP_QUERY
[INFO] - [Link Host 4 NET 0] ARP ip 5.0.0.0 not cached.
[INFO] - [Link Host 4 NET 0] Sending ARP query for ip: 5.0.0.0
[INFO] - [Link Router 2 NET 1] Process ARP packet.
[INFO] - [Link Host 5 NET 0] Process ARP packet.
[INFO] - [Link Host 4 NET 0] Process ARP packet.
[INFO] - [Link Host 4 NET 0] Send control ARP_READY
[INFO] - [Link Host 4 NET 0] SEND_PACKET
[INFO] - [Link Host 4 NET 0] ARP ip 5.0.0.0 is in cache.
[INFO] - [Link Router 2 NET 0] ARP_QUERY
[INFO] - [Link Router 2 NET 0] ARP ip 3.0.0.0 not cached.
[INFO] - [Link Router 2 NET 0] Sending ARP query for ip: 3.0.0.0
[INFO] - [Link Router 1 NET 1] Process ARP packet.
[INFO] - [Link Router 2 NET 0] Process ARP packet.
[INFO] - [Link Router 2 NET 0] Send control ARP_READY
[INFO] - [Link Router 2 NET 0] SEND_PACKET
[INFO] - [Link Router 2 NET 0] ARP ip 3.0.0.0 is in cache.
[INFO] - [Link Router 1 NET 0] ARP_QUERY
[INFO] - [Link Router 1 NET 0] ARP ip 1.0.0.2 not cached.
[INFO] - [Link Router 1 NET 0] Sending ARP query for ip: 1.0.0.2
[INFO] - [Link Host 2 NET 0] Process ARP packet.
[INFO] - [Link Host 1 NET 0] Process ARP packet.
[INFO] - [Link Router 1 NET 0] Process ARP packet.
[INFO] - [Link Router 1 NET 0] Send control ARP_READY
[INFO] - [Link Router 1 NET 0] SEND_PACKET
[INFO] - [Link Router 1 NET 0] ARP ip 1.0.0.2 is in cache.
[INFO] - [Link Host 2 NET 0] ARP_QUERY
[INFO] - [Link Host 2 NET 0] ARP ip 1.0.0.0 not cached.
[INFO] - [Link Host 2 NET 0] Sending ARP query for ip: 1.0.0.0
[INFO] - [Link Host 1 NET 0] Process ARP packet.
[INFO] - [Link Router 1 NET 0] Process ARP packet.
[INFO] - [Link Host 2 NET 0] Process ARP packet.
[INFO] - [Link Host 2 NET 0] Send control ARP_READY
[INFO] - [Link Host 2 NET 0] SEND_PACKET
[INFO] - [Link Host 2 NET 0] ARP ip 1.0.0.0 is in cache.
[INFO] - [Link Router 1 NET 2] ARP_QUERY
[INFO] - [Link Router 1 NET 2] ARP ip 2.0.0.1 not cached.
[INFO] - [Link Router 1 NET 2] Sending ARP query for ip: 2.0.0.1
[INFO] - [Link Router 3 NET 0] Process ARP packet.
[INFO] - [Link Router 1 NET 2] Process ARP packet.
[INFO] - [Link Router 1 NET 2] Send control ARP_READY
[INFO] - [Link Router 1 NET 2] SEND_PACKET
[INFO] - [Link Router 1 NET 2] ARP ip 2.0.0.1 is in cache.
[INFO] - [Link Router 3 NET 1] ARP_QUERY
[INFO] - [Link Router 3 NET 1] ARP ip 4.0.0.1 not cached.
[INFO] - [Link Router 3 NET 1] Sending ARP query for ip: 4.0.0.1
[INFO] - [Link Host 3 NET 0] Process ARP packet.
[INFO] - [Link Router 3 NET 1] Process ARP packet.
[INFO] - [Link Router 3 NET 1] Send control ARP_READY
[INFO] - [Link Router 3 NET 1] SEND_PACKET
[INFO] - [Link Router 3 NET 1] ARP ip 4.0.0.1 is in cache.
[INFO] - [Link Host 3 NET 0] ARP_QUERY

```

```

[INFO] - [Link Host 3 NET 0] ARP ip 4.0.0.0 is in cache.
[INFO] - [Link Host 3 NET 0] Send control ARP_READY
[INFO] - [Link Host 3 NET 0] SEND_PACKET
[INFO] - [Link Host 3 NET 0] ARP ip 4.0.0.0 is in cache.
[INFO] - [Link Router 3 NET 0] ARP_QUERY
[INFO] - [Link Router 3 NET 0] ARP ip 2.0.0.0 is in cache.
[INFO] - [Link Router 3 NET 0] Send control ARP_READY
[INFO] - [Link Router 3 NET 0] SEND_PACKET
[INFO] - [Link Router 3 NET 0] ARP ip 2.0.0.0 is in cache.
[INFO] - [Link Router 1 NET 0] ARP_QUERY
[INFO] - [Link Router 1 NET 0] ARP ip 1.0.0.2 not cached.
[INFO] - [Link Router 1 NET 0] Sending ARP query for ip: 1.0.0.2
[INFO] - [Link Host 2 NET 0] Process ARP packet.
[INFO] - [Link Host 1 NET 0] Process ARP packet.
[INFO] - [Link Router 1 NET 0] Process ARP packet.
[INFO] - [Link Router 1 NET 0] Send control ARP_READY
[INFO] - [Link Router 1 NET 0] SEND_PACKET
[INFO] - [Link Router 1 NET 0] ARP ip 1.0.0.2 is in cache.
[INFO] - [Link Host 2 NET 0] ARP_QUERY
[INFO] - [Link Host 2 NET 0] ARP ip 1.0.0.1 not cached.
[INFO] - [Link Host 2 NET 0] Sending ARP query for ip: 1.0.0.1
[INFO] - [Link Host 1 NET 0] Process ARP packet.
[INFO] - [Link Router 1 NET 0] Process ARP packet.
[INFO] - [Link Host 2 NET 0] Process ARP packet.
[INFO] - [Link Host 2 NET 0] Send control ARP_READY
[INFO] - [Link Host 2 NET 0] SEND_PACKET
[INFO] - [Link Host 2 NET 0] ARP ip 1.0.0.1 is in cache.

```

Output del módulo SWITCH generado por el logger durante la simulación

```

[INFO] - [Switch Switch 1] MAC: 0:0:0:0:0:13
[INFO] - [Switch Switch 1] Interface amounts: 3
[INFO] - [Switch Switch 1] Added swp protocol for port 0
[INFO] - [Switch Switch 1] Added swp protocol for port 1
[INFO] - [Switch Switch 1] Added swp protocol for port 2
[INFO] - [Switch Switch 1] Forwarding table:
[INFO] - [Switch Switch 1] 0:0:0:0:0:1 | 0
[INFO] - [Switch Switch 1] 0:0:0:0:0:2 | 2
[INFO] - [Switch Switch 1] 0:0:0:0:0:3 | 2
[INFO] - [Switch Switch 2] MAC: 0:0:0:0:0:14
[INFO] - [Switch Switch 2] Interface amounts: 3
[INFO] - [Switch Switch 2] Added swp protocol for port 0
[INFO] - [Switch Switch 2] Added swp protocol for port 1
[INFO] - [Switch Switch 2] Added swp protocol for port 2
[INFO] - [Switch Switch 2] Forwarding table:
[INFO] - [Switch Switch 2] 0:0:0:0:0:1 | 1
[INFO] - [Switch Switch 2] 0:0:0:0:0:2 | 0
[INFO] - [Switch Switch 2] 0:0:0:0:0:3 | 2
[INFO] - [Switch Switch 3] MAC: 0:0:0:0:0:15
[INFO] - [Switch Switch 3] Interface amounts: 3
[INFO] - [Switch Switch 3] Added swp protocol for port 0
[INFO] - [Switch Switch 3] Added swp protocol for port 1
[INFO] - [Switch Switch 3] Added swp protocol for port 2
[INFO] - [Switch Switch 3] Forwarding table:
[INFO] - [Switch Switch 3] 0:0:0:0:0:1 | 0
[INFO] - [Switch Switch 3] 0:0:0:0:0:2 | 0
[INFO] - [Switch Switch 3] 0:0:0:0:0:3 | 2
[INFO] - [Switch Switch 4] MAC: 0:0:0:0:0:16
[INFO] - [Switch Switch 4] Interface amounts: 3
[INFO] - [Switch Switch 4] Added swp protocol for port 0
[INFO] - [Switch Switch 4] Added swp protocol for port 1
[INFO] - [Switch Switch 4] Added swp protocol for port 2
[INFO] - [Switch Switch 4] Forwarding table:
[INFO] - [Switch Switch 4] 0:0:0:0:0:10 | 0
[INFO] - [Switch Switch 4] 0:0:0:0:0:11 | 1
[INFO] - [Switch Switch 4] 0:0:0:0:0:12 | 2
Simulation Initialized
[INFO] - [Switch Switch 1] Broadcast MAC sent throw all interfaces

```


Output del módulo SWP generado por el logger durante la simulación

```
[INFO] - [SWP Switch 1 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 1 Port: 0] Deliver frame with SeqNum: 1
[INFO] - [SWP Switch 1 Port: 0] send ACK LFR: 1
[INFO] - [SWP Host 1 NET 0] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 1] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 1] Deliver frame with SeqNum: 1
[INFO] - [SWP Switch 2 Port: 1] send ACK LFR: 1
[INFO] - [SWP Host 2 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 2 NET 0] Deliver frame with SeqNum: 1
[INFO] - [SWP Host 2 NET 0] send ACK LFR: 1
[INFO] - [SWP Switch 3 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 3 Port: 0] Deliver frame with SeqNum: 1
[INFO] - [SWP Switch 3 Port: 0] send ACK LFR: 1
[INFO] - [SWP Router 1 NET 0] swp_protocol::processFrame
[INFO] - [SWP Router 1 NET 0] Deliver frame with SeqNum: 1
[INFO] - [SWP Router 1 NET 0] send ACK LFR: 1
[INFO] - [SWP Switch 1 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Host 2 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 2 NET 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 2 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 0] Deliver frame with SeqNum: 1
[INFO] - [SWP Switch 2 Port: 0] send ACK LFR: 1
```

```

[INFO] - [SWP Host 2 NET 0] swp_protocol::processFrame
[INFO] - [SWP Switch 3 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Router 1 NET 0] swp_protocol::processFrame
[INFO] - [SWP Router 1 NET 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 3 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 3 Port: 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 2 Port: 1] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 1] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 1 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 1 Port: 2] Deliver frame with SeqNum: 1
[INFO] - [SWP Switch 1 Port: 2] send ACK LFR: 1
[INFO] - [SWP Switch 2 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Host 1 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 1 NET 0] Deliver frame with SeqNum: 1
[INFO] - [SWP Host 1 NET 0] send ACK LFR: 1
[INFO] - [SWP Switch 2 Port: 1] swp_protocol::processFrame
[INFO] - [SWP Switch 1 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Host 1 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 1 NET 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 1 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 1 Port: 0] Deliver frame with SeqNum: 2
[INFO] - [SWP Switch 1 Port: 0] send ACK LFR: 2
[INFO] - [SWP Host 1 NET 0] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 1] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 1] Deliver frame with SeqNum: 2
[INFO] - [SWP Switch 2 Port: 1] send ACK LFR: 2
[INFO] - [SWP Host 2 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 2 NET 0] Deliver frame with SeqNum: 2
[INFO] - [SWP Host 2 NET 0] send ACK LFR: 2
[INFO] - [SWP Switch 1 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Host 2 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 2 NET 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 2 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 0] Deliver frame with SeqNum: 2
[INFO] - [SWP Switch 2 Port: 0] send ACK LFR: 2
[INFO] - [SWP Host 2 NET 0] swp_protocol::processFrame
[INFO] - [SWP Switch 1 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 1 Port: 2] Deliver frame with SeqNum: 2
[INFO] - [SWP Switch 1 Port: 2] send ACK LFR: 2
[INFO] - [SWP Switch 2 Port: 1] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 1] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 3 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 3 Port: 0] Deliver frame with SeqNum: 2
[INFO] - [SWP Switch 3 Port: 0] send ACK LFR: 2
[INFO] - [SWP Switch 2 Port: 1] swp_protocol::processFrame
[INFO] - [SWP Host 1 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 1 NET 0] Deliver frame with SeqNum: 2
[INFO] - [SWP Host 1 NET 0] send ACK LFR: 2
[INFO] - [SWP Router 1 NET 0] swp_protocol::processFrame
[INFO] - [SWP Router 1 NET 0] Deliver frame with SeqNum: 2
[INFO] - [SWP Router 1 NET 0] send ACK LFR: 2
[INFO] - [SWP Switch 2 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 1 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Host 1 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 1 NET 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 3 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Router 1 NET 0] swp_protocol::processFrame
[INFO] - [SWP Router 1 NET 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 3 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 3 Port: 2] Deliver frame with SeqNum: 1
[INFO] - [SWP Switch 3 Port: 2] send ACK LFR: 1
[INFO] - [SWP Router 1 NET 0] swp_protocol::processFrame
[INFO] - [SWP Switch 3 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 3 Port: 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 2 Port: 2] swp_protocol::processFrame

```

```

[INFO] - [SWP Switch 2 Port: 2] Deliver frame with SeqNum: 1
[INFO] - [SWP Switch 2 Port: 2] send ACK LFR: 1
[INFO] - [SWP Switch 3 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 3 Port: 2] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Host 2 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 2 NET 0] Deliver frame with SeqNum: 3
[INFO] - [SWP Host 2 NET 0] send ACK LFR: 3
[INFO] - [SWP Switch 3 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Host 2 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 2 NET 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 2 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 0] Deliver frame with SeqNum: 3
[INFO] - [SWP Switch 2 Port: 0] send ACK LFR: 3
[INFO] - [SWP Host 2 NET 0] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 2] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 3 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 3 Port: 0] Deliver frame with SeqNum: 3
[INFO] - [SWP Switch 3 Port: 0] send ACK LFR: 3
[INFO] - [SWP Router 1 NET 0] swp_protocol::processFrame
[INFO] - [SWP Router 1 NET 0] Deliver frame with SeqNum: 3
[INFO] - [SWP Router 1 NET 0] send ACK LFR: 3
[INFO] - [SWP Router 2 NET 0] swp_protocol::processFrame
[INFO] - [SWP Router 2 NET 0] Deliver frame with SeqNum: 1
[INFO] - [SWP Router 2 NET 0] send ACK LFR: 1
[INFO] - [SWP Router 1 NET 1] swp_protocol::processFrame
[INFO] - [SWP Router 1 NET 1] swp_protocol::processFrame
[INFO] - [SWP Router 1 NET 1] Deliver frame with SeqNum: 1
[INFO] - [SWP Router 1 NET 1] send ACK LFR: 1
[INFO] - [SWP Router 2 NET 0] swp_protocol::processFrame
[INFO] - [SWP Router 2 NET 0] swp_protocol::processFrame
[INFO] - [SWP Router 2 NET 0] Deliver frame with SeqNum: 2
[INFO] - [SWP Router 2 NET 0] send ACK LFR: 2
[INFO] - [SWP Router 1 NET 1] swp_protocol::processFrame
[INFO] - [SWP Switch 4 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 4 Port: 0] Deliver frame with SeqNum: 1
[INFO] - [SWP Switch 4 Port: 0] send ACK LFR: 1
[INFO] - [SWP Router 2 NET 1] swp_protocol::processFrame
[INFO] - [SWP Host 5 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 5 NET 0] Deliver frame with SeqNum: 1
[INFO] - [SWP Host 5 NET 0] send ACK LFR: 1
[INFO] - [SWP Switch 4 Port: 1] swp_protocol::processFrame
[INFO] - [SWP Host 4 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 4 NET 0] Deliver frame with SeqNum: 1
[INFO] - [SWP Host 4 NET 0] send ACK LFR: 1
[INFO] - [SWP Switch 2 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 3 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Router 1 NET 0] swp_protocol::processFrame
[INFO] - [SWP Router 1 NET 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 3 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 3 Port: 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 4 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Host 4 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 4 NET 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 4 Port: 1] swp_protocol::processFrame
[INFO] - [SWP Switch 4 Port: 1] Deliver frame with SeqNum: 1
[INFO] - [SWP Switch 4 Port: 1] send ACK LFR: 1
[INFO] - [SWP Host 5 NET 0] swp_protocol::processFrame
[INFO] - [SWP Switch 4 Port: 1] swp_protocol::processFrame
[INFO] - [SWP Switch 4 Port: 1] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Router 2 NET 1] swp_protocol::processFrame
[INFO] - [SWP Router 2 NET 1] Deliver frame with SeqNum: 1
[INFO] - [SWP Router 2 NET 1] send ACK LFR: 1
[INFO] - [SWP Switch 4 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Router 2 NET 1] swp_protocol::processFrame
[INFO] - [SWP Router 2 NET 1] Discarded frame: Frame not in RWS windows

```

```

[INFO] - [SWP Switch 4 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 4 Port: 0] Deliver frame with SeqNum: 2
[INFO] - [SWP Switch 4 Port: 0] send ACK LFR: 2
[INFO] - [SWP Router 2 NET 1] swp_protocol::processFrame
[INFO] - [SWP Host 5 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 5 NET 0] Deliver frame with SeqNum: 2
[INFO] - [SWP Host 5 NET 0] send ACK LFR: 2
[INFO] - [SWP Switch 4 Port: 1] swp_protocol::processFrame
[INFO] - [SWP Host 5 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 5 NET 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 4 Port: 1] swp_protocol::processFrame
[INFO] - [SWP Switch 4 Port: 1] Deliver frame with SeqNum: 2
[INFO] - [SWP Switch 4 Port: 1] send ACK LFR: 2
[INFO] - [SWP Host 5 NET 0] swp_protocol::processFrame
[INFO] - [SWP Router 2 NET 1] swp_protocol::processFrame
[INFO] - [SWP Router 2 NET 1] Deliver frame with SeqNum: 2
[INFO] - [SWP Router 2 NET 1] send ACK LFR: 2
[INFO] - [SWP Switch 4 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Host 4 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 4 NET 0] Deliver frame with SeqNum: 2
[INFO] - [SWP Host 4 NET 0] send ACK LFR: 2
[INFO] - [SWP Switch 4 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Host 4 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 4 NET 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 4 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 4 Port: 0] Deliver frame with SeqNum: 3
[INFO] - [SWP Switch 4 Port: 0] send ACK LFR: 3
[INFO] - [SWP Router 2 NET 1] swp_protocol::processFrame
[INFO] - [SWP Switch 4 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 4 Port: 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Host 5 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 5 NET 0] Deliver frame with SeqNum: 3
[INFO] - [SWP Host 5 NET 0] send ACK LFR: 3
[INFO] - [SWP Switch 4 Port: 1] swp_protocol::processFrame
[INFO] - [SWP Host 5 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 5 NET 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 4 Port: 1] swp_protocol::processFrame
[INFO] - [SWP Switch 4 Port: 1] Deliver frame with SeqNum: 3
[INFO] - [SWP Switch 4 Port: 1] send ACK LFR: 3
[INFO] - [SWP Host 5 NET 0] swp_protocol::processFrame
[INFO] - [SWP Router 2 NET 1] swp_protocol::processFrame
[INFO] - [SWP Router 2 NET 1] Deliver frame with SeqNum: 3
[INFO] - [SWP Router 2 NET 1] send ACK LFR: 3
[INFO] - [SWP Router 1 NET 1] swp_protocol::processFrame
[INFO] - [SWP Router 1 NET 1] Deliver frame with SeqNum: 2
[INFO] - [SWP Router 1 NET 1] send ACK LFR: 2
[INFO] - [SWP Router 2 NET 0] swp_protocol::processFrame
[INFO] - [SWP Router 2 NET 0] swp_protocol::processFrame
[INFO] - [SWP Router 2 NET 0] Deliver frame with SeqNum: 3
[INFO] - [SWP Router 2 NET 0] send ACK LFR: 3
[INFO] - [SWP Router 1 NET 1] swp_protocol::processFrame
[INFO] - [SWP Router 1 NET 1] swp_protocol::processFrame
[INFO] - [SWP Router 1 NET 1] Deliver frame with SeqNum: 3
[INFO] - [SWP Router 1 NET 1] send ACK LFR: 3
[INFO] - [SWP Router 2 NET 0] swp_protocol::processFrame
[INFO] - [SWP Switch 3 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 3 Port: 2] Deliver frame with SeqNum: 2
[INFO] - [SWP Switch 3 Port: 2] send ACK LFR: 2
[INFO] - [SWP Router 1 NET 0] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 2] Deliver frame with SeqNum: 2
[INFO] - [SWP Switch 2 Port: 2] send ACK LFR: 2
[INFO] - [SWP Host 2 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 2 NET 0] Deliver frame with SeqNum: 4
[INFO] - [SWP Host 2 NET 0] send ACK LFR: 4
[INFO] - [SWP Switch 1 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 1 Port: 2] Deliver frame with SeqNum: 3

```

```

[INFO] - [SWP Switch 1 Port: 2] send ACK LFR: 3
[INFO] - [SWP Host 1 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 1 NET 0] Deliver frame with SeqNum: 3
[INFO] - [SWP Host 1 NET 0] send ACK LFR: 3
[INFO] - [SWP Switch 4 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Router 2 NET 1] swp_protocol::processFrame
[INFO] - [SWP Router 2 NET 1] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 3 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Host 2 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 2 NET 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 2 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 1] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 0] Deliver frame with SeqNum: 4
[INFO] - [SWP Switch 2 Port: 0] send ACK LFR: 4
[INFO] - [SWP Host 2 NET 0] swp_protocol::processFrame
[INFO] - [SWP Switch 1 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Host 1 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 1 NET 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 1 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 1 Port: 2] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 2 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 2] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 3 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 3 Port: 0] Deliver frame with SeqNum: 4
[INFO] - [SWP Switch 3 Port: 0] send ACK LFR: 4
[INFO] - [SWP Switch 2 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Router 1 NET 0] swp_protocol::processFrame
[INFO] - [SWP Router 1 NET 0] Deliver frame with SeqNum: 4
[INFO] - [SWP Router 1 NET 0] send ACK LFR: 4
[INFO] - [SWP Switch 2 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 3 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Router 1 NET 0] swp_protocol::processFrame
[INFO] - [SWP Router 1 NET 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 3 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 3 Port: 2] Deliver frame with SeqNum: 3
[INFO] - [SWP Switch 3 Port: 2] send ACK LFR: 3
[INFO] - [SWP Router 1 NET 0] swp_protocol::processFrame
[INFO] - [SWP Switch 3 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 3 Port: 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 2 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 2] Deliver frame with SeqNum: 3
[INFO] - [SWP Switch 2 Port: 2] send ACK LFR: 3
[INFO] - [SWP Host 2 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 2 NET 0] Deliver frame with SeqNum: 5
[INFO] - [SWP Host 2 NET 0] send ACK LFR: 5
[INFO] - [SWP Switch 3 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Host 2 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 2 NET 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 2 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 0] Deliver frame with SeqNum: 5
[INFO] - [SWP Switch 2 Port: 0] send ACK LFR: 5
[INFO] - [SWP Host 2 NET 0] swp_protocol::processFrame
[INFO] - [SWP Switch 1 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 1 Port: 2] Deliver frame with SeqNum: 4
[INFO] - [SWP Switch 1 Port: 2] send ACK LFR: 4
[INFO] - [SWP Switch 2 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 2] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 3 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 3 Port: 0] Deliver frame with SeqNum: 5
[INFO] - [SWP Switch 3 Port: 0] send ACK LFR: 5
[INFO] - [SWP Switch 2 Port: 1] swp_protocol::processFrame
[INFO] - [SWP Host 1 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 1 NET 0] Deliver frame with SeqNum: 4
[INFO] - [SWP Host 1 NET 0] send ACK LFR: 4

```

```

[INFO] - [SWP Router 1 NET 0] swp-protocol::processFrame
[INFO] - [SWP Router 1 NET 0] Deliver frame with SeqNum: 5
[INFO] - [SWP Router 1 NET 0] send ACK LFR: 5
[INFO] - [SWP Switch 2 Port: 2] swp-protocol::processFrame
[INFO] - [SWP Switch 1 Port: 0] swp-protocol::processFrame
[INFO] - [SWP Host 1 NET 0] swp-protocol::processFrame
[INFO] - [SWP Host 1 NET 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 3 Port: 2] swp-protocol::processFrame
[INFO] - [SWP Router 1 NET 0] swp-protocol::processFrame
[INFO] - [SWP Router 1 NET 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 3 Port: 2] swp-protocol::processFrame
[INFO] - [SWP Switch 3 Port: 2] Deliver frame with SeqNum: 4
[INFO] - [SWP Switch 3 Port: 2] send ACK LFR: 4
[INFO] - [SWP Router 1 NET 0] swp-protocol::processFrame
[INFO] - [SWP Switch 3 Port: 0] swp-protocol::processFrame
[INFO] - [SWP Switch 3 Port: 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 2 Port: 2] swp-protocol::processFrame
[INFO] - [SWP Switch 2 Port: 2] Deliver frame with SeqNum: 4
[INFO] - [SWP Switch 2 Port: 2] send ACK LFR: 4
[INFO] - [SWP Switch 3 Port: 2] swp-protocol::processFrame
[INFO] - [SWP Switch 3 Port: 2] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Host 2 NET 0] swp-protocol::processFrame
[INFO] - [SWP Host 2 NET 0] Deliver frame with SeqNum: 6
[INFO] - [SWP Host 2 NET 0] send ACK LFR: 6
[INFO] - [SWP Switch 3 Port: 0] swp-protocol::processFrame
[INFO] - [SWP Switch 2 Port: 0] swp-protocol::processFrame
[INFO] - [SWP Host 2 NET 0] swp-protocol::processFrame
[INFO] - [SWP Host 2 NET 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 2 Port: 0] swp-protocol::processFrame
[INFO] - [SWP Switch 2 Port: 0] Deliver frame with SeqNum: 6
[INFO] - [SWP Switch 2 Port: 0] send ACK LFR: 6
[INFO] - [SWP Host 2 NET 0] swp-protocol::processFrame
[INFO] - [SWP Switch 2 Port: 2] swp-protocol::processFrame
[INFO] - [SWP Switch 2 Port: 2] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 3 Port: 0] swp-protocol::processFrame
[INFO] - [SWP Switch 3 Port: 0] Deliver frame with SeqNum: 6
[INFO] - [SWP Switch 3 Port: 0] send ACK LFR: 6
[INFO] - [SWP Router 1 NET 0] swp-protocol::processFrame
[INFO] - [SWP Router 1 NET 0] Deliver frame with SeqNum: 6
[INFO] - [SWP Router 1 NET 0] send ACK LFR: 6
[INFO] - [SWP Router 2 NET 0] swp-protocol::processFrame
[INFO] - [SWP Router 2 NET 0] Deliver frame with SeqNum: 4
[INFO] - [SWP Router 2 NET 0] send ACK LFR: 4
[INFO] - [SWP Router 1 NET 1] swp-protocol::processFrame
[INFO] - [SWP Router 1 NET 1] swp-protocol::processFrame
[INFO] - [SWP Router 1 NET 1] Deliver frame with SeqNum: 4
[INFO] - [SWP Router 1 NET 1] send ACK LFR: 4
[INFO] - [SWP Router 2 NET 0] swp-protocol::processFrame
[INFO] - [SWP Router 2 NET 0] swp-protocol::processFrame
[INFO] - [SWP Router 2 NET 0] Deliver frame with SeqNum: 5
[INFO] - [SWP Router 2 NET 0] send ACK LFR: 5
[INFO] - [SWP Router 1 NET 1] swp-protocol::processFrame
[INFO] - [SWP Switch 4 Port: 0] swp-protocol::processFrame
[INFO] - [SWP Switch 4 Port: 0] Deliver frame with SeqNum: 4
[INFO] - [SWP Switch 4 Port: 0] send ACK LFR: 4
[INFO] - [SWP Router 2 NET 1] swp-protocol::processFrame
[INFO] - [SWP Host 5 NET 0] swp-protocol::processFrame
[INFO] - [SWP Host 5 NET 0] Deliver frame with SeqNum: 4
[INFO] - [SWP Host 5 NET 0] send ACK LFR: 4
[INFO] - [SWP Switch 4 Port: 1] swp-protocol::processFrame
[INFO] - [SWP Host 4 NET 0] swp-protocol::processFrame
[INFO] - [SWP Host 4 NET 0] Deliver frame with SeqNum: 3
[INFO] - [SWP Host 4 NET 0] send ACK LFR: 3
[INFO] - [SWP Switch 2 Port: 2] swp-protocol::processFrame
[INFO] - [SWP Switch 3 Port: 2] swp-protocol::processFrame
[INFO] - [SWP Router 1 NET 0] swp-protocol::processFrame
[INFO] - [SWP Router 1 NET 0] Discarded frame: Frame not in RWS windows

```

```

[INFO] - [SWP Switch 3 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 3 Port: 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 4 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Host 4 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 4 NET 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 4 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 4 Port: 2] Deliver frame with SeqNum: 1
[INFO] - [SWP Switch 4 Port: 2] send ACK LFR: 1
[INFO] - [SWP Host 4 NET 0] swp_protocol::processFrame
[INFO] - [SWP Switch 4 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 4 Port: 2] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Router 2 NET 1] swp_protocol::processFrame
[INFO] - [SWP Router 2 NET 1] Deliver frame with SeqNum: 4
[INFO] - [SWP Router 2 NET 1] send ACK LFR: 4
[INFO] - [SWP Switch 4 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Router 2 NET 1] swp_protocol::processFrame
[INFO] - [SWP Router 2 NET 1] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 4 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 4 Port: 0] Deliver frame with SeqNum: 5
[INFO] - [SWP Switch 4 Port: 0] send ACK LFR: 5
[INFO] - [SWP Router 2 NET 1] swp_protocol::processFrame
[INFO] - [SWP Host 4 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 4 NET 0] Deliver frame with SeqNum: 4
[INFO] - [SWP Host 4 NET 0] send ACK LFR: 4
[INFO] - [SWP Switch 4 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Host 4 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 4 NET 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 4 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 4 Port: 2] Deliver frame with SeqNum: 2
[INFO] - [SWP Switch 4 Port: 2] send ACK LFR: 2
[INFO] - [SWP Host 4 NET 0] swp_protocol::processFrame
[INFO] - [SWP Router 2 NET 1] swp_protocol::processFrame
[INFO] - [SWP Router 2 NET 1] Deliver frame with SeqNum: 5
[INFO] - [SWP Router 2 NET 1] send ACK LFR: 5
[INFO] - [SWP Switch 4 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Host 5 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 5 NET 0] Deliver frame with SeqNum: 5
[INFO] - [SWP Host 5 NET 0] send ACK LFR: 5
[INFO] - [SWP Switch 4 Port: 1] swp_protocol::processFrame
[INFO] - [SWP Host 5 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 5 NET 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 4 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 4 Port: 0] Deliver frame with SeqNum: 6
[INFO] - [SWP Switch 4 Port: 0] send ACK LFR: 6
[INFO] - [SWP Router 2 NET 1] swp_protocol::processFrame
[INFO] - [SWP Switch 4 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 4 Port: 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Host 4 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 4 NET 0] Deliver frame with SeqNum: 5
[INFO] - [SWP Host 4 NET 0] send ACK LFR: 5
[INFO] - [SWP Switch 4 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Host 4 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 4 NET 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 4 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 4 Port: 2] Deliver frame with SeqNum: 3
[INFO] - [SWP Switch 4 Port: 2] send ACK LFR: 3
[INFO] - [SWP Host 4 NET 0] swp_protocol::processFrame
[INFO] - [SWP Router 2 NET 1] swp_protocol::processFrame
[INFO] - [SWP Router 2 NET 1] Deliver frame with SeqNum: 6
[INFO] - [SWP Router 2 NET 1] send ACK LFR: 6
[INFO] - [SWP Router 1 NET 1] swp_protocol::processFrame
[INFO] - [SWP Router 1 NET 1] Deliver frame with SeqNum: 5
[INFO] - [SWP Router 1 NET 1] send ACK LFR: 5
[INFO] - [SWP Router 2 NET 0] swp_protocol::processFrame
[INFO] - [SWP Router 2 NET 0] swp_protocol::processFrame
[INFO] - [SWP Router 2 NET 0] Deliver frame with SeqNum: 6
[INFO] - [SWP Router 2 NET 0] send ACK LFR: 6

```

```

[INFO] - [SWP Router 1 NET 1] swp_protocol::processFrame
[INFO] - [SWP Router 1 NET 1] swp_protocol::processFrame
[INFO] - [SWP Router 1 NET 1] Deliver frame with SeqNum: 6
[INFO] - [SWP Router 1 NET 1] send ACK LFR: 6
[INFO] - [SWP Router 2 NET 0] swp_protocol::processFrame
[INFO] - [SWP Switch 3 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 3 Port: 2] Deliver frame with SeqNum: 5
[INFO] - [SWP Switch 3 Port: 2] send ACK LFR: 5
[INFO] - [SWP Router 1 NET 0] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 2] Deliver frame with SeqNum: 5
[INFO] - [SWP Switch 2 Port: 2] send ACK LFR: 5
[INFO] - [SWP Host 2 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 2 NET 0] Deliver frame with SeqNum: 7
[INFO] - [SWP Host 2 NET 0] send ACK LFR: 7
[INFO] - [SWP Switch 1 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 1 Port: 2] Deliver frame with SeqNum: 5
[INFO] - [SWP Switch 1 Port: 2] send ACK LFR: 5
[INFO] - [SWP Host 1 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 1 NET 0] Deliver frame with SeqNum: 5
[INFO] - [SWP Host 1 NET 0] send ACK LFR: 5
[INFO] - [SWP Switch 4 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Router 2 NET 1] swp_protocol::processFrame
[INFO] - [SWP Router 2 NET 1] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 3 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Host 2 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 2 NET 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 2 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 1] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 0] Deliver frame with SeqNum: 7
[INFO] - [SWP Switch 2 Port: 0] send ACK LFR: 7
[INFO] - [SWP Host 2 NET 0] swp_protocol::processFrame
[INFO] - [SWP Switch 1 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Host 1 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 1 NET 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 1 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 1 Port: 2] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 2 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 2] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 3 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 3 Port: 0] Deliver frame with SeqNum: 7
[INFO] - [SWP Switch 3 Port: 0] send ACK LFR: 7
[INFO] - [SWP Switch 2 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Router 1 NET 0] swp_protocol::processFrame
[INFO] - [SWP Router 1 NET 0] Deliver frame with SeqNum: 7
[INFO] - [SWP Router 1 NET 0] send ACK LFR: 7
[INFO] - [SWP Switch 2 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 3 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Router 1 NET 0] swp_protocol::processFrame
[INFO] - [SWP Router 1 NET 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 3 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 3 Port: 2] Deliver frame with SeqNum: 6
[INFO] - [SWP Switch 3 Port: 2] send ACK LFR: 6
[INFO] - [SWP Router 1 NET 0] swp_protocol::processFrame
[INFO] - [SWP Switch 3 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 3 Port: 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 2 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 2] Deliver frame with SeqNum: 6
[INFO] - [SWP Switch 2 Port: 2] send ACK LFR: 6
[INFO] - [SWP Host 2 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 2 NET 0] Deliver frame with SeqNum: 8
[INFO] - [SWP Host 2 NET 0] send ACK LFR: 8
[INFO] - [SWP Switch 3 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Host 2 NET 0] swp_protocol::processFrame

```

```

[INFO] - [SWP Host 2 NET 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 2 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 0] Deliver frame with SeqNum: 8
[INFO] - [SWP Switch 2 Port: 0] send ACK LFR: 8
[INFO] - [SWP Host 2 NET 0] swp_protocol::processFrame
[INFO] - [SWP Switch 1 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 1 Port: 2] Deliver frame with SeqNum: 6
[INFO] - [SWP Switch 1 Port: 2] send ACK LFR: 6
[INFO] - [SWP Switch 2 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 2] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 3 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 3 Port: 0] Deliver frame with SeqNum: 8
[INFO] - [SWP Switch 3 Port: 0] send ACK LFR: 8
[INFO] - [SWP Switch 2 Port: 1] swp_protocol::processFrame
[INFO] - [SWP Host 1 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 1 NET 0] Deliver frame with SeqNum: 6
[INFO] - [SWP Host 1 NET 0] send ACK LFR: 6
[INFO] - [SWP Router 1 NET 0] swp_protocol::processFrame
[INFO] - [SWP Router 1 NET 0] Deliver frame with SeqNum: 8
[INFO] - [SWP Router 1 NET 0] send ACK LFR: 8
[INFO] - [SWP Switch 2 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 1 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Host 1 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 1 NET 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 3 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Router 1 NET 0] swp_protocol::processFrame
[INFO] - [SWP Router 1 NET 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 3 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 3 Port: 2] Deliver frame with SeqNum: 7
[INFO] - [SWP Switch 3 Port: 2] send ACK LFR: 7
[INFO] - [SWP Router 1 NET 0] swp_protocol::processFrame
[INFO] - [SWP Switch 3 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 3 Port: 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 2 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 2] Deliver frame with SeqNum: 7
[INFO] - [SWP Switch 2 Port: 2] send ACK LFR: 7
[INFO] - [SWP Switch 3 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 3 Port: 2] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Host 2 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 2 NET 0] Deliver frame with SeqNum: 9
[INFO] - [SWP Host 2 NET 0] send ACK LFR: 9
[INFO] - [SWP Switch 3 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Host 2 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 2 NET 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 2 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 0] Deliver frame with SeqNum: 9
[INFO] - [SWP Switch 2 Port: 0] send ACK LFR: 9
[INFO] - [SWP Host 2 NET 0] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 2] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 3 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 3 Port: 0] Deliver frame with SeqNum: 9
[INFO] - [SWP Switch 3 Port: 0] send ACK LFR: 9
[INFO] - [SWP Router 1 NET 0] swp_protocol::processFrame
[INFO] - [SWP Router 1 NET 0] Deliver frame with SeqNum: 9
[INFO] - [SWP Router 1 NET 0] send ACK LFR: 9
[INFO] - [SWP Router 3 NET 0] swp_protocol::processFrame
[INFO] - [SWP Router 3 NET 0] Deliver frame with SeqNum: 1
[INFO] - [SWP Router 3 NET 0] send ACK LFR: 1
[INFO] - [SWP Router 1 NET 2] swp_protocol::processFrame
[INFO] - [SWP Router 1 NET 2] swp_protocol::processFrame
[INFO] - [SWP Router 1 NET 2] Deliver frame with SeqNum: 1
[INFO] - [SWP Router 1 NET 2] send ACK LFR: 1
[INFO] - [SWP Router 3 NET 0] swp_protocol::processFrame
[INFO] - [SWP Router 3 NET 0] swp_protocol::processFrame
[INFO] - [SWP Router 3 NET 0] Deliver frame with SeqNum: 2

```

```

[INFO] - [SWP Router 3 NET 0] send ACK LFR: 2
[INFO] - [SWP Router 1 NET 2] swp_protocol::processFrame
[INFO] - [SWP Host 3 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 3 NET 0] Deliver frame with SeqNum: 1
[INFO] - [SWP Host 3 NET 0] send ACK LFR: 1
[INFO] - [SWP Router 3 NET 1] swp_protocol::processFrame
[INFO] - [SWP Router 3 NET 1] swp_protocol::processFrame
[INFO] - [SWP Router 3 NET 1] Deliver frame with SeqNum: 1
[INFO] - [SWP Router 3 NET 1] send ACK LFR: 1
[INFO] - [SWP Host 3 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 3 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 3 NET 0] Deliver frame with SeqNum: 2
[INFO] - [SWP Host 3 NET 0] send ACK LFR: 2
[INFO] - [SWP Router 3 NET 1] swp_protocol::processFrame
[INFO] - [SWP Router 3 NET 1] swp_protocol::processFrame
[INFO] - [SWP Router 3 NET 1] Deliver frame with SeqNum: 2
[INFO] - [SWP Router 3 NET 1] send ACK LFR: 2
[INFO] - [SWP Host 3 NET 0] swp_protocol::processFrame
[INFO] - [SWP Router 1 NET 2] swp_protocol::processFrame
[INFO] - [SWP Router 1 NET 2] Deliver frame with SeqNum: 2
[INFO] - [SWP Router 1 NET 2] send ACK LFR: 2
[INFO] - [SWP Router 3 NET 0] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 3 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Router 1 NET 0] swp_protocol::processFrame
[INFO] - [SWP Router 1 NET 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 3 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 3 Port: 2] Deliver frame with SeqNum: 8
[INFO] - [SWP Switch 3 Port: 2] send ACK LFR: 8
[INFO] - [SWP Router 1 NET 0] swp_protocol::processFrame
[INFO] - [SWP Switch 3 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 3 Port: 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 2 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 2] Deliver frame with SeqNum: 8
[INFO] - [SWP Switch 2 Port: 2] send ACK LFR: 8
[INFO] - [SWP Host 2 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 2 NET 0] Deliver frame with SeqNum: 10
[INFO] - [SWP Host 2 NET 0] send ACK LFR: 10
[INFO] - [SWP Switch 1 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 1 Port: 2] Deliver frame with SeqNum: 7
[INFO] - [SWP Switch 1 Port: 2] send ACK LFR: 7
[INFO] - [SWP Host 1 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 1 NET 0] Deliver frame with SeqNum: 7
[INFO] - [SWP Host 1 NET 0] send ACK LFR: 7
[INFO] - [SWP Switch 3 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Host 2 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 2 NET 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 2 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 1] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 0] Deliver frame with SeqNum: 10
[INFO] - [SWP Switch 2 Port: 0] send ACK LFR: 10
[INFO] - [SWP Host 2 NET 0] swp_protocol::processFrame
[INFO] - [SWP Switch 1 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Host 1 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 1 NET 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 1 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 1 Port: 2] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 2 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 2] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 3 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 3 Port: 0] Deliver frame with SeqNum: 10
[INFO] - [SWP Switch 3 Port: 0] send ACK LFR: 10
[INFO] - [SWP Switch 2 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Router 1 NET 0] swp_protocol::processFrame
[INFO] - [SWP Router 1 NET 0] Deliver frame with SeqNum: 10

```

```

[INFO] - [SWP Router 1 NET 0] send ACK LFR: 10
[INFO] - [SWP Switch 2 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 3 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Router 1 NET 0] swp_protocol::processFrame
[INFO] - [SWP Router 1 NET 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 3 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 3 Port: 2] Deliver frame with SeqNum: 9
[INFO] - [SWP Switch 3 Port: 2] send ACK LFR: 9
[INFO] - [SWP Router 1 NET 0] swp_protocol::processFrame
[INFO] - [SWP Switch 3 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 3 Port: 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 2 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 2] Deliver frame with SeqNum: 9
[INFO] - [SWP Switch 2 Port: 2] send ACK LFR: 9
[INFO] - [SWP Host 2 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 2 NET 0] Deliver frame with SeqNum: 11
[INFO] - [SWP Host 2 NET 0] send ACK LFR: 11
[INFO] - [SWP Switch 3 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Host 2 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 2 NET 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 2 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 0] Deliver frame with SeqNum: 11
[INFO] - [SWP Switch 2 Port: 0] send ACK LFR: 11
[INFO] - [SWP Host 2 NET 0] swp_protocol::processFrame
[INFO] - [SWP Switch 1 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 1 Port: 2] Deliver frame with SeqNum: 8
[INFO] - [SWP Switch 1 Port: 2] send ACK LFR: 8
[INFO] - [SWP Switch 2 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 2] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 3 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 3 Port: 0] Deliver frame with SeqNum: 11
[INFO] - [SWP Switch 3 Port: 0] send ACK LFR: 11
[INFO] - [SWP Switch 2 Port: 1] swp_protocol::processFrame
[INFO] - [SWP Host 1 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 1 NET 0] Deliver frame with SeqNum: 8
[INFO] - [SWP Host 1 NET 0] send ACK LFR: 8
[INFO] - [SWP Router 1 NET 0] swp_protocol::processFrame
[INFO] - [SWP Router 1 NET 0] Deliver frame with SeqNum: 11
[INFO] - [SWP Router 1 NET 0] send ACK LFR: 11
[INFO] - [SWP Switch 2 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 1 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Host 1 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 1 NET 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 1 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 1 Port: 0] Deliver frame with SeqNum: 3
[INFO] - [SWP Switch 1 Port: 0] send ACK LFR: 3
[INFO] - [SWP Switch 3 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Router 1 NET 0] swp_protocol::processFrame
[INFO] - [SWP Router 1 NET 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 3 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 3 Port: 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Host 1 NET 0] swp_protocol::processFrame
[INFO] - [SWP Switch 1 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 1 Port: 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 2 Port: 1] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 1] Deliver frame with SeqNum: 3
[INFO] - [SWP Switch 2 Port: 1] send ACK LFR: 3
[INFO] - [SWP Host 2 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 2 NET 0] Deliver frame with SeqNum: 12
[INFO] - [SWP Host 2 NET 0] send ACK LFR: 12
[INFO] - [SWP Switch 1 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Host 2 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 2 NET 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 2 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 0] Deliver frame with SeqNum: 12

```

```

[INFO] - [SWP Switch 2 Port: 0] send ACK LFR: 12
[INFO] - [SWP Host 2 NET 0] swp-protocol::processFrame
[INFO] - [SWP Switch 2 Port: 1] swp-protocol::processFrame
[INFO] - [SWP Switch 2 Port: 1] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 1 Port: 2] swp-protocol::processFrame
[INFO] - [SWP Switch 1 Port: 2] Deliver frame with SeqNum: 9
[INFO] - [SWP Switch 1 Port: 2] send ACK LFR: 9
[INFO] - [SWP Host 1 NET 0] swp-protocol::processFrame
[INFO] - [SWP Host 1 NET 0] Deliver frame with SeqNum: 9
[INFO] - [SWP Host 1 NET 0] send ACK LFR: 9
[INFO] - [SWP Switch 2 Port: 1] swp-protocol::processFrame
[INFO] - [SWP Switch 1 Port: 0] swp-protocol::processFrame
[INFO] - [SWP Host 1 NET 0] swp-protocol::processFrame
[INFO] - [SWP Host 1 NET 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 1 Port: 2] swp-protocol::processFrame
[INFO] - [SWP Switch 1 Port: 2] Discarded frame: Frame not in RWS windows

```

Output de todo el scenario integrado generado por el logger durante la simulación

```

[INFO] - [IPv4 Host 1] ip amount: 2
[INFO] - [IPv4 Host 1] ips:
[INFO] - [IPv4 Host 1] 1.0.0.1
[INFO] - [IPv4 Host 1] 127.0.0.1
[INFO] - [IPv4 Host 1] Routing table:
[INFO] - [IPv4 Host 1] | 0.0.0.0 | 0.0.0.0 | 1.0.0.3 | 1 | Default Route |
[INFO] - [IPv4 Host 1] | 1.0.0.2 | 255.255.255.255 | 1.0.0.2 | 1 | Host 2 |
[INFO] - [IPv4 Host 1] Forwarding table:
[INFO] - [IPv4 Host 1] | 0.0.0.0 | 0.0.0.0 | 0 |
[INFO] - [Link Host 1 NET 0] MAC: 0:0:0:0:0:1
[INFO] - [Link Host 1 NET 0] IP: 1.0.0.1
[INFO] - [Link Host 1 NET 0] Interface: 0
[INFO] - [DNS Host 1] DNS local ip: 1.0.0.1
[INFO] - [DNS Host 1] DNS root server ip: 1.0.0.2
[INFO] - [DNS Host 1] No authoritative RR table file.
[INFO] - [DNS Host 1] No zone server table file.
[INFO] - [DNS Host 1] Recursive allowed: True
[INFO] - [Demultiplexer Host 1] Initialized with 1 interfaces
[INFO] - [IPv4 Host 3] ip amount: 2
[INFO] - [IPv4 Host 3] ips:
[INFO] - [IPv4 Host 3] 4.0.0.1
[INFO] - [IPv4 Host 3] 127.0.0.1
[INFO] - [IPv4 Host 3] Routing table:
[INFO] - [IPv4 Host 3] | 0.0.0.0 | 0.0.0.0 | 4.0.0.0 | 1 | Default Route |
[INFO] - [IPv4 Host 3] Forwarding table:
[INFO] - [IPv4 Host 3] | 0.0.0.0 | 0.0.0.0 | 0 |
[INFO] - [Link Host 3 NET 0] MAC: 0:0:0:0:0:8
[INFO] - [Link Host 3 NET 0] IP: 4.0.0.1
[INFO] - [Link Host 3 NET 0] Interface: 0
[INFO] - [DNS Host 3] DNS local ip: 4.0.0.1
[INFO] - [DNS Host 3] DNS root server ip: 5.0.0.1
[INFO] - [DNS Host 3] Authoritative RR table:
[INFO] - [DNS Host 3] network.devs.com A 6.0.0.1 IN 65535
[INFO] - [DNS Host 3] No zone server table file.
[INFO] - [DNS Host 3] Recursive allowed: True
[INFO] - [Demultiplexer Host 3] Initialized with 1 interfaces
[INFO] - [IPv4 Host 4] ip amount: 2
[INFO] - [IPv4 Host 4] ips:
[INFO] - [IPv4 Host 4] 5.0.0.2
[INFO] - [IPv4 Host 4] 127.0.0.1
[INFO] - [IPv4 Host 4] Routing table:
[INFO] - [IPv4 Host 4] | 0.0.0.0 | 0.0.0.0 | 5.0.0.0 | 1 | Default Route |
[INFO] - [IPv4 Host 4] | 5.0.0.1 | 255.255.255.255 | 5.0.0.1 | 1 | Host 5 |
[INFO] - [IPv4 Host 4] Forwarding table:
[INFO] - [IPv4 Host 4] | 0.0.0.0 | 0.0.0.0 | 0 |
[INFO] - [Link Host 4 NET 0] MAC: 0:0:0:0:0:12
[INFO] - [Link Host 4 NET 0] IP: 5.0.0.2
[INFO] - [Link Host 4 NET 0] Interface: 0
[INFO] - [DNS Host 4] DNS local ip: 5.0.0.2

```

```

[INFO] - [DNS Host 4] DNS root server ip: 5.0.0.1
[INFO] - [DNS Host 4] No authoritative RR table file.
[INFO] - [DNS Host 4]
----- Zone -----
** NS **
devs.com NS ns1.devs.com IN 65535
** A **
ns1.dev.com A 4.0.0.1 IN 65535
----- Zone -----
[INFO] - [DNS Host 4] Zone servers table:
[INFO] - [DNS Host 4]
----- Zone -----
** NS **
devs.com NS ns1.devs.com IN 65535
** A **
ns1.dev.com A 4.0.0.1 IN 65535
----- Zone -----
[INFO] - [DNS Host 4] Recursive allowed: False
[INFO] - [Demultiplexer Host 4] Initialized with 1 interfaces
[INFO] - [IPv4 Host 5] ip amount: 2
[INFO] - [IPv4 Host 5] ips:
[INFO] - [IPv4 Host 5] 5.0.0.1
[INFO] - [IPv4 Host 5] 127.0.0.1
[INFO] - [IPv4 Host 5] Routing table:
[INFO] - [IPv4 Host 5] | 0.0.0.0 | 0.0.0.0 | 5.0.0.0 | 1 | Default Route |
[INFO] - [IPv4 Host 5] | 5.0.0.2 | 255.255.255.255 | 5.0.0.2 | 1 | Host 4 |
[INFO] - [IPv4 Host 5] Forwarding table:
[INFO] - [IPv4 Host 5] | 0.0.0.0 | 0.0.0.0 | 0 |
[INFO] - [Link Host 5 NET 0] MAC: 0:0:0:0:0:11
[INFO] - [Link Host 5 NET 0] IP: 5.0.0.1
[INFO] - [Link Host 5 NET 0] Interface: 0
[INFO] - [DNS Host 5] DNS local ip: 5.0.0.1
[INFO] - [DNS Host 5] DNS root server ip: 5.0.0.1
[INFO] - [DNS Host 5] No authoritative RR table file.
[INFO] - [DNS Host 5]
----- Zone -----
** NS **
com NS ns1.com IN 65535
** A **
ns1.com A 5.0.0.2 IN 65535
----- Zone -----
[INFO] - [DNS Host 5] Zone servers table:
[INFO] - [DNS Host 5]
----- Zone -----
** NS **
com NS ns1.com IN 65535
** A **
ns1.com A 5.0.0.2 IN 65535
----- Zone -----
[INFO] - [DNS Host 5] Recursive allowed: False
[INFO] - [Demultiplexer Host 5] Initialized with 1 interfaces
[INFO] - [IPv4 Router 1] ip amount: 3
[INFO] - [IPv4 Router 1] ips:
[INFO] - [IPv4 Router 1] 1.0.0.0
[INFO] - [IPv4 Router 1] 2.0.0.0
[INFO] - [IPv4 Router 1] 3.0.0.0
[INFO] - [IPv4 Router 1] Routing table:
[INFO] - [IPv4 Router 1] | 0.0.0.0 | 0.0.0.0 | 3.0.0.1 | 1 | Default gateway |
[INFO] - [IPv4 Router 1] | 5.0.0.0 | 255.0.0.0 | 3.0.0.1 | 2 | Router 2 |
[INFO] - [IPv4 Router 1] | 4.0.0.0 | 255.0.0.0 | 2.0.0.1 | 2 | Router 3 |
[INFO] - [IPv4 Router 1] | 1.0.0.1 | 255.255.255.255 | 1.0.0.1 | 1 | Host 1 |
[INFO] - [IPv4 Router 1] | 1.0.0.2 | 255.255.255.255 | 1.0.0.2 | 1 | Host 2 |
[INFO] - [IPv4 Router 1] Forwarding table:
[INFO] - [IPv4 Router 1] | 1.0.0.0 | 255.0.0.0 | 0 |
[INFO] - [IPv4 Router 1] | 3.0.0.0 | 255.0.0.0 | 1 |
[INFO] - [IPv4 Router 1] | 2.0.0.0 | 255.0.0.0 | 2 |
[INFO] - [Link Router 1 NET 0] MAC: 0:0:0:0:0:3

```

```

[INFO] - [Link Router 1 NET 0] IP: 1.0.0.0
[INFO] - [Link Router 1 NET 0] Interface: 0
[INFO] - [Link Router 1 NET 1] MAC: 0:0:0:0:0:5
[INFO] - [Link Router 1 NET 1] IP: 3.0.0.0
[INFO] - [Link Router 1 NET 1] Interface: 1
[INFO] - [Demultiplexer Router 1] Initialized with 3 interfaces
[INFO] - [Link Router 1 NET 2] MAC: 0:0:0:0:0:4
[INFO] - [Link Router 1 NET 2] IP: 2.0.0.0
[INFO] - [Link Router 1 NET 2] Interface: 2
[INFO] - [IPv4 Router 2] ip amount: 2
[INFO] - [IPv4 Router 2] ips:
[INFO] - [IPv4 Router 2] 3.0.0.1
[INFO] - [IPv4 Router 2] 5.0.0.0
[INFO] - [IPv4 Router 2] Routing table:
[INFO] - [IPv4 Router 2] | 0.0.0.0 | 0.0.0.0 | 3.0.0.0 | 1 | Default gateway |
[INFO] - [IPv4 Router 2] | 1.0.0.0 | 255.0.0.0 | 3.0.0.0 | 2 | Router 1 |
[INFO] - [IPv4 Router 2] | 2.0.0.0 | 255.0.0.0 | 3.0.0.0 | 2 | Router 1 |
[INFO] - [IPv4 Router 2] | 4.0.0.0 | 255.0.0.0 | 3.0.0.0 | 3 | Router 1 |
[INFO] - [IPv4 Router 2] | 5.0.0.1 | 255.255.255.255 | 5.0.0.1 | 1 | Host 5 |
[INFO] - [IPv4 Router 2] | 5.0.0.2 | 255.255.255.255 | 5.0.0.2 | 1 | Host 4 |
[INFO] - [IPv4 Router 2] Forwarding table:
[INFO] - [IPv4 Router 2] | 3.0.0.0 | 255.0.0.0 | 0 |
[INFO] - [IPv4 Router 2] | 5.0.0.0 | 255.0.0.0 | 1 |
[INFO] - [Link Router 2 NET 0] MAC: 0:0:0:0:0:9
[INFO] - [Link Router 2 NET 0] IP: 3.0.0.1
[INFO] - [Link Router 2 NET 0] Interface: 0
[INFO] - [Link Router 2 NET 1] MAC: 0:0:0:0:0:10
[INFO] - [Link Router 2 NET 1] IP: 5.0.0.0
[INFO] - [Link Router 2 NET 1] Interface: 1
[INFO] - [Demultiplexer router 2] Initialized with 2 interfaces
[INFO] - [IPv4 Router 3] ip amount: 2
[INFO] - [IPv4 Router 3] ips:
[INFO] - [IPv4 Router 3] 2.0.0.1
[INFO] - [IPv4 Router 3] 4.0.0.0
[INFO] - [IPv4 Router 3] Routing table:
[INFO] - [IPv4 Router 3] | 0.0.0.0 | 0.0.0.0 | 2.0.0.0 | 1 | Default gateway |
[INFO] - [IPv4 Router 3] | 1.0.0.0 | 255.0.0.0 | 2.0.0.0 | 2 | Router 1 |
[INFO] - [IPv4 Router 3] | 3.0.0.0 | 255.0.0.0 | 2.0.0.0 | 2 | Router 1 |
[INFO] - [IPv4 Router 3] | 5.0.0.0 | 255.0.0.0 | 2.0.0.0 | 3 | Router 1 |
[INFO] - [IPv4 Router 3] | 4.0.0.1 | 255.255.255.255 | 4.0.0.1 | 1 | Host 3 |
[INFO] - [IPv4 Router 3] Forwarding table:
[INFO] - [IPv4 Router 3] | 2.0.0.0 | 255.0.0.0 | 0 |
[INFO] - [IPv4 Router 3] | 4.0.0.0 | 255.0.0.0 | 1 |
[INFO] - [Link Router 3 NET 0] MAC: 0:0:0:0:0:6
[INFO] - [Link Router 3 NET 0] IP: 2.0.0.1
[INFO] - [Link Router 3 NET 0] Interface: 0
[INFO] - [Link Router 3 NET 1] MAC: 0:0:0:0:0:7
[INFO] - [Link Router 3 NET 1] IP: 4.0.0.0
[INFO] - [Link Router 3 NET 1] Interface: 1
[INFO] - [Demultiplexer router 3] Initialized with 2 interfaces
[INFO] - [Switch Switch 1] MAC: 0:0:0:0:0:13
[INFO] - [Switch Switch 1] Interface amounts: 3
[INFO] - [Switch Switch 1] Added swp protocol for port 0
[INFO] - [Switch Switch 1] Added swp protocol for port 1
[INFO] - [Switch Switch 1] Added swp protocol for port 2
[INFO] - [Switch Switch 1] Forwarding table:
[INFO] - [Switch Switch 1] 0:0:0:0:0:1 | 0
[INFO] - [Switch Switch 1] 0:0:0:0:0:2 | 2
[INFO] - [Switch Switch 1] 0:0:0:0:0:3 | 2
[INFO] - [Demultiplexer Switch 1] Initialized with 3 interfaces
[INFO] - [Multiplexer Switch 1] Multiplexer initialized.
[INFO] - [Switch Switch 2] MAC: 0:0:0:0:0:14
[INFO] - [Switch Switch 2] Interface amounts: 3
[INFO] - [Switch Switch 2] Added swp protocol for port 0
[INFO] - [Switch Switch 2] Added swp protocol for port 1
[INFO] - [Switch Switch 2] Added swp protocol for port 2
[INFO] - [Switch Switch 2] Forwarding table:

```

```

[INFO] - [Switch Switch 2] 0:0:0:0:0:1 | 1
[INFO] - [Switch Switch 2] 0:0:0:0:0:2 | 0
[INFO] - [Switch Switch 2] 0:0:0:0:0:3 | 2
[INFO] - [Demultiplexer Switch 2] Initialized with 3 interfaces
[INFO] - [Multiplexer Switch 2] Multiplexer initialized .
[INFO] - [Switch Switch 3] MAC: 0:0:0:0:0:15
[INFO] - [Switch Switch 3] Interface amounts: 3
[INFO] - [Switch Switch 3] Added swp protocol for port 0
[INFO] - [Switch Switch 3] Added swp protocol for port 1
[INFO] - [Switch Switch 3] Added swp protocol for port 2
[INFO] - [Switch Switch 3] Forwarding table:
[INFO] - [Switch Switch 3] 0:0:0:0:0:1 | 0
[INFO] - [Switch Switch 3] 0:0:0:0:0:2 | 0
[INFO] - [Switch Switch 3] 0:0:0:0:0:3 | 2
[INFO] - [Demultiplexer Switch 3] Initialized with 3 interfaces
[INFO] - [Multiplexer Switch 3] Multiplexer initialized .
[INFO] - [Demultiplexer Switch 4] Initialized with 3 interfaces
[INFO] - [Multiplexer Switch 4] Multiplexer initialized .
[INFO] - [Switch Switch 4] MAC: 0:0:0:0:0:16
[INFO] - [Switch Switch 4] Interface amounts: 3
[INFO] - [Switch Switch 4] Added swp protocol for port 0
[INFO] - [Switch Switch 4] Added swp protocol for port 1
[INFO] - [Switch Switch 4] Added swp protocol for port 2
[INFO] - [Switch Switch 4] Forwarding table:
[INFO] - [Switch Switch 4] 0:0:0:0:0:10 | 0
[INFO] - [Switch Switch 4] 0:0:0:0:0:11 | 1
[INFO] - [Switch Switch 4] 0:0:0:0:0:12 | 2
[INFO] - [IPv4 Host 2] ip amount: 2
[INFO] - [IPv4 Host 2] ips:
[INFO] - [IPv4 Host 2] 1.0.0.2
[INFO] - [IPv4 Host 2] 127.0.0.1
[INFO] - [IPv4 Host 2] Routing table:
[INFO] - [IPv4 Host 2] | 0.0.0.0 | 0.0.0.0 | 1.0.0.0 | 1 | Default Route |
[INFO] - [IPv4 Host 2] | 1.0.0.1 | 255.255.255.255 | 1.0.0.1 | 1 | Host 1 |
[INFO] - [IPv4 Host 2] Forwarding table:
[INFO] - [IPv4 Host 2] | 0.0.0.0 | 0.0.0.0 | 0 |
[INFO] - [Link Host 2 NET 0] MAC: 0:0:0:0:0:2
[INFO] - [Link Host 2 NET 0] IP: 1.0.0.2
[INFO] - [Link Host 2 NET 0] Interface: 0
[INFO] - [DNS Host 2] DNS local ip: 1.0.0.2
[INFO] - [DNS Host 2] DNS root server ip: 5.0.0.1
[INFO] - [DNS Host 2] No authoritative RR table file .
[INFO] - [DNS Host 2] No zone server table file .
[INFO] - [DNS Host 2] Recursive allowed: True
[INFO] - [Demultiplexer Host 2] Initialized with 1 interfaces
Simulation Initialized
[INFO] - [UDP Host 1] Bind
[INFO] - [UDP Host 1] SUCCESS
[INFO] - [UDP Host 3] Bind
[INFO] - [UDP Host 3] SUCCESS
[INFO] - [UDP Host 4] Bind
[INFO] - [UDP Host 4] SUCCESS
[INFO] - [UDP Host 5] Bind
[INFO] - [UDP Host 5] SUCCESS
[INFO] - [UDP Host 2] Bind
[INFO] - [UDP Host 2] SUCCESS
[INFO] - [UDP Host 1] Read/Recv
[INFO] - [UDP Host 3] Read/Recv
[INFO] - [UDP Host 4] Read/Recv
[INFO] - [UDP Host 5] Read/Recv
[INFO] - [UDP Host 2] Read/Recv
[INFO] - [DNS Host 1] Process domain name network.devs.com
[INFO] - [DNS Host 1] Domain not in cache
[INFO] - [DNS Host 1] Send DNS Query to: (1.0.0.2, 53)
[INFO] - [UDP Host 1] Write_to/Send_to
[INFO] - [IPv4 Host 1] Forwarding ip packet: 1.0.0.2
[INFO] - [IPv4 Host 1] Best route for packet with dest_ip: 1.0.0.2

```

```

is : | 1.0.0.2 | 255.255.255.255 | 1.0.0.2 | 1 | Host 2 |
[INFO] - [IPv4 Host 1] Added packet to wait ARP for nexthop: 1.0.0.2
[INFO] - [IPv4 Host 1] ARP query throw interface: NET 0
[INFO] - [Link Host 1 NET 0] ARP_QUERY
[INFO] - [Link Host 1 NET 0] ARP ip 1.0.0.2 not cached.
[INFO] - [Link Host 1 NET 0] Sending ARP query for ip: 1.0.0.2
[INFO] - [SWP Switch 1 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 1 Port: 0] Deliver frame with SeqNum: 1
[INFO] - [SWP Switch 1 Port: 0] send ACK LFR: 1
[INFO] - [Switch Switch 1] Broadcast MAC sent throw all interfaces
[INFO] - [Switch Switch 1] send frame for MAC: ff:ff:ff:ff:ff:ff throw interface 2
[INFO] - [SWP Host 1 NET 0] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 1] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 1] Deliver frame with SeqNum: 1
[INFO] - [SWP Switch 2 Port: 1] send ACK LFR: 1
[INFO] - [Switch Switch 2] Broadcast MAC sent throw all interfaces
[INFO] - [Switch Switch 2] send frame for MAC: ff:ff:ff:ff:ff:ff throw interface 0
[INFO] - [Switch Switch 2] send frame for MAC: ff:ff:ff:ff:ff:ff throw interface 2
[INFO] - [SWP Host 2 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 2 NET 0] Deliver frame with SeqNum: 1
[INFO] - [SWP Host 2 NET 0] send ACK LFR: 1
[INFO] - [SWP Switch 3 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 3 Port: 0] Deliver frame with SeqNum: 1
[INFO] - [SWP Switch 3 Port: 0] send ACK LFR: 1
[INFO] - [Link Host 2 NET 0] Process ARP packet.
[INFO] - [Switch Switch 3] Broadcast MAC sent throw all interfaces
[INFO] - [Switch Switch 3] send frame for MAC: ff:ff:ff:ff:ff:ff throw interface 2
[INFO] - [SWP Router 1 NET 0] swp_protocol::processFrame
[INFO] - [SWP Router 1 NET 0] Deliver frame with SeqNum: 1
[INFO] - [SWP Router 1 NET 0] send ACK LFR: 1
[INFO] - [Link Router 1 NET 0] Process ARP packet.
[INFO] - [SWP Switch 1 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Host 2 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 2 NET 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 2 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 0] Deliver frame with SeqNum: 1
[INFO] - [SWP Switch 2 Port: 0] send ACK LFR: 1
[INFO] - [Switch Switch 2] send frame for MAC: 0:0:0:0:0:1 throw interface 1
[INFO] - [SWP Host 2 NET 0] swp_protocol::processFrame
[INFO] - [SWP Switch 3 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Router 1 NET 0] swp_protocol::processFrame
[INFO] - [SWP Router 1 NET 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 3 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 3 Port: 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 2 Port: 1] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 1] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 1 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 1 Port: 2] Deliver frame with SeqNum: 1
[INFO] - [SWP Switch 1 Port: 2] send ACK LFR: 1
[INFO] - [SWP Switch 2 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 0] Discarded frame: Frame not in RWS windows
[INFO] - [Switch Switch 1] send frame for MAC: 0:0:0:0:0:1 throw interface 0
[INFO] - [SWP Host 1 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 1 NET 0] Deliver frame with SeqNum: 1
[INFO] - [SWP Host 1 NET 0] send ACK LFR: 1
[INFO] - [Link Host 1 NET 0] Process ARP packet.
[INFO] - [Link Host 1 NET 0] Send control ARP_READY
[INFO] - [IPv4 Host 1] link::Ctrl::ARP_READY
[INFO] - [Link Host 1 NET 0] SEND_PACKET
[INFO] - [Link Host 1 NET 0] ARP ip 1.0.0.2 is in cache.
[INFO] - [SWP Switch 2 Port: 1] swp_protocol::processFrame
[INFO] - [SWP Switch 1 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Host 1 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 1 NET 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 1 Port: 0] swp_protocol::processFrame

```

```

[INFO] - [SWP Switch 1 Port: 0] Deliver frame with SeqNum: 2
[INFO] - [SWP Switch 1 Port: 0] send ACK LFR: 2
[INFO] - [Switch Switch 1] send frame for MAC: 0:0:0:0:0:2 throw interface 2
[INFO] - [SWP Host 1 NET 0] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 1] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 1] Deliver frame with SeqNum: 2
[INFO] - [SWP Switch 2 Port: 1] send ACK LFR: 2
[INFO] - [Switch Switch 2] send frame for MAC: 0:0:0:0:0:2 throw interface 0
[INFO] - [SWP Host 2 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 2 NET 0] Deliver frame with SeqNum: 2
[INFO] - [SWP Host 2 NET 0] send ACK LFR: 2
[INFO] - [IPv4 Host 2] Processing arrived ip packet: 1.0.0.2
[INFO] - [IPv4 Host 2] Valid checksum: true
[INFO] - [UDP Host 2] process Segment:
[INFO] - [UDP Host 2] local_port: 53
[INFO] - [UDP Host 2] remote_port: 53
[INFO] - [UDP Host 2] local_ip: 1.0.0.2
[INFO] - [UDP Host 2] remote_ip: 1.0.0.1
[INFO] - [UDP Host 2] valid checksum: true
[INFO] - [UDP Host 2] socket local_ip: 1.0.0.2
[INFO] - [UDP Host 2] socket local_ip: 53
[INFO] - [UDP Host 2] Deliver dns packet
[INFO] - [DNS Host 2] Incoming packet:
[INFO] - [DNS Host 2] dns packet
** HEADER **
id: 0
flags_code: 33024
QDCount: 1
ANCount: 0
NSCount: 0
ARCount: 1

** QUESTION SECTION **
network.devs.com A 0.0.0.0 IN 0

** ANSWER SECTION **

** AUTHORITATIVES SECTION **

** ADDITIONAL SECTION **
network.devs.com A 1.0.0.1 IN 9227
[INFO] - [DNS Host 2] Query packet
[INFO] - [DNS Host 2] Recursive query for domain network.devs.com
[INFO] - [DNS Host 2] Send DNS Query to: (5.0.0.1, 53)
[INFO] - [UDP Host 2] Write_to/Send_to
[INFO] - [UDP Host 2] Read/Recv
[INFO] - [IPv4 Host 2] Forwarding ip packet: 5.0.0.1
[INFO] - [IPv4 Host 2] Best route for packet with dest_ip: 5.0.0.1
is: | 0.0.0.0 | 0.0.0.0 | 1.0.0.0 | 1 | Default Route |
[INFO] - [IPv4 Host 2] Added packet to wait ARP for nexthop: 1.0.0.0
[INFO] - [IPv4 Host 2] ARP query throw interface: NET 0
[INFO] - [Link Host 2 NET 0] ARP_QUERY
[INFO] - [Link Host 2 NET 0] ARP ip 1.0.0.0 not cached.
[INFO] - [Link Host 2 NET 0] Sending ARP query for ip: 1.0.0.0
[INFO] - [SWP Switch 1 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Host 2 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 2 NET 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 2 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 0] Deliver frame with SeqNum: 2
[INFO] - [SWP Switch 2 Port: 0] send ACK LFR: 2
[INFO] - [Switch Switch 2] Broadcast MAC sent throw all interfaces
[INFO] - [Switch Switch 2] send frame for MAC: ff:ff:ff:ff:ff:ff throw interface 1
[INFO] - [Switch Switch 2] send frame for MAC: ff:ff:ff:ff:ff:ff throw interface 2
[INFO] - [SWP Host 2 NET 0] swp_protocol::processFrame

```

```

[INFO] - [SWP Switch 1 Port: 2] swp-protocol::processFrame
[INFO] - [SWP Switch 1 Port: 2] Deliver frame with SeqNum: 2
[INFO] - [SWP Switch 1 Port: 2] send ACK LFR: 2
[INFO] - [Switch Switch 1] Broadcast MAC sent throw all interfaces
[INFO] - [Switch Switch 1] send frame for MAC: ff:ff:ff:ff:ff:ff throw interface 0
[INFO] - [SWP Switch 2 Port: 1] swp-protocol::processFrame
[INFO] - [SWP Switch 2 Port: 1] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 3 Port: 0] swp-protocol::processFrame
[INFO] - [SWP Switch 3 Port: 0] Deliver frame with SeqNum: 2
[INFO] - [SWP Switch 3 Port: 0] send ACK LFR: 2
[INFO] - [SWP Switch 2 Port: 1] swp-protocol::processFrame
[INFO] - [Switch Switch 3] Broadcast MAC sent throw all interfaces
[INFO] - [Switch Switch 3] send frame for MAC: ff:ff:ff:ff:ff:ff throw interface 2
[INFO] - [SWP Host 1 NET 0] swp-protocol::processFrame
[INFO] - [SWP Host 1 NET 0] Deliver frame with SeqNum: 2
[INFO] - [SWP Host 1 NET 0] send ACK LFR: 2
[INFO] - [Link Host 1 NET 0] Process ARP packet.
[INFO] - [SWP Router 1 NET 0] swp-protocol::processFrame
[INFO] - [SWP Router 1 NET 0] Deliver frame with SeqNum: 2
[INFO] - [SWP Router 1 NET 0] send ACK LFR: 2
[INFO] - [Link Router 1 NET 0] Process ARP packet.
[INFO] - [SWP Switch 2 Port: 2] swp-protocol::processFrame
[INFO] - [SWP Switch 1 Port: 0] swp-protocol::processFrame
[INFO] - [SWP Host 1 NET 0] swp-protocol::processFrame
[INFO] - [SWP Host 1 NET 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 3 Port: 2] swp-protocol::processFrame
[INFO] - [SWP Router 1 NET 0] swp-protocol::processFrame
[INFO] - [SWP Router 1 NET 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 3 Port: 2] swp-protocol::processFrame
[INFO] - [SWP Switch 3 Port: 2] Deliver frame with SeqNum: 1
[INFO] - [SWP Switch 3 Port: 2] send ACK LFR: 1
[INFO] - [Switch Switch 3] send frame for MAC: 0:0:0:0:0:2 throw interface 0
[INFO] - [SWP Router 1 NET 0] swp-protocol::processFrame
[INFO] - [SWP Switch 3 Port: 0] swp-protocol::processFrame
[INFO] - [SWP Switch 3 Port: 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 2 Port: 2] swp-protocol::processFrame
[INFO] - [SWP Switch 2 Port: 2] Deliver frame with SeqNum: 1
[INFO] - [SWP Switch 2 Port: 2] send ACK LFR: 1
[INFO] - [SWP Switch 3 Port: 2] swp-protocol::processFrame
[INFO] - [SWP Switch 3 Port: 2] Discarded frame: Frame not in RWS windows
[INFO] - [Switch Switch 2] send frame for MAC: 0:0:0:0:0:2 throw interface 0
[INFO] - [SWP Host 2 NET 0] swp-protocol::processFrame
[INFO] - [SWP Host 2 NET 0] Deliver frame with SeqNum: 3
[INFO] - [SWP Host 2 NET 0] send ACK LFR: 3
[INFO] - [Link Host 2 NET 0] Process ARP packet.
[INFO] - [Link Host 2 NET 0] Send control ARP_READY
[INFO] - [IPv4 Host 2] link::Ctrl::ARP_READY
[INFO] - [Link Host 2 NET 0] SEND_PACKET
[INFO] - [Link Host 2 NET 0] ARP ip 1.0.0.0 is in cache.
[INFO] - [SWP Switch 3 Port: 0] swp-protocol::processFrame
[INFO] - [SWP Switch 2 Port: 0] swp-protocol::processFrame
[INFO] - [SWP Host 2 NET 0] swp-protocol::processFrame
[INFO] - [SWP Host 2 NET 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 2 Port: 0] swp-protocol::processFrame
[INFO] - [SWP Switch 2 Port: 0] Deliver frame with SeqNum: 3
[INFO] - [SWP Switch 2 Port: 0] send ACK LFR: 3
[INFO] - [Switch Switch 2] send frame for MAC: 0:0:0:0:0:3 throw interface 2
[INFO] - [SWP Host 2 NET 0] swp-protocol::processFrame
[INFO] - [SWP Switch 2 Port: 2] swp-protocol::processFrame
[INFO] - [SWP Switch 2 Port: 2] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 3 Port: 0] swp-protocol::processFrame
[INFO] - [SWP Switch 3 Port: 0] Deliver frame with SeqNum: 3
[INFO] - [SWP Switch 3 Port: 0] send ACK LFR: 3
[INFO] - [Switch Switch 3] send frame for MAC: 0:0:0:0:0:3 throw interface 2
[INFO] - [SWP Router 1 NET 0] swp-protocol::processFrame
[INFO] - [SWP Router 1 NET 0] Deliver frame with SeqNum: 3
[INFO] - [SWP Router 1 NET 0] send ACK LFR: 3

```

```

[INFO] - [IPv4 Router 1] Process Ip Datagram
[INFO] - [IPv4 Router 1] Valid checksum: true
[INFO] - [IPv4 Router 1] Forwarding ip packet: 5.0.0.1
[INFO] - [IPv4 Router 1] Best route for packet with dest_ip: 5.0.0.1
is : | 5.0.0.0 | 255.0.0.0 | 3.0.0.1 | 2 | Router 2 |
[INFO] - [IPv4 Router 1] Added packet to wait ARP for nexthop: 3.0.0.1
[INFO] - [IPv4 Router 1] ARP query throw interface: NET 1
[INFO] - [Link Router 1 NET 1] ARP_QUERY
[INFO] - [Link Router 1 NET 1] ARP ip 3.0.0.1 not cached.
[INFO] - [Link Router 1 NET 1] Sending ARP query for ip: 3.0.0.1
[INFO] - [SWP Router 2 NET 0] swp_protocol::processFrame
[INFO] - [SWP Router 2 NET 0] Deliver frame with SeqNum: 1
[INFO] - [SWP Router 2 NET 0] send ACK LFR: 1
[INFO] - [Link Router 2 NET 0] Process ARP packet.
[INFO] - [SWP Router 1 NET 1] swp_protocol::processFrame
[INFO] - [SWP Router 1 NET 1] swp_protocol::processFrame
[INFO] - [SWP Router 1 NET 1] Deliver frame with SeqNum: 1
[INFO] - [SWP Router 1 NET 1] send ACK LFR: 1
[INFO] - [Link Router 1 NET 1] Process ARP packet.
[INFO] - [Link Router 1 NET 1] Send control ARP_READY
[INFO] - [SWP Router 2 NET 0] swp_protocol::processFrame
[INFO] - [IPv4 Router 1] link::Ctrl::ARP_READY
[INFO] - [Link Router 1 NET 1] SEND_PACKET
[INFO] - [Link Router 1 NET 1] ARP ip 3.0.0.1 is in cache.
[INFO] - [SWP Router 2 NET 0] swp_protocol::processFrame
[INFO] - [SWP Router 2 NET 0] Deliver frame with SeqNum: 2
[INFO] - [SWP Router 2 NET 0] send ACK LFR: 2
[INFO] - [SWP Router 1 NET 1] swp_protocol::processFrame
[INFO] - [IPv4 Router 2] Process Ip Datagram
[INFO] - [IPv4 Router 2] Valid checksum: true
[INFO] - [IPv4 Router 2] Forwarding ip packet: 5.0.0.1
[INFO] - [IPv4 Router 2] Best route for packet with dest_ip: 5.0.0.1
is : | 5.0.0.1 | 255.255.255.255 | 5.0.0.1 | 1 | Host 5 |
[INFO] - [IPv4 Router 2] Added packet to wait ARP for nexthop: 5.0.0.1
[INFO] - [IPv4 Router 2] ARP query throw interface: NET 1
[INFO] - [Link Router 2 NET 1] ARP_QUERY
[INFO] - [Link Router 2 NET 1] ARP ip 5.0.0.1 not cached.
[INFO] - [Link Router 2 NET 1] Sending ARP query for ip: 5.0.0.1
[INFO] - [SWP Switch 4 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 4 Port: 0] Deliver frame with SeqNum: 1
[INFO] - [SWP Switch 4 Port: 0] send ACK LFR: 1
[INFO] - [Switch Switch 4] Broadcast MAC sent throw all interfaces
[INFO] - [Switch Switch 4] send frame for MAC: ff:ff:ff:ff:ff:ff throw interface 1
[INFO] - [Switch Switch 4] send frame for MAC: ff:ff:ff:ff:ff:ff throw interface 2
[INFO] - [SWP Router 2 NET 1] swp_protocol::processFrame
[INFO] - [SWP Host 5 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 5 NET 0] Deliver frame with SeqNum: 1
[INFO] - [SWP Host 5 NET 0] send ACK LFR: 1
[INFO] - [Link Host 5 NET 0] Process ARP packet.
[INFO] - [SWP Switch 4 Port: 1] swp_protocol::processFrame
[INFO] - [SWP Host 4 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 4 NET 0] Deliver frame with SeqNum: 1
[INFO] - [SWP Host 4 NET 0] send ACK LFR: 1
[INFO] - [Link Host 4 NET 0] Process ARP packet.
[INFO] - [SWP Switch 2 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 3 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Router 1 NET 0] swp_protocol::processFrame
[INFO] - [SWP Router 1 NET 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 3 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 3 Port: 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 4 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Host 4 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 4 NET 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 4 Port: 1] swp_protocol::processFrame
[INFO] - [SWP Switch 4 Port: 1] Deliver frame with SeqNum: 1
[INFO] - [SWP Switch 4 Port: 1] send ACK LFR: 1
[INFO] - [Switch Switch 4] send frame for MAC: 0:0:0:0:0:10 throw interface 0

```

```

[INFO] - [SWP Host 5 NET 0] swp_protocol::processFrame
[INFO] - [SWP Switch 4 Port: 1] swp_protocol::processFrame
[INFO] - [SWP Switch 4 Port: 1] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Router 2 NET 1] swp_protocol::processFrame
[INFO] - [SWP Router 2 NET 1] Deliver frame with SeqNum: 1
[INFO] - [SWP Router 2 NET 1] send ACK LFR: 1
[INFO] - [Link Router 2 NET 1] Process ARP packet.
[INFO] - [Link Router 2 NET 1] Send control ARP_READY
[INFO] - [IPv4 Router 2] link::Ctrl::ARP_READY
[INFO] - [Link Router 2 NET 1] SEND_PACKET
[INFO] - [Link Router 2 NET 1] ARP ip 5.0.0.1 is in cache.
[INFO] - [SWP Switch 4 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Router 2 NET 1] swp_protocol::processFrame
[INFO] - [SWP Router 2 NET 1] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 4 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 4 Port: 0] Deliver frame with SeqNum: 2
[INFO] - [SWP Switch 4 Port: 0] send ACK LFR: 2
[INFO] - [Switch Switch 4] send frame for MAC: 0:0:0:0:0:11 throw interface 1
[INFO] - [SWP Router 2 NET 1] swp_protocol::processFrame
[INFO] - [SWP Host 5 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 5 NET 0] Deliver frame with SeqNum: 2
[INFO] - [SWP Host 5 NET 0] send ACK LFR: 2
[INFO] - [IPv4 Host 5] Processing arrived ip packet: 5.0.0.1
[INFO] - [IPv4 Host 5] Valid checksum: true
[INFO] - [UDP Host 5] process Segment:
[INFO] - [UDP Host 5] local_port: 53
[INFO] - [UDP Host 5] remote_port: 53
[INFO] - [UDP Host 5] local_ip: 5.0.0.1
[INFO] - [UDP Host 5] remote_ip: 1.0.0.2
[INFO] - [UDP Host 5] valid checksum: true
[INFO] - [UDP Host 5] socket local_ip: 5.0.0.1
[INFO] - [UDP Host 5] socket local_ip: 53
[INFO] - [UDP Host 5] Deliver dns packet
[INFO] - [DNS Host 5] Incoming packet:
[INFO] - [DNS Host 5]
----- dns packet -----
** HEADER **
id: 0
flags_code: 33024
QDCount: 1
ANCount: 0
NSCount: 0
ARCount: 1

** QUESTION SECTION **
network.devs.com A 0.0.0.0 IN 0

** ANSWER SECTION **

** AUTHORITATIVES SECTION **

** ADDITIONAL SECTION **
network.devs.com A 1.0.0.2 IN 9227
-----
```

```

[INFO] - [DNS Host 5] Query packet
[INFO] - [DNS Host 5] No recursive query for domain network.devs.com
[INFO] - [DNS Host 5] Zone found:
----- Zone -----
** NS **
com NS ns1.com IN 65535
** A **
ns1.com A 5.0.0.2 IN 65535
-----
```

```

[INFO] - [DNS Host 5] Send DNS Query to: (1.0.0.2, 53)
[INFO] - [UDP Host 5] Write_to/Send_to
[INFO] - [UDP Host 5] Read/Recv
[INFO] - [IPv4 Host 5] Forwarding ip packet: 1.0.0.2
```

```

[INFO] - [IPv4 Host 5] Best route for packet with dest_ip: 1.0.0.2
is : | 0.0.0.0 | 0.0.0.0 | 5.0.0.0 | 1 | Default Route |
[INFO] - [IPv4 Host 5] Added packet to wait ARP for nexthop: 5.0.0.0
[INFO] - [IPv4 Host 5] ARP query throw interface: NET 0
[INFO] - [Link Host 5 NET 0] ARP_QUERY
[INFO] - [Link Host 5 NET 0] ARP ip 5.0.0.0 not cached.
[INFO] - [Link Host 5 NET 0] Sending ARP query for ip: 5.0.0.0
[INFO] - [SWP Switch 4 Port: 1] swp_protocol::processFrame
[INFO] - [SWP Host 5 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 5 NET 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 4 Port: 1] swp_protocol::processFrame
[INFO] - [SWP Switch 4 Port: 1] Deliver frame with SeqNum: 2
[INFO] - [SWP Switch 4 Port: 1] send ACK LFR: 2
[INFO] - [Switch Switch 4] Broadcast MAC sent throw all interfaces
[INFO] - [Switch Switch 4] send frame for MAC: ff:ff:ff:ff:ff:ff throw interface 0
[INFO] - [Switch Switch 4] send frame for MAC: ff:ff:ff:ff:ff:ff throw interface 2
[INFO] - [SWP Host 5 NET 0] swp_protocol::processFrame
[INFO] - [SWP Router 2 NET 1] swp_protocol::processFrame
[INFO] - [SWP Router 2 NET 1] Deliver frame with SeqNum: 2
[INFO] - [SWP Router 2 NET 1] send ACK LFR: 2
[INFO] - [Link Router 2 NET 1] Process ARP packet.
[INFO] - [SWP Switch 4 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Host 4 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 4 NET 0] Deliver frame with SeqNum: 2
[INFO] - [SWP Host 4 NET 0] send ACK LFR: 2
[INFO] - [Link Host 4 NET 0] Process ARP packet.
[INFO] - [SWP Switch 4 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Host 4 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 4 NET 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 4 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 4 Port: 0] Deliver frame with SeqNum: 3
[INFO] - [SWP Switch 4 Port: 0] send ACK LFR: 3
[INFO] - [Switch Switch 4] send frame for MAC: 0:0:0:0:0:11 throw interface 1
[INFO] - [SWP Router 2 NET 1] swp_protocol::processFrame
[INFO] - [SWP Switch 4 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 4 Port: 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Host 5 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 5 NET 0] Deliver frame with SeqNum: 3
[INFO] - [SWP Host 5 NET 0] send ACK LFR: 3
[INFO] - [Link Host 5 NET 0] Process ARP packet.
[INFO] - [Link Host 5 NET 0] Send control ARP_READY
[INFO] - [IPv4 Host 5] link::Ctrl::ARP_READY
[INFO] - [Link Host 5 NET 0] SEND_PACKET
[INFO] - [Link Host 5 NET 0] ARP ip 5.0.0.0 is in cache.
[INFO] - [SWP Switch 4 Port: 1] swp_protocol::processFrame
[INFO] - [SWP Host 5 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 5 NET 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 4 Port: 1] swp_protocol::processFrame
[INFO] - [SWP Switch 4 Port: 1] Deliver frame with SeqNum: 3
[INFO] - [SWP Switch 4 Port: 1] send ACK LFR: 3
[INFO] - [Switch Switch 4] send frame for MAC: 0:0:0:0:0:10 throw interface 0
[INFO] - [SWP Host 5 NET 0] swp_protocol::processFrame
[INFO] - [SWP Router 2 NET 1] swp_protocol::processFrame
[INFO] - [SWP Router 2 NET 1] Deliver frame with SeqNum: 3
[INFO] - [SWP Router 2 NET 1] send ACK LFR: 3
[INFO] - [IPv4 Router 2] Process Ip Datagram
[INFO] - [IPv4 Router 2] Valid checksum: true
[INFO] - [IPv4 Router 2] Forwarding ip packet: 1.0.0.2
[INFO] - [IPv4 Router 2] Best route for packet with dest_ip: 1.0.0.2
is : | 1.0.0.0 | 255.0.0.0 | 3.0.0.0 | 2 | Router 1 |
[INFO] - [IPv4 Router 2] Added packet to wait ARP for nexthop: 3.0.0.0
[INFO] - [IPv4 Router 2] ARP query throw interface: NET 0
[INFO] - [Link Router 2 NET 0] ARP_QUERY
[INFO] - [Link Router 2 NET 0] ARP ip 3.0.0.0 not cached.
[INFO] - [Link Router 2 NET 0] Sending ARP query for ip: 3.0.0.0
[INFO] - [SWP Router 1 NET 1] swp_protocol::processFrame
[INFO] - [SWP Router 1 NET 1] Deliver frame with SeqNum: 2

```

```

[INFO] - [SWP Router 1 NET 1] send ACK LFR: 2
[INFO] - [Link Router 1 NET 1] Process ARP packet.
[INFO] - [SWP Router 2 NET 0] swp_protocol::processFrame
[INFO] - [SWP Router 2 NET 0] swp_protocol::processFrame
[INFO] - [SWP Router 2 NET 0] Deliver frame with SeqNum: 3
[INFO] - [SWP Router 2 NET 0] send ACK LFR: 3
[INFO] - [Link Router 2 NET 0] Process ARP packet.
[INFO] - [Link Router 2 NET 0] Send control ARP_READY
[INFO] - [SWP Router 1 NET 1] swp_protocol::processFrame
[INFO] - [IPv4 Router 2] link::Ctrl::ARP_READY
[INFO] - [Link Router 2 NET 0] SEND_PACKET
[INFO] - [Link Router 2 NET 0] ARP ip 3.0.0.0 is in cache.
[INFO] - [SWP Router 1 NET 1] swp_protocol::processFrame
[INFO] - [SWP Router 1 NET 1] Deliver frame with SeqNum: 3
[INFO] - [SWP Router 1 NET 1] send ACK LFR: 3
[INFO] - [SWP Router 2 NET 0] swp_protocol::processFrame
[INFO] - [IPv4 Router 1] Process Ip Datagram
[INFO] - [IPv4 Router 1] Valid checksum: true
[INFO] - [IPv4 Router 1] Forwarding ip packet: 1.0.0.2
[INFO] - [IPv4 Router 1] Best route for packet with dest_ip: 1.0.0.2
is: | 1.0.0.2 | 255.255.255.255 | 1.0.0.2 | 1 | Host 2 |
[INFO] - [IPv4 Router 1] Added packet to wait ARP for nexthop: 1.0.0.2
[INFO] - [IPv4 Router 1] ARP query throw interface: NET 0
[INFO] - [Link Router 1 NET 0] ARP_QUERY
[INFO] - [Link Router 1 NET 0] ARP ip 1.0.0.2 not cached.
[INFO] - [Link Router 1 NET 0] Sending ARP query for ip: 1.0.0.2
[INFO] - [SWP Switch 3 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 3 Port: 2] Deliver frame with SeqNum: 2
[INFO] - [SWP Switch 3 Port: 2] send ACK LFR: 2
[INFO] - [Switch Switch 3] Broadcast MAC sent throw all interfaces
[INFO] - [Switch Switch 3] send frame for MAC: ff:ff:ff:ff:ff:ff throw interface 0
[INFO] - [SWP Router 1 NET 0] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 2] Deliver frame with SeqNum: 2
[INFO] - [SWP Switch 2 Port: 2] send ACK LFR: 2
[INFO] - [Switch Switch 2] Broadcast MAC sent throw all interfaces
[INFO] - [Switch Switch 2] send frame for MAC: ff:ff:ff:ff:ff:ff throw interface 0
[INFO] - [Switch Switch 2] send frame for MAC: ff:ff:ff:ff:ff:ff throw interface 1
[INFO] - [SWP Host 2 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 2 NET 0] Deliver frame with SeqNum: 4
[INFO] - [SWP Host 2 NET 0] send ACK LFR: 4
[INFO] - [SWP Switch 1 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 1 Port: 2] Deliver frame with SeqNum: 3
[INFO] - [SWP Switch 1 Port: 2] send ACK LFR: 3
[INFO] - [Link Host 2 NET 0] Process ARP packet.
[INFO] - [Switch Switch 1] Broadcast MAC sent throw all interfaces
[INFO] - [Switch Switch 1] send frame for MAC: ff:ff:ff:ff:ff:ff throw interface 0
[INFO] - [SWP Host 1 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 1 NET 0] Deliver frame with SeqNum: 3
[INFO] - [SWP Host 1 NET 0] send ACK LFR: 3
[INFO] - [Link Host 1 NET 0] Process ARP packet.
[INFO] - [SWP Switch 4 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Router 2 NET 1] swp_protocol::processFrame
[INFO] - [SWP Router 2 NET 1] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 3 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Host 2 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 2 NET 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 2 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 1] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 0] Deliver frame with SeqNum: 4
[INFO] - [SWP Switch 2 Port: 0] send ACK LFR: 4
[INFO] - [Switch Switch 2] send frame for MAC: 0:0:0:0:0:3 throw interface 2
[INFO] - [SWP Host 2 NET 0] swp_protocol::processFrame
[INFO] - [SWP Switch 1 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Host 1 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 1 NET 0] Discarded frame: Frame not in RWS windows

```

```

[INFO] - [SWP Switch 1 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 1 Port: 2] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 2 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 2] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 3 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 3 Port: 0] Deliver frame with SeqNum: 4
[INFO] - [SWP Switch 3 Port: 0] send ACK LFR: 4
[INFO] - [SWP Switch 2 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 0] Discarded frame: Frame not in RWS windows
[INFO] - [Switch Switch 3] send frame for MAC: 0:0:0:0:0:3 throw interface 2
[INFO] - [SWP Router 1 NET 0] swp_protocol::processFrame
[INFO] - [SWP Router 1 NET 0] Deliver frame with SeqNum: 4
[INFO] - [SWP Router 1 NET 0] send ACK LFR: 4
[INFO] - [Link Router 1 NET 0] Process ARP packet.
[INFO] - [Link Router 1 NET 0] Send control ARP_READY
[INFO] - [IPv4 Router 1] link::Ctrl::ARP_READY
[INFO] - [Link Router 1 NET 0] SEND_PACKET
[INFO] - [Link Router 1 NET 0] ARP ip 1.0.0.2 is in cache.
[INFO] - [SWP Switch 2 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 3 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Router 1 NET 0] swp_protocol::processFrame
[INFO] - [SWP Router 1 NET 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 3 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 3 Port: 2] Deliver frame with SeqNum: 3
[INFO] - [SWP Switch 3 Port: 2] send ACK LFR: 3
[INFO] - [Switch Switch 3] send frame for MAC: 0:0:0:0:0:2 throw interface 0
[INFO] - [SWP Router 1 NET 0] swp_protocol::processFrame
[INFO] - [SWP Switch 3 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 3 Port: 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 2 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 2] Deliver frame with SeqNum: 3
[INFO] - [SWP Switch 2 Port: 2] send ACK LFR: 3
[INFO] - [Switch Switch 2] send frame for MAC: 0:0:0:0:0:2 throw interface 0
[INFO] - [SWP Host 2 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 2 NET 0] Deliver frame with SeqNum: 5
[INFO] - [SWP Host 2 NET 0] send ACK LFR: 5
[INFO] - [IPv4 Host 2] Processing arrived ip packet: 1.0.0.2
[INFO] - [IPv4 Host 2] Valid checksum: true
[INFO] - [UDP Host 2] process Segment:
[INFO] - [UDP Host 2] local_port: 53
[INFO] - [UDP Host 2] remote_port: 53
[INFO] - [UDP Host 2] local_ip: 1.0.0.2
[INFO] - [UDP Host 2] remote_ip: 5.0.0.1
[INFO] - [UDP Host 2] valid checksum: true
[INFO] - [UDP Host 2] socket local_ip: 1.0.0.2
[INFO] - [UDP Host 2] socket local_ip: 53
[INFO] - [UDP Host 2] Deliver dns packet
[INFO] - [DNS Host 2] Incoming packet:
[INFO] - [DNS Host 2] dns packet

** HEADER **
id: 0
flags_code: 256
QDCount: 1
ANCount: 1
NSCount: 1
ARCount: 1

** QUESTION SECTION **
network.devs.com A 0.0.0.0 IN 0

** ANSWER SECTION **
com NS ns1.com IN 65535

** AUTHORITATIVES SECTION **
ns1.com A 5.0.0.2 IN 65535

```

** ADDITIONAL SECTION **

network.devs.com A 5.0.0.1 IN 9227

```
[INFO] - [DNS Host 2] Answer packet
[INFO] - [DNS Host 2] Answer Type: NS
[INFO] - [DNS Host 2] Send DNS Query to: (5.0.0.2, 53)
[INFO] - [UDP Host 2] Write_to/Send_to
[INFO] - [UDP Host 2] Read/Recv
[INFO] - [IPv4 Host 2] Forwarding ip packet: 5.0.0.2
[INFO] - [IPv4 Host 2] Best route for packet with dest_ip: 5.0.0.2
is: | 0.0.0.0 | 0.0.0.0 | 1.0.0.0 | 1 | Default Route |
[INFO] - [IPv4 Host 2] Added packet to wait ARP for nexthop: 1.0.0.0
[INFO] - [IPv4 Host 2] ARP query throw interface: NET 0
[INFO] - [Link Host 2 NET 0] ARP_QUERY
[INFO] - [Link Host 2 NET 0] ARP ip 1.0.0.0 not cached.
[INFO] - [Link Host 2 NET 0] Sending ARP query for ip: 1.0.0.0
[INFO] - [SWP Switch 3 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Host 2 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 2 NET 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 2 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 0] Deliver frame with SeqNum: 5
[INFO] - [SWP Switch 2 Port: 0] send ACK LFR: 5
[INFO] - [Switch Switch 2] Broadcast MAC sent throw all interfaces
[INFO] - [Switch Switch 2] send frame for MAC: ff:ff:ff:ff:ff:ff throw interface 1
[INFO] - [Switch Switch 2] send frame for MAC: ff:ff:ff:ff:ff:ff throw interface 2
[INFO] - [SWP Host 2 NET 0] swp_protocol::processFrame
[INFO] - [SWP Switch 1 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 1 Port: 2] Deliver frame with SeqNum: 4
[INFO] - [SWP Switch 1 Port: 2] send ACK LFR: 4
[INFO] - [Switch Switch 1] Broadcast MAC sent throw all interfaces
[INFO] - [Switch Switch 1] send frame for MAC: ff:ff:ff:ff:ff:ff throw interface 0
[INFO] - [SWP Switch 2 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 2] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 3 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 3 Port: 0] Deliver frame with SeqNum: 5
[INFO] - [SWP Switch 3 Port: 0] send ACK LFR: 5
[INFO] - [SWP Switch 2 Port: 1] swp_protocol::processFrame
[INFO] - [Switch Switch 3] Broadcast MAC sent throw all interfaces
[INFO] - [Switch Switch 3] send frame for MAC: ff:ff:ff:ff:ff:ff throw interface 2
[INFO] - [SWP Host 1 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 1 NET 0] Deliver frame with SeqNum: 4
[INFO] - [SWP Host 1 NET 0] send ACK LFR: 4
[INFO] - [Link Host 1 NET 0] Process ARP packet.
[INFO] - [SWP Router 1 NET 0] swp_protocol::processFrame
[INFO] - [SWP Router 1 NET 0] Deliver frame with SeqNum: 5
[INFO] - [SWP Router 1 NET 0] send ACK LFR: 5
[INFO] - [Link Router 1 NET 0] Process ARP packet.
[INFO] - [SWP Switch 2 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 1 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Host 1 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 1 NET 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 3 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Router 1 NET 0] swp_protocol::processFrame
[INFO] - [SWP Router 1 NET 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 3 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 3 Port: 2] Deliver frame with SeqNum: 4
[INFO] - [SWP Switch 3 Port: 2] send ACK LFR: 4
[INFO] - [Switch Switch 3] send frame for MAC: 0:0:0:0:0:2 throw interface 0
[INFO] - [SWP Router 1 NET 0] swp_protocol::processFrame
[INFO] - [SWP Switch 3 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 3 Port: 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 2 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 2] Deliver frame with SeqNum: 4
[INFO] - [SWP Switch 2 Port: 2] send ACK LFR: 4
[INFO] - [SWP Switch 3 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 3 Port: 2] Discarded frame: Frame not in RWS windows
```

```

[INFO] - [Switch Switch 2] send frame for MAC: 0:0:0:0:0:2 throw interface 0
[INFO] - [SWP Host 2 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 2 NET 0] Deliver frame with SeqNum: 6
[INFO] - [SWP Host 2 NET 0] send ACK LFR: 6
[INFO] - [Link Host 2 NET 0] Process ARP packet.
[INFO] - [Link Host 2 NET 0] Send control ARP_READY
[INFO] - [IPv4 Host 2] link::Ctrl::ARP_READY
[INFO] - [Link Host 2 NET 0] SEND_PACKET
[INFO] - [Link Host 2 NET 0] ARP ip 1.0.0.0 is in cache.
[INFO] - [SWP Switch 3 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Host 2 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 2 NET 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 2 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 0] Deliver frame with SeqNum: 6
[INFO] - [SWP Switch 2 Port: 0] send ACK LFR: 6
[INFO] - [Switch Switch 2] send frame for MAC: 0:0:0:0:0:3 throw interface 2
[INFO] - [SWP Host 2 NET 0] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 2] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 3 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 3 Port: 0] Deliver frame with SeqNum: 6
[INFO] - [SWP Switch 3 Port: 0] send ACK LFR: 6
[INFO] - [Switch Switch 3] send frame for MAC: 0:0:0:0:0:3 throw interface 2
[INFO] - [SWP Router 1 NET 0] swp_protocol::processFrame
[INFO] - [SWP Router 1 NET 0] Deliver frame with SeqNum: 6
[INFO] - [SWP Router 1 NET 0] send ACK LFR: 6
[INFO] - [IPv4 Router 1] Process Ip Datagram
[INFO] - [IPv4 Router 1] Valid checksum: true
[INFO] - [IPv4 Router 1] Forwarding ip packet: 5.0.0.2
[INFO] - [IPv4 Router 1] Best route for packet with dest_ip: 5.0.0.2
is : | 5.0.0.0 | 255.0.0.0 | 3.0.0.1 | 2 | Router 2 |
[INFO] - [IPv4 Router 1] Added packet to wait ARP for nexthop: 3.0.0.1
[INFO] - [IPv4 Router 1] ARP query throw interface: NET 1
[INFO] - [Link Router 1 NET 1] ARP_QUERY
[INFO] - [Link Router 1 NET 1] ARP ip 3.0.0.1 not cached.
[INFO] - [Link Router 1 NET 1] Sending ARP query for ip: 3.0.0.1
[INFO] - [SWP Router 2 NET 0] swp_protocol::processFrame
[INFO] - [SWP Router 2 NET 0] Deliver frame with SeqNum: 4
[INFO] - [SWP Router 2 NET 0] send ACK LFR: 4
[INFO] - [Link Router 2 NET 0] Process ARP packet.
[INFO] - [SWP Router 1 NET 1] swp_protocol::processFrame
[INFO] - [SWP Router 1 NET 1] swp_protocol::processFrame
[INFO] - [SWP Router 1 NET 1] Deliver frame with SeqNum: 4
[INFO] - [SWP Router 1 NET 1] send ACK LFR: 4
[INFO] - [Link Router 1 NET 1] Process ARP packet.
[INFO] - [Link Router 1 NET 1] Send control ARP_READY
[INFO] - [SWP Router 2 NET 0] swp_protocol::processFrame
[INFO] - [IPv4 Router 1] link::Ctrl::ARP_READY
[INFO] - [Link Router 1 NET 1] SEND_PACKET
[INFO] - [Link Router 1 NET 1] ARP ip 3.0.0.1 is in cache.
[INFO] - [SWP Router 2 NET 0] swp_protocol::processFrame
[INFO] - [SWP Router 2 NET 0] Deliver frame with SeqNum: 5
[INFO] - [SWP Router 2 NET 0] send ACK LFR: 5
[INFO] - [SWP Router 1 NET 1] swp_protocol::processFrame
[INFO] - [IPv4 Router 2] Process Ip Datagram
[INFO] - [IPv4 Router 2] Valid checksum: true
[INFO] - [IPv4 Router 2] Forwarding ip packet: 5.0.0.2
[INFO] - [IPv4 Router 2] Best route for packet with dest_ip: 5.0.0.2
is : | 5.0.0.2 | 255.255.255.255 | 5.0.0.2 | 1 | Host 4 |
[INFO] - [IPv4 Router 2] Added packet to wait ARP for nexthop: 5.0.0.2
[INFO] - [IPv4 Router 2] ARP query throw interface: NET 1
[INFO] - [Link Router 2 NET 1] ARP_QUERY
[INFO] - [Link Router 2 NET 1] ARP ip 5.0.0.2 not cached.
[INFO] - [Link Router 2 NET 1] Sending ARP query for ip: 5.0.0.2
[INFO] - [SWP Switch 4 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 4 Port: 0] Deliver frame with SeqNum: 4

```

```

[INFO] - [SWP Switch 4 Port: 0] send ACK LFR: 4
[INFO] - [Switch Switch 4] Broadcast MAC sent throw all interfaces
[INFO] - [Switch Switch 4] send frame for MAC: ff:ff:ff:ff:ff:ff throw interface 1
[INFO] - [Switch Switch 4] send frame for MAC: ff:ff:ff:ff:ff:ff throw interface 2
[INFO] - [SWP Router 2 NET 1] swp_protocol::processFrame
[INFO] - [SWP Host 5 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 5 NET 0] Deliver frame with SeqNum: 4
[INFO] - [SWP Host 5 NET 0] send ACK LFR: 4
[INFO] - [Link Host 5 NET 0] Process ARP packet.
[INFO] - [SWP Switch 4 Port: 1] swp_protocol::processFrame
[INFO] - [SWP Host 4 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 4 NET 0] Deliver frame with SeqNum: 3
[INFO] - [SWP Host 4 NET 0] send ACK LFR: 3
[INFO] - [Link Host 4 NET 0] Process ARP packet.
[INFO] - [SWP Switch 2 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 3 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Router 1 NET 0] swp_protocol::processFrame
[INFO] - [SWP Router 1 NET 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 3 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 3 Port: 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 4 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Host 4 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 4 NET 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 4 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 4 Port: 2] Deliver frame with SeqNum: 1
[INFO] - [SWP Switch 4 Port: 2] send ACK LFR: 1
[INFO] - [Switch Switch 4] send frame for MAC: 0:0:0:0:0:10 throw interface 0
[INFO] - [SWP Host 4 NET 0] swp_protocol::processFrame
[INFO] - [SWP Switch 4 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 4 Port: 2] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Router 2 NET 1] swp_protocol::processFrame
[INFO] - [SWP Router 2 NET 1] Deliver frame with SeqNum: 4
[INFO] - [SWP Router 2 NET 1] send ACK LFR: 4
[INFO] - [Link Router 2 NET 1] Process ARP packet.
[INFO] - [Link Router 2 NET 1] Send control ARP_READY
[INFO] - [IPv4 Router 2] link::Ctrl::ARP_READY
[INFO] - [Link Router 2 NET 1] SEND_PACKET
[INFO] - [Link Router 2 NET 1] ARP ip 5.0.0.2 is in cache.
[INFO] - [SWP Switch 4 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Router 2 NET 1] swp_protocol::processFrame
[INFO] - [SWP Router 2 NET 1] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 4 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 4 Port: 0] Deliver frame with SeqNum: 5
[INFO] - [SWP Switch 4 Port: 0] send ACK LFR: 5
[INFO] - [Switch Switch 4] send frame for MAC: 0:0:0:0:0:12 throw interface 2
[INFO] - [SWP Router 2 NET 1] swp_protocol::processFrame
[INFO] - [SWP Host 4 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 4 NET 0] Deliver frame with SeqNum: 4
[INFO] - [SWP Host 4 NET 0] send ACK LFR: 4
[INFO] - [IPv4 Host 4] Processing arrived ip packet: 5.0.0.2
[INFO] - [IPv4 Host 4] Valid checksum: true
[INFO] - [UDP Host 4] process Segment:
[INFO] - [UDP Host 4] local_port: 53
[INFO] - [UDP Host 4] remote_port: 53
[INFO] - [UDP Host 4] local_ip: 5.0.0.2
[INFO] - [UDP Host 4] remote_ip: 1.0.0.2
[INFO] - [UDP Host 4] valid checksum: true
[INFO] - [UDP Host 4] socket local_ip: 5.0.0.2
[INFO] - [UDP Host 4] socket local_ip: 53
[INFO] - [UDP Host 4] Deliver dns packet
[INFO] - [DNS Host 4] Incoming packet:
[INFO] - [DNS Host 4] dns packet
** HEADER **
id: 0
flags_code: 33024
QDCount: 1

```

```

ANCount: 0
NSCount: 0
ARCount: 1

** QUESTION SECTION **
network.devs.com A 0.0.0.0 IN 0

** ANSWER SECTION **

** AUTHORITATIVES SECTION **

** ADDITIONAL SECTION **
network.devs.com A 1.0.0.2 IN 9227
_____
[INFO] - [DNS Host 4] Query packet
[INFO] - [DNS Host 4] No recursive query for domain network.devs.com
[INFO] - [DNS Host 4] Zone found:
_____
** NS **
devs.com NS ns1.devs.com IN 65535
** A **
ns1.dev.com A 4.0.0.1 IN 65535
_____
[INFO] - [DNS Host 4] Send DNS Query to: (1.0.0.2, 53)
[INFO] - [UDP Host 4] Write_to/Send_to
[INFO] - [UDP Host 4] Read/Recv
[INFO] - [IPv4 Host 4] Forwarding ip packet: 1.0.0.2
[INFO] - [IPv4 Host 4] Best route for packet with dest_ip: 1.0.0.2
is: | 0.0.0.0 | 0.0.0.0 | 5.0.0.0 | 1 | Default Route |
[INFO] - [IPv4 Host 4] Added packet to wait ARP for nexthop: 5.0.0.0
[INFO] - [IPv4 Host 4] ARP query throw interface: NET 0
[INFO] - [Link Host 4 NET 0] ARP_QUERY
[INFO] - [Link Host 4 NET 0] ARP ip 5.0.0.0 not cached.
[INFO] - [Link Host 4 NET 0] Sending ARP query for ip: 5.0.0.0
[INFO] - [SWP Switch 4 Port: 2] swp-protocol::processFrame
[INFO] - [SWP Host 4 NET 0] swp-protocol::processFrame
[INFO] - [SWP Host 4 NET 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 4 Port: 2] swp-protocol::processFrame
[INFO] - [SWP Switch 4 Port: 2] Deliver frame with SeqNum: 2
[INFO] - [SWP Switch 4 Port: 2] send ACK LFR: 2
[INFO] - [Switch Switch 4] Broadcast MAC sent throw all interfaces
[INFO] - [Switch Switch 4] send frame for MAC: ff:ff:ff:ff:ff:ff throw interface 0
[INFO] - [Switch Switch 4] send frame for MAC: ff:ff:ff:ff:ff:ff throw interface 1
[INFO] - [SWP Host 4 NET 0] swp-protocol::processFrame
[INFO] - [SWP Router 2 NET 1] swp-protocol::processFrame
[INFO] - [SWP Router 2 NET 1] Deliver frame with SeqNum: 5
[INFO] - [SWP Router 2 NET 1] send ACK LFR: 5
[INFO] - [Link Router 2 NET 1] Process ARP packet.
[INFO] - [SWP Switch 4 Port: 0] swp-protocol::processFrame
[INFO] - [SWP Host 5 NET 0] swp-protocol::processFrame
[INFO] - [SWP Host 5 NET 0] Deliver frame with SeqNum: 5
[INFO] - [SWP Host 5 NET 0] send ACK LFR: 5
[INFO] - [Link Host 5 NET 0] Process ARP packet.
[INFO] - [SWP Switch 4 Port: 1] swp-protocol::processFrame
[INFO] - [SWP Host 5 NET 0] swp-protocol::processFrame
[INFO] - [SWP Host 5 NET 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 4 Port: 0] swp-protocol::processFrame
[INFO] - [SWP Switch 4 Port: 0] Deliver frame with SeqNum: 6
[INFO] - [SWP Switch 4 Port: 0] send ACK LFR: 6
[INFO] - [Switch Switch 4] send frame for MAC: 0:0:0:0:0:12 throw interface 2
[INFO] - [SWP Router 2 NET 1] swp-protocol::processFrame
[INFO] - [SWP Switch 4 Port: 0] swp-protocol::processFrame
[INFO] - [SWP Switch 4 Port: 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Host 4 NET 0] swp-protocol::processFrame
[INFO] - [SWP Host 4 NET 0] Deliver frame with SeqNum: 5
[INFO] - [SWP Host 4 NET 0] send ACK LFR: 5
[INFO] - [Link Host 4 NET 0] Process ARP packet.

```

```

[INFO] - [Link Host 4 NET 0] Send control ARP_READY
[INFO] - [IPv4 Host 4] link::Ctrl::ARP_READY
[INFO] - [Link Host 4 NET 0] SEND_PACKET
[INFO] - [Link Host 4 NET 0] ARP ip 5.0.0.0 is in cache.
[INFO] - [SWP Switch 4 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Host 4 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 4 NET 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 4 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 4 Port: 2] Deliver frame with SeqNum: 3
[INFO] - [SWP Switch 4 Port: 2] send ACK LFR: 3
[INFO] - [Switch Switch 4] send frame for MAC: 0:0:0:0:0:10 throw interface 0
[INFO] - [SWP Host 4 NET 0] swp_protocol::processFrame
[INFO] - [SWP Router 2 NET 1] swp_protocol::processFrame
[INFO] - [SWP Router 2 NET 1] Deliver frame with SeqNum: 6
[INFO] - [SWP Router 2 NET 1] send ACK LFR: 6
[INFO] - [IPv4 Router 2] Process Ip Datagram
[INFO] - [IPv4 Router 2] Valid checksum: true
[INFO] - [IPv4 Router 2] Forwarding ip packet: 1.0.0.2
[INFO] - [IPv4 Router 2] Best route for packet with dest_ip: 1.0.0.2
is : | 1.0.0.0 | 255.0.0.0 | 3.0.0.0 | 2 | Router 1 |
[INFO] - [IPv4 Router 2] Added packet to wait ARP for nexthop: 3.0.0.0
[INFO] - [IPv4 Router 2] ARP query throw interface: NET 0
[INFO] - [Link Router 2 NET 0] ARP_QUERY
[INFO] - [Link Router 2 NET 0] ARP ip 3.0.0.0 not cached.
[INFO] - [Link Router 2 NET 0] Sending ARP query for ip: 3.0.0.0
[INFO] - [SWP Router 1 NET 1] swp_protocol::processFrame
[INFO] - [SWP Router 1 NET 1] Deliver frame with SeqNum: 5
[INFO] - [SWP Router 1 NET 1] send ACK LFR: 5
[INFO] - [Link Router 1 NET 1] Process ARP packet.
[INFO] - [SWP Router 2 NET 0] swp_protocol::processFrame
[INFO] - [SWP Router 2 NET 0] swp_protocol::processFrame
[INFO] - [SWP Router 2 NET 0] Deliver frame with SeqNum: 6
[INFO] - [SWP Router 2 NET 0] send ACK LFR: 6
[INFO] - [Link Router 2 NET 0] Process ARP packet.
[INFO] - [Link Router 2 NET 0] Send control ARP_READY
[INFO] - [SWP Router 1 NET 1] swp_protocol::processFrame
[INFO] - [IPv4 Router 2] link::Ctrl::ARP_READY
[INFO] - [Link Router 2 NET 0] SEND_PACKET
[INFO] - [Link Router 2 NET 0] ARP ip 3.0.0.0 is in cache.
[INFO] - [SWP Router 1 NET 1] swp_protocol::processFrame
[INFO] - [SWP Router 1 NET 1] Deliver frame with SeqNum: 6
[INFO] - [SWP Router 1 NET 1] send ACK LFR: 6
[INFO] - [SWP Router 2 NET 0] swp_protocol::processFrame
[INFO] - [IPv4 Router 1] Process Ip Datagram
[INFO] - [IPv4 Router 1] Valid checksum: true
[INFO] - [IPv4 Router 1] Forwarding ip packet: 1.0.0.2
[INFO] - [IPv4 Router 1] Best route for packet with dest_ip: 1.0.0.2
is : | 1.0.0.2 | 255.255.255.255 | 1.0.0.2 | 1 | Host 2 |
[INFO] - [IPv4 Router 1] Added packet to wait ARP for nexthop: 1.0.0.2
[INFO] - [IPv4 Router 1] ARP query throw interface: NET 0
[INFO] - [Link Router 1 NET 0] ARP_QUERY
[INFO] - [Link Router 1 NET 0] ARP ip 1.0.0.2 not cached.
[INFO] - [Link Router 1 NET 0] Sending ARP query for ip: 1.0.0.2
[INFO] - [SWP Switch 3 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 3 Port: 2] Deliver frame with SeqNum: 5
[INFO] - [SWP Switch 3 Port: 2] send ACK LFR: 5
[INFO] - [Switch Switch 3] Broadcast MAC sent throw all interfaces
[INFO] - [Switch Switch 3] send frame for MAC: ff:ff:ff:ff:ff:ff throw interface 0
[INFO] - [SWP Router 1 NET 0] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 2] Deliver frame with SeqNum: 5
[INFO] - [SWP Switch 2 Port: 2] send ACK LFR: 5
[INFO] - [Switch Switch 2] Broadcast MAC sent throw all interfaces
[INFO] - [Switch Switch 2] send frame for MAC: ff:ff:ff:ff:ff:ff throw interface 0
[INFO] - [Switch Switch 2] send frame for MAC: ff:ff:ff:ff:ff:ff throw interface 1
[INFO] - [SWP Host 2 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 2 NET 0] Deliver frame with SeqNum: 7

```

```

[INFO] - [SWP Host 2 NET 0] send ACK LFR: 7
[INFO] - [SWP Switch 1 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 1 Port: 2] Deliver frame with SeqNum: 5
[INFO] - [SWP Switch 1 Port: 2] send ACK LFR: 5
[INFO] - [Link Host 2 NET 0] Process ARP packet.
[INFO] - [Switch Switch 1] Broadcast MAC sent throw all interfaces
[INFO] - [Switch Switch 1] send frame for MAC: ff:ff:ff:ff:ff:ff throw interface 0
[INFO] - [SWP Host 1 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 1 NET 0] Deliver frame with SeqNum: 5
[INFO] - [SWP Host 1 NET 0] send ACK LFR: 5
[INFO] - [Link Host 1 NET 0] Process ARP packet.
[INFO] - [SWP Switch 4 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Router 2 NET 1] swp_protocol::processFrame
[INFO] - [SWP Router 2 NET 1] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 3 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Host 2 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 2 NET 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 2 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 1] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 0] Deliver frame with SeqNum: 7
[INFO] - [SWP Switch 2 Port: 0] send ACK LFR: 7
[INFO] - [Switch Switch 2] send frame for MAC: 0:0:0:0:0:3 throw interface 2
[INFO] - [SWP Host 2 NET 0] swp_protocol::processFrame
[INFO] - [SWP Switch 1 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Host 1 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 1 NET 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 1 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 1 Port: 2] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 2 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 2] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 3 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 3 Port: 0] Deliver frame with SeqNum: 7
[INFO] - [SWP Switch 3 Port: 0] send ACK LFR: 7
[INFO] - [SWP Switch 2 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 0] Discarded frame: Frame not in RWS windows
[INFO] - [Switch Switch 3] send frame for MAC: 0:0:0:0:0:3 throw interface 2
[INFO] - [SWP Router 1 NET 0] swp_protocol::processFrame
[INFO] - [SWP Router 1 NET 0] Deliver frame with SeqNum: 7
[INFO] - [SWP Router 1 NET 0] send ACK LFR: 7
[INFO] - [Link Router 1 NET 0] Process ARP packet.
[INFO] - [Link Router 1 NET 0] Send control ARP_READY
[INFO] - [IPv4 Router 1] link::Ctrl::ARP_READY
[INFO] - [Link Router 1 NET 0] SEND_PACKET
[INFO] - [Link Router 1 NET 0] ARP ip 1.0.0.2 is in cache.
[INFO] - [SWP Switch 2 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 3 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Router 1 NET 0] swp_protocol::processFrame
[INFO] - [SWP Router 1 NET 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 3 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 3 Port: 2] Deliver frame with SeqNum: 6
[INFO] - [SWP Switch 3 Port: 2] send ACK LFR: 6
[INFO] - [Switch Switch 3] send frame for MAC: 0:0:0:0:0:2 throw interface 0
[INFO] - [SWP Router 1 NET 0] swp_protocol::processFrame
[INFO] - [SWP Switch 3 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 3 Port: 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 2 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 2] Deliver frame with SeqNum: 6
[INFO] - [SWP Switch 2 Port: 2] send ACK LFR: 6
[INFO] - [Switch Switch 2] send frame for MAC: 0:0:0:0:0:2 throw interface 0
[INFO] - [SWP Host 2 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 2 NET 0] Deliver frame with SeqNum: 8
[INFO] - [SWP Host 2 NET 0] send ACK LFR: 8
[INFO] - [IPv4 Host 2] Processing arrived ip packet: 1.0.0.2
[INFO] - [IPv4 Host 2] Valid checksum: true
[INFO] - [UDP Host 2] process Segment:
[INFO] - [UDP Host 2] local_port: 53

```

```

[INFO] - [UDP Host 2] remote_port: 53
[INFO] - [UDP Host 2] local_ip: 1.0.0.2
[INFO] - [UDP Host 2] remote_ip: 5.0.0.2
[INFO] - [UDP Host 2] valid checksum: true
[INFO] - [UDP Host 2] socket local_ip: 1.0.0.2
[INFO] - [UDP Host 2] socket local_ip: 53
[INFO] - [UDP Host 2] Deliver dns packet
[INFO] - [DNS Host 2] Incoming packet:
[INFO] - [DNS Host 2] dns packet


---


** HEADER **
id: 0
flags_code: 256
QDCount: 1
ANCount: 1
NSCount: 1
ARCount: 1

** QUESTION SECTION **
network.devs.com A 0.0.0.0 IN 0

** ANSWER SECTION **
devs.com NS ns1.devs.com IN 65535

** AUTHORITATIVES SECTION **
ns1.dev.com A 4.0.0.1 IN 65535

** ADDITIONAL SECTION **
network.devs.com A 5.0.0.2 IN 9227


---


[INFO] - [DNS Host 2] Answer packet
[INFO] - [DNS Host 2] Answer Type: NS
[INFO] - [DNS Host 2] Send DNS Query to: (4.0.0.1, 53)
[INFO] - [UDP Host 2] Write_to/Send_to
[INFO] - [UDP Host 2] Read/Recv
[INFO] - [IPv4 Host 2] Forwarding ip packet: 4.0.0.1
[INFO] - [IPv4 Host 2] Best route for packet with dest_ip: 4.0.0.1
is: | 0.0.0.0 | 0.0.0.0 | 1.0.0.0 | 1 | Default Route |
[INFO] - [IPv4 Host 2] Added packet to wait ARP for nexthop: 1.0.0.0
[INFO] - [IPv4 Host 2] ARP query throw interface: NET 0
[INFO] - [Link Host 2 NET 0] ARP_QUERY
[INFO] - [Link Host 2 NET 0] ARP ip 1.0.0.0 not cached.
[INFO] - [Link Host 2 NET 0] Sending ARP query for ip: 1.0.0.0
[INFO] - [SWP Switch 3 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Host 2 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 2 NET 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 2 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 0] Deliver frame with SeqNum: 8
[INFO] - [SWP Switch 2 Port: 0] send ACK LFR: 8
[INFO] - [Switch Switch 2] Broadcast MAC sent throw all interfaces
[INFO] - [Switch Switch 2] send frame for MAC: ff:ff:ff:ff:ff:ff throw interface 1
[INFO] - [Switch Switch 2] send frame for MAC: ff:ff:ff:ff:ff:ff throw interface 2
[INFO] - [SWP Host 2 NET 0] swp_protocol::processFrame
[INFO] - [SWP Switch 1 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 1 Port: 2] Deliver frame with SeqNum: 6
[INFO] - [SWP Switch 1 Port: 2] send ACK LFR: 6
[INFO] - [Switch Switch 1] Broadcast MAC sent throw all interfaces
[INFO] - [Switch Switch 1] send frame for MAC: ff:ff:ff:ff:ff:ff throw interface 0
[INFO] - [SWP Switch 2 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 2] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 3 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 3 Port: 0] Deliver frame with SeqNum: 8
[INFO] - [SWP Switch 3 Port: 0] send ACK LFR: 8
[INFO] - [SWP Switch 2 Port: 1] swp_protocol::processFrame
[INFO] - [Switch Switch 3] Broadcast MAC sent throw all interfaces
[INFO] - [Switch Switch 3] send frame for MAC: ff:ff:ff:ff:ff:ff throw interface 2

```

```

[INFO] - [SWP Host 1 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 1 NET 0] Deliver frame with SeqNum: 6
[INFO] - [SWP Host 1 NET 0] send ACK LFR: 6
[INFO] - [Link Host 1 NET 0] Process ARP packet.
[INFO] - [SWP Router 1 NET 0] swp_protocol::processFrame
[INFO] - [SWP Router 1 NET 0] Deliver frame with SeqNum: 8
[INFO] - [SWP Router 1 NET 0] send ACK LFR: 8
[INFO] - [Link Router 1 NET 0] Process ARP packet.
[INFO] - [SWP Switch 2 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 1 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Host 1 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 1 NET 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 3 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Router 1 NET 0] swp_protocol::processFrame
[INFO] - [SWP Router 1 NET 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 3 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 3 Port: 2] Deliver frame with SeqNum: 7
[INFO] - [SWP Switch 3 Port: 2] send ACK LFR: 7
[INFO] - [Switch Switch 3] send frame for MAC: 0:0:0:0:0:2 throw interface 0
[INFO] - [SWP Router 1 NET 0] swp_protocol::processFrame
[INFO] - [SWP Switch 3 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 3 Port: 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 2 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 2] Deliver frame with SeqNum: 7
[INFO] - [SWP Switch 2 Port: 2] send ACK LFR: 7
[INFO] - [SWP Switch 3 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 3 Port: 2] Discarded frame: Frame not in RWS windows
[INFO] - [Switch Switch 2] send frame for MAC: 0:0:0:0:0:2 throw interface 0
[INFO] - [SWP Host 2 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 2 NET 0] Deliver frame with SeqNum: 9
[INFO] - [SWP Host 2 NET 0] send ACK LFR: 9
[INFO] - [Link Host 2 NET 0] Process ARP packet.
[INFO] - [Link Host 2 NET 0] Send control ARP_READY
[INFO] - [IPv4 Host 2] link::Ctrl::ARP_READY
[INFO] - [Link Host 2 NET 0] SEND_PACKET
[INFO] - [Link Host 2 NET 0] ARP ip 1.0.0.0 is in cache.
[INFO] - [SWP Switch 3 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Host 2 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 2 NET 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 2 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 0] Deliver frame with SeqNum: 9
[INFO] - [SWP Switch 2 Port: 0] send ACK LFR: 9
[INFO] - [Switch Switch 2] send frame for MAC: 0:0:0:0:0:3 throw interface 2
[INFO] - [SWP Host 2 NET 0] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 2] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 3 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 3 Port: 0] Deliver frame with SeqNum: 9
[INFO] - [SWP Switch 3 Port: 0] send ACK LFR: 9
[INFO] - [Switch Switch 3] send frame for MAC: 0:0:0:0:0:3 throw interface 2
[INFO] - [SWP Router 1 NET 0] swp_protocol::processFrame
[INFO] - [SWP Router 1 NET 0] Deliver frame with SeqNum: 9
[INFO] - [SWP Router 1 NET 0] send ACK LFR: 9
[INFO] - [IPv4 Router 1] Process Ip Datagram
[INFO] - [IPv4 Router 1] Valid checksum: true
[INFO] - [IPv4 Router 1] Forwarding ip packet: 4.0.0.1
[INFO] - [IPv4 Router 1] Best route for packet with dest_ip: 4.0.0.1
is: | 4.0.0.0 | 255.0.0.0 | 2.0.0.1 | 2 | Router 3 |
[INFO] - [IPv4 Router 1] Added packet to wait ARP for nexthop: 2.0.0.1
[INFO] - [IPv4 Router 1] ARP query throw interface: NET 2
[INFO] - [Link Router 1 NET 2] ARP_QUERY
[INFO] - [Link Router 1 NET 2] ARP ip 2.0.0.1 not cached.
[INFO] - [Link Router 1 NET 2] Sending ARP query for ip: 2.0.0.1
[INFO] - [SWP Router 3 NET 0] swp_protocol::processFrame
[INFO] - [SWP Router 3 NET 0] Deliver frame with SeqNum: 1
[INFO] - [SWP Router 3 NET 0] send ACK LFR: 1

```

```

[INFO] - [Link Router 3 NET 0] Process ARP packet.
[INFO] - [SWP Router 1 NET 2] swp_protocol::processFrame
[INFO] - [SWP Router 1 NET 2] swp_protocol::processFrame
[INFO] - [SWP Router 1 NET 2] Deliver frame with SeqNum: 1
[INFO] - [SWP Router 1 NET 2] send ACK LFR: 1
[INFO] - [Link Router 1 NET 2] Process ARP packet.
[INFO] - [Link Router 1 NET 2] Send control ARP_READY
[INFO] - [SWP Router 3 NET 0] swp_protocol::processFrame
[INFO] - [IPv4 Router 1] link::Ctrl::ARP_READY
[INFO] - [Link Router 1 NET 2] SEND_PACKET
[INFO] - [Link Router 1 NET 2] ARP ip 2.0.0.1 is in cache.
[INFO] - [SWP Router 3 NET 0] swp_protocol::processFrame
[INFO] - [SWP Router 3 NET 0] Deliver frame with SeqNum: 2
[INFO] - [SWP Router 3 NET 0] send ACK LFR: 2
[INFO] - [SWP Router 1 NET 2] swp_protocol::processFrame
[INFO] - [IPv4 Router 3] Process Ip Datagram
[INFO] - [IPv4 Router 3] Valid checksum: true
[INFO] - [IPv4 Router 3] Forwarding ip packet: 4.0.0.1
[INFO] - [IPv4 Router 3] Best route for packet with dest_ip: 4.0.0.1
is : | 4.0.0.1 | 255.255.255.255 | 4.0.0.1 | 1 | Host 3 |
[INFO] - [IPv4 Router 3] Added packet to wait ARP for nexthop: 4.0.0.1
[INFO] - [IPv4 Router 3] ARP query throw interface: NET 1
[INFO] - [Link Router 3 NET 1] ARP_QUERY
[INFO] - [Link Router 3 NET 1] ARP ip 4.0.0.1 not cached.
[INFO] - [Link Router 3 NET 1] Sending ARP query for ip: 4.0.0.1
[INFO] - [SWP Host 3 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 3 NET 0] Deliver frame with SeqNum: 1
[INFO] - [SWP Host 3 NET 0] send ACK LFR: 1
[INFO] - [Link Host 3 NET 0] Process ARP packet.
[INFO] - [SWP Router 3 NET 1] swp_protocol::processFrame
[INFO] - [SWP Router 3 NET 1] swp_protocol::processFrame
[INFO] - [SWP Router 3 NET 1] Deliver frame with SeqNum: 1
[INFO] - [SWP Router 3 NET 1] send ACK LFR: 1
[INFO] - [Link Router 3 NET 1] Process ARP packet.
[INFO] - [Link Router 3 NET 1] Send control ARP_READY
[INFO] - [SWP Host 3 NET 0] swp_protocol::processFrame
[INFO] - [IPv4 Router 3] link::Ctrl::ARP_READY
[INFO] - [Link Router 3 NET 1] SEND_PACKET
[INFO] - [Link Router 3 NET 1] ARP ip 4.0.0.1 is in cache.
[INFO] - [SWP Host 3 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 3 NET 0] Deliver frame with SeqNum: 2
[INFO] - [SWP Host 3 NET 0] send ACK LFR: 2
[INFO] - [SWP Router 3 NET 1] swp_protocol::processFrame
[INFO] - [IPv4 Host 3] Processing arrived ip packet: 4.0.0.1
[INFO] - [IPv4 Host 3] Valid checksum: true
[INFO] - [UDP Host 3] process Segment:
[INFO] - [UDP Host 3] local_port: 53
[INFO] - [UDP Host 3] remote_port: 53
[INFO] - [UDP Host 3] local_ip: 4.0.0.1
[INFO] - [UDP Host 3] remote_ip: 1.0.0.2
[INFO] - [UDP Host 3] valid checksum: true
[INFO] - [UDP Host 3] socket local_ip: 4.0.0.1
[INFO] - [UDP Host 3] socket local_ip: 53
[INFO] - [UDP Host 3] Deliver dns packet
[INFO] - [DNS Host 3] Incoming packet:
[INFO] - [DNS Host 3] dns packet _____
** HEADER **
id: 0
flags_code: 33024
QDCount: 1
ANCount: 0
NSCount: 0
ARCount: 1

** QUESTION SECTION **
network.devs.com A 0.0.0.0 IN 0

```

```
** ANSWER SECTION **
```

```
** AUTHORITATIVES SECTION **
```

```
** ADDITIONAL SECTION **
```

```
network.devs.com A 1.0.0.2 IN 9227
```

```
[INFO] - [DNS Host 3] Query packet
[INFO] - [DNS Host 3] Resource for domain network.devs.com exist
[INFO] - [DNS Host 3] Send DNS Query to: (1.0.0.2, 53)
[INFO] - [UDP Host 3] Write_to/Send_to
[INFO] - [UDP Host 3] Read/Recv
[INFO] - [IPv4 Host 3] Forwarding ip packet: 1.0.0.2
[INFO] - [IPv4 Host 3] Best route for packet with dest_ip: 1.0.0.2
is: | 0.0.0.0 | 0.0.0.0 | 4.0.0.0 | 1 | Default Route |
[INFO] - [IPv4 Host 3] Added packet to wait ARP for nexthop: 4.0.0.0
[INFO] - [IPv4 Host 3] ARP query throw interface: NET 0
[INFO] - [Link Host 3 NET 0] ARP_QUERY
[INFO] - [Link Host 3 NET 0] ARP ip 4.0.0.0 is in cache.
[INFO] - [Link Host 3 NET 0] Send control ARP_READY
[INFO] - [IPv4 Host 3] link::Ctrl::ARP_READY
[INFO] - [Link Host 3 NET 0] SEND_PACKET
[INFO] - [Link Host 3 NET 0] ARP ip 4.0.0.0 is in cache.
[INFO] - [SWP Router 3 NET 1] swp_protocol::processFrame
[INFO] - [SWP Router 3 NET 1] Deliver frame with SeqNum: 2
[INFO] - [SWP Router 3 NET 1] send ACK LFR: 2
[INFO] - [SWP Host 3 NET 0] swp_protocol::processFrame
[INFO] - [IPv4 Router 3] Process Ip Datagram
[INFO] - [IPv4 Router 3] Valid checksum: true
[INFO] - [IPv4 Router 3] Forwarding ip packet: 1.0.0.2
[INFO] - [IPv4 Router 3] Best route for packet with dest_ip: 1.0.0.2
is: | 1.0.0.0 | 255.0.0.0 | 2.0.0.0 | 2 | Router 1 |
[INFO] - [IPv4 Router 3] Added packet to wait ARP for nexthop: 2.0.0.0
[INFO] - [IPv4 Router 3] ARP query throw interface: NET 0
[INFO] - [Link Router 3 NET 0] ARP_QUERY
[INFO] - [Link Router 3 NET 0] ARP ip 2.0.0.0 is in cache.
[INFO] - [Link Router 3 NET 0] Send control ARP_READY
[INFO] - [IPv4 Router 3] link::Ctrl::ARP_READY
[INFO] - [Link Router 3 NET 0] SEND_PACKET
[INFO] - [Link Router 3 NET 0] ARP ip 2.0.0.0 is in cache.
[INFO] - [SWP Router 1 NET 2] swp_protocol::processFrame
[INFO] - [SWP Router 1 NET 2] Deliver frame with SeqNum: 2
[INFO] - [SWP Router 1 NET 2] send ACK LFR: 2
[INFO] - [SWP Router 3 NET 0] swp_protocol::processFrame
[INFO] - [IPv4 Router 1] Process Ip Datagram
[INFO] - [IPv4 Router 1] Valid checksum: true
[INFO] - [IPv4 Router 1] Forwarding ip packet: 1.0.0.2
[INFO] - [IPv4 Router 1] Best route for packet with dest_ip: 1.0.0.2
is: | 1.0.0.2 | 255.255.255.255 | 1.0.0.2 | 1 | Host 2 |
[INFO] - [IPv4 Router 1] Added packet to wait ARP for nexthop: 1.0.0.2
[INFO] - [IPv4 Router 1] ARP query throw interface: NET 0
[INFO] - [Link Router 1 NET 0] ARP_QUERY
[INFO] - [Link Router 1 NET 0] ARP ip 1.0.0.2 not cached.
[INFO] - [Link Router 1 NET 0] Sending ARP query for ip: 1.0.0.2
[INFO] - [SWP Switch 2 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 3 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Router 1 NET 0] swp_protocol::processFrame
[INFO] - [SWP Router 1 NET 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 3 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 3 Port: 2] Deliver frame with SeqNum: 8
[INFO] - [SWP Switch 3 Port: 2] send ACK LFR: 8
[INFO] - [Switch Switch 3] Broadcast MAC sent throw all interfaces
[INFO] - [Switch Switch 3] send frame for MAC: ff:ff:ff:ff:ff:ff throw interface 0
[INFO] - [SWP Router 1 NET 0] swp_protocol::processFrame
[INFO] - [SWP Switch 3 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 3 Port: 0] Discarded frame: Frame not in RWS windows
```

```

[INFO] - [SWP Switch 2 Port: 2] swp-protocol::processFrame
[INFO] - [SWP Switch 2 Port: 2] Deliver frame with SeqNum: 8
[INFO] - [SWP Switch 2 Port: 2] send ACK LFR: 8
[INFO] - [Switch Switch 2] Broadcast MAC sent throw all interfaces
[INFO] - [Switch Switch 2] send frame for MAC: ff:ff:ff:ff:ff:ff throw interface 0
[INFO] - [Switch Switch 2] send frame for MAC: ff:ff:ff:ff:ff:ff throw interface 1
[INFO] - [SWP Host 2 NET 0] swp-protocol::processFrame
[INFO] - [SWP Host 2 NET 0] Deliver frame with SeqNum: 10
[INFO] - [SWP Host 2 NET 0] send ACK LFR: 10
[INFO] - [SWP Switch 1 Port: 2] swp-protocol::processFrame
[INFO] - [SWP Switch 1 Port: 2] Deliver frame with SeqNum: 7
[INFO] - [SWP Switch 1 Port: 2] send ACK LFR: 7
[INFO] - [Link Host 2 NET 0] Process ARP packet.
[INFO] - [Switch Switch 1] Broadcast MAC sent throw all interfaces
[INFO] - [Switch Switch 1] send frame for MAC: ff:ff:ff:ff:ff:ff throw interface 0
[INFO] - [SWP Host 1 NET 0] swp-protocol::processFrame
[INFO] - [SWP Host 1 NET 0] Deliver frame with SeqNum: 7
[INFO] - [SWP Host 1 NET 0] send ACK LFR: 7
[INFO] - [Link Host 1 NET 0] Process ARP packet.
[INFO] - [SWP Switch 3 Port: 0] swp-protocol::processFrame
[INFO] - [SWP Host 2 NET 0] swp-protocol::processFrame
[INFO] - [SWP Host 2 NET 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 2 Port: 0] swp-protocol::processFrame
[INFO] - [SWP Switch 2 Port: 1] swp-protocol::processFrame
[INFO] - [SWP Switch 2 Port: 0] swp-protocol::processFrame
[INFO] - [SWP Switch 2 Port: 0] Deliver frame with SeqNum: 10
[INFO] - [SWP Switch 2 Port: 0] send ACK LFR: 10
[INFO] - [Switch Switch 2] send frame for MAC: 0:0:0:0:0:3 throw interface 2
[INFO] - [SWP Host 2 NET 0] swp-protocol::processFrame
[INFO] - [SWP Switch 1 Port: 0] swp-protocol::processFrame
[INFO] - [SWP Host 1 NET 0] swp-protocol::processFrame
[INFO] - [SWP Host 1 NET 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 1 Port: 2] swp-protocol::processFrame
[INFO] - [SWP Switch 1 Port: 2] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 2 Port: 2] swp-protocol::processFrame
[INFO] - [SWP Switch 2 Port: 2] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 3 Port: 0] swp-protocol::processFrame
[INFO] - [SWP Switch 3 Port: 0] Deliver frame with SeqNum: 10
[INFO] - [SWP Switch 3 Port: 0] send ACK LFR: 10
[INFO] - [SWP Switch 2 Port: 0] swp-protocol::processFrame
[INFO] - [SWP Switch 2 Port: 0] Discarded frame: Frame not in RWS windows
[INFO] - [Switch Switch 3] send frame for MAC: 0:0:0:0:0:3 throw interface 2
[INFO] - [SWP Router 1 NET 0] swp-protocol::processFrame
[INFO] - [SWP Router 1 NET 0] Deliver frame with SeqNum: 10
[INFO] - [SWP Router 1 NET 0] send ACK LFR: 10
[INFO] - [Link Router 1 NET 0] Process ARP packet.
[INFO] - [Link Router 1 NET 0] Send control ARP_READY
[INFO] - [IPv4 Router 1] link::Ctrl::ARP_READY
[INFO] - [Link Router 1 NET 0] SEND_PACKET
[INFO] - [Link Router 1 NET 0] ARP ip 1.0.0.2 is in cache.
[INFO] - [SWP Switch 2 Port: 2] swp-protocol::processFrame
[INFO] - [SWP Switch 3 Port: 2] swp-protocol::processFrame
[INFO] - [SWP Router 1 NET 0] swp-protocol::processFrame
[INFO] - [SWP Router 1 NET 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 3 Port: 2] swp-protocol::processFrame
[INFO] - [SWP Switch 3 Port: 2] Deliver frame with SeqNum: 9
[INFO] - [SWP Switch 3 Port: 2] send ACK LFR: 9
[INFO] - [Switch Switch 3] send frame for MAC: 0:0:0:0:0:2 throw interface 0
[INFO] - [SWP Router 1 NET 0] swp-protocol::processFrame
[INFO] - [SWP Switch 3 Port: 0] swp-protocol::processFrame
[INFO] - [SWP Switch 3 Port: 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 2 Port: 2] swp-protocol::processFrame
[INFO] - [SWP Switch 2 Port: 2] Deliver frame with SeqNum: 9
[INFO] - [SWP Switch 2 Port: 2] send ACK LFR: 9
[INFO] - [Switch Switch 2] send frame for MAC: 0:0:0:0:0:2 throw interface 0
[INFO] - [SWP Host 2 NET 0] swp-protocol::processFrame
[INFO] - [SWP Host 2 NET 0] Deliver frame with SeqNum: 11

```

```

[INFO] - [SWP Host 2 NET 0] send ACK LFR: 11
[INFO] - [IPv4 Host 2] Processing arrived ip packet: 1.0.0.2
[INFO] - [IPv4 Host 2] Valid checksum: true
[INFO] - [UDP Host 2] process Segment:
[INFO] - [UDP Host 2] local_port: 53
[INFO] - [UDP Host 2] remote_port: 53
[INFO] - [UDP Host 2] local_ip: 1.0.0.2
[INFO] - [UDP Host 2] remote_ip: 4.0.0.1
[INFO] - [UDP Host 2] valid checksum: true
[INFO] - [UDP Host 2] socket local_ip: 1.0.0.2
[INFO] - [UDP Host 2] socket local_ip: 53
[INFO] - [UDP Host 2] Deliver dns packet
[INFO] - [DNS Host 2] Incoming packet:
[INFO] - [DNS Host 2]
----- dns packet -----
** HEADER **
id: 0
flags_code: 1280
QDCount: 1
ANCount: 1
NSCount: 0
ARCount: 1

** QUESTION SECTION **
network.devs.com A 0.0.0.0 IN 0

** ANSWER SECTION **
network.devs.com A 6.0.0.1 IN 65535

** AUTHORITATIVES SECTION **

** ADDITIONAL SECTION **
network.devs.com A 4.0.0.1 IN 9227
-----  

[INFO] - [DNS Host 2] Answer packet
[INFO] - [DNS Host 2] Answer Type: A
[INFO] - [DNS Host 2] Caching authoritative answer
[INFO] - [DNS Host 2] Send DNS Query to: (1.0.0.1, 53)
[INFO] - [UDP Host 2] Write_to/Send_to
[INFO] - [UDP Host 2] Read/Recv
[INFO] - [IPv4 Host 2] Forwarding ip packet: 1.0.0.1
[INFO] - [IPv4 Host 2] Best route for packet with dest_ip: 1.0.0.1
is: | 1.0.0.1 | 255.255.255.255 | 1.0.0.1 | 1 | Host 1 |
[INFO] - [IPv4 Host 2] Added packet to wait ARP for nexthop: 1.0.0.1
[INFO] - [IPv4 Host 2] ARP query throw interface: NET 0
[INFO] - [Link Host 2 NET 0] ARP_QUERY
[INFO] - [Link Host 2 NET 0] ARP ip 1.0.0.1 not cached.
[INFO] - [Link Host 2 NET 0] Sending ARP query for ip: 1.0.0.1
[INFO] - [SWP Switch 3 Port: 0] swp-protocol::processFrame
[INFO] - [SWP Switch 2 Port: 0] swp-protocol::processFrame
[INFO] - [SWP Host 2 NET 0] swp-protocol::processFrame
[INFO] - [SWP Host 2 NET 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 2 Port: 0] swp-protocol::processFrame
[INFO] - [SWP Switch 2 Port: 0] Deliver frame with SeqNum: 11
[INFO] - [SWP Switch 2 Port: 0] send ACK LFR: 11
[INFO] - [Switch Switch 2] Broadcast MAC sent throw all interfaces
[INFO] - [Switch Switch 2] send frame for MAC: ff:ff:ff:ff:ff:ff throw interface 1
[INFO] - [Switch Switch 2] send frame for MAC: ff:ff:ff:ff:ff:ff throw interface 2
[INFO] - [SWP Host 2 NET 0] swp-protocol::processFrame
[INFO] - [SWP Switch 1 Port: 2] swp-protocol::processFrame
[INFO] - [SWP Switch 1 Port: 2] Deliver frame with SeqNum: 8
[INFO] - [SWP Switch 1 Port: 2] send ACK LFR: 8
[INFO] - [Switch Switch 1] Broadcast MAC sent throw all interfaces
[INFO] - [Switch Switch 1] send frame for MAC: ff:ff:ff:ff:ff:ff throw interface 0
[INFO] - [SWP Switch 2 Port: 2] swp-protocol::processFrame
[INFO] - [SWP Switch 2 Port: 2] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 3 Port: 0] swp-protocol::processFrame

```

```

[INFO] - [SWP Switch 3 Port: 0] Deliver frame with SeqNum: 11
[INFO] - [SWP Switch 3 Port: 0] send ACK LFR: 11
[INFO] - [SWP Switch 2 Port: 1] swp_protocol::processFrame
[INFO] - [Switch Switch 3] Broadcast MAC sent throw all interfaces
[INFO] - [Switch Switch 3] send frame for MAC: ff:ff:ff:ff:ff:ff throw interface 2
[INFO] - [SWP Host 1 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 1 NET 0] Deliver frame with SeqNum: 8
[INFO] - [SWP Host 1 NET 0] send ACK LFR: 8
[INFO] - [Link Host 1 NET 0] Process ARP packet.
[INFO] - [SWP Router 1 NET 0] swp_protocol::processFrame
[INFO] - [SWP Router 1 NET 0] Deliver frame with SeqNum: 11
[INFO] - [SWP Router 1 NET 0] send ACK LFR: 11
[INFO] - [Link Router 1 NET 0] Process ARP packet.
[INFO] - [SWP Switch 2 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 1 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Host 1 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 1 NET 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 1 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 1 Port: 0] Deliver frame with SeqNum: 3
[INFO] - [SWP Switch 1 Port: 0] send ACK LFR: 3
[INFO] - [SWP Switch 3 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Router 1 NET 0] swp_protocol::processFrame
[INFO] - [SWP Router 1 NET 0] Discarded frame: Frame not in RWS windows
[INFO] - [Switch Switch 1] send frame for MAC: 0:0:0:0:0:2 throw interface 2
[INFO] - [SWP Switch 3 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 3 Port: 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Host 1 NET 0] swp_protocol::processFrame
[INFO] - [SWP Switch 1 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 1 Port: 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 2 Port: 1] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 1] Deliver frame with SeqNum: 3
[INFO] - [SWP Switch 2 Port: 1] send ACK LFR: 3
[INFO] - [Switch Switch 2] send frame for MAC: 0:0:0:0:0:2 throw interface 0
[INFO] - [SWP Host 2 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 2 NET 0] Deliver frame with SeqNum: 12
[INFO] - [SWP Host 2 NET 0] send ACK LFR: 12
[INFO] - [Link Host 2 NET 0] Process ARP packet.
[INFO] - [Link Host 2 NET 0] Send control ARP_READY
[INFO] - [IPv4 Host 2] link::Ctrl::ARP_READY
[INFO] - [Link Host 2 NET 0] SEND_PACKET
[INFO] - [Link Host 2 NET 0] ARP ip 1.0.0.1 is in cache.
[INFO] - [SWP Switch 1 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Host 2 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 2 NET 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 2 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 0] Deliver frame with SeqNum: 12
[INFO] - [SWP Switch 2 Port: 0] send ACK LFR: 12
[INFO] - [Switch Switch 2] send frame for MAC: 0:0:0:0:0:1 throw interface 1
[INFO] - [SWP Host 2 NET 0] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 1] swp_protocol::processFrame
[INFO] - [SWP Switch 2 Port: 1] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 1 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 1 Port: 2] Deliver frame with SeqNum: 9
[INFO] - [SWP Switch 1 Port: 2] send ACK LFR: 9
[INFO] - [Switch Switch 1] send frame for MAC: 0:0:0:0:0:1 throw interface 0
[INFO] - [SWP Host 1 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 1 NET 0] Deliver frame with SeqNum: 9
[INFO] - [SWP Host 1 NET 0] send ACK LFR: 9
[INFO] - [IPv4 Host 1] Processing arrived ip packet: 1.0.0.1
[INFO] - [IPv4 Host 1] Valid checksum: true
[INFO] - [UDP Host 1] process Segment:
[INFO] - [UDP Host 1] local_port: 53
[INFO] - [UDP Host 1] remote_port: 53
[INFO] - [UDP Host 1] local_ip: 1.0.0.1
[INFO] - [UDP Host 1] remote_ip: 1.0.0.2
[INFO] - [UDP Host 1] valid checksum: true

```

```

[INFO] - [UDP Host 1] socket local_ip: 1.0.0.1
[INFO] - [UDP Host 1] socket local_ip: 53
[INFO] - [UDP Host 1] Deliver dns packet
[INFO] - [DNS Host 1] Incoming packet:
[INFO] - [DNS Host 1]
----- dns packet -----
** HEADER **
id: 0
flags_code: 1280
QDCount: 1
ANCount: 1
NSCount: 0
ARCount: 1

** QUESTION SECTION **
network.devs.com A 0.0.0.0 IN 0

** ANSWER SECTION **
network.devs.com A 6.0.0.1 IN 65535

** AUTHORITATIVES SECTION **

** ADDITIONAL SECTION **
network.devs.com A 1.0.0.2 IN 9227
-----  

[INFO] - [DNS Host 1] Answer packet
[INFO] - [DNS Host 1] Answer Type: A
[INFO] - [DNS Host 1] Caching authoritative answer
[INFO] - [UDP Host 1] Read/Recv
[INFO] - [SWP Switch 2 Port: 1] swp_protocol::processFrame
[INFO] - [SWP Switch 1 Port: 0] swp_protocol::processFrame
[INFO] - [SWP Host 1 NET 0] swp_protocol::processFrame
[INFO] - [SWP Host 1 NET 0] Discarded frame: Frame not in RWS windows
[INFO] - [SWP Switch 1 Port: 2] swp_protocol::processFrame
[INFO] - [SWP Switch 1 Port: 2] Discarded frame: Frame not in RWS windows
Simulation Ended (0.0217 sec)

```

10. References

Referencias

[1] Federico Bergero and Ernesto Kofman. Powerdevs: a tool for hybrid system modeling and real-time simulation. *Simulation*, 87(1-2):113–132, 2011.

[2] The TCP/IP Guide. Dns message header and question section format. http://www.tcpipguide.com/free/t_DNSMessageHeaderandQuestionSectionFormat.htm.

[3] Larry L Peterson and Bruce S Davie. *Computer networks: a systems approach*. Elsevier, fifth edition, 2011.

[4] STD. std::va_arg. http://www.cplusplus.com/reference/cstdarg/va_arg/.

[5] STD. std::va_start. http://www.cplusplus.com/reference/cstdarg/va_start/.

- IP
- DNS
- UDP
- routing tables

↗?
No son parte de la librería estandar de c++. No creo que valga la pena referenciarlas. A lo sumo referencia a c++

NO!

Poner la referencia al RFC!