

Enunciado

Una red SDN (Software Defined Network) es una red de computadores moderna en la que los *planos de control y datos* están completamente desacoplados. Una red SDN está compuesta por tres tipos de nodos: los computadores (*hosts*) donde comienza y termina la comunicación de datos (*paquetes*), los nodos conmutadores programables (*switches*), a través de los cuales fluyen los datos, y los *controladores*, que son nodos especiales capaces de programar dinámicamente el comportamiento de los *switches*. Los nodos se conectan entre sí a través de *puertos*. Aunque, en general, en una red SDN puede haber muchos nodos *controladores*, para simplificar esta práctica suponemos que el **controlador** es **único**. La Figura 1 muestra un ejemplo de la estructura de una red SDN con un controlador, tres switches y varios hosts conectados a cada uno de los switches.

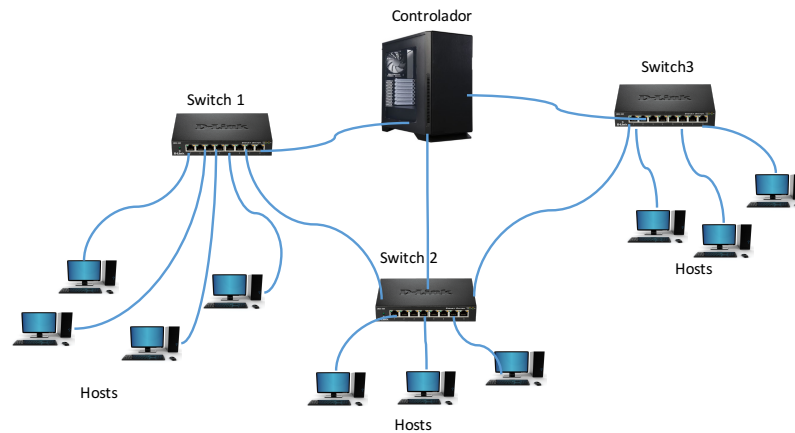


Figura 1: Ejemplo de red SDN

Los *switches* contienen *tablas* con *reglas* que utilizan para saber cómo encaminar los datos que le llegan a través de sus puertos. De forma muy simplificada, una *regla* de un *switch* es un registro que contiene la siguiente información:

1. *Tipo de Paquete* al que se aplica la regla
2. *Puerto de Entrada* por el que llega el paquete al que se le aplica la regla
3. *Conjunto de Puertos* por los que el paquete debe ser reenviado

De este modo cuando, un paquete llega a un switch por un puerto de entrada, el switch busca en su tabla de reglas alguna que pueda aplicarse (coincide el puerto de entrada y el tipo de paquete). Si hay alguna reenvía el paquete por los puertos que indica la regla. Si no hay ninguna regla, reenvía el paquete al nodo controlador y espera a que éste le diga qué tiene que hacer con el paquete. El nodo controlador también puede enviar paquetes a los switches para actualizar las tablas de reglas.

El objetivo de esta práctica es la descripción en Alloy de redes SDN. Para simplificar la práctica, la tarea de modelado se realizará progresivamente en distintas fases. En este primer enunciado se modela la parte estática de la red SDN.

Primera Fase

En esta primera fase, se construyen topologías de redes SDN.

1. Define una signatura **Puerto** y una relación **conectado**, que relaciona dos puertos.
2. Define una signatura **abstracta** **Nodo** y dos relaciones **puertos** y **conectadoCon** que, respectivamente, relacionan cada nodo con sus puertos, y con el conjunto de nodos con los que está directamente conectado.
3. Define las signaturas **Host**, **Switch** y **Controlador** como extensiones de **Nodo**.
4. Implementa las siguientes funciones:
 - a) **Puertos(n:Nodo):set Puerto** que, dado un nodo, devuelve el conjunto de sus puertos.
 - b) **Nodo(p:Puerto):Nodo** que, dado un puerto devuelve el nodo al que pertenece
 - c) **NodosConectados(n:Nodo):set Nodo** que, dado un nodo, devuelve el conjunto de nodos con los que está directamente conectado a través de sus puertos.
 - d) **NodosAlcanzables(n:Nodo):set Nodos** que, dado un nodo, devuelve el conjunto de nodos con los que está conectado (directa o indirectamente) a través de sus puertos, excepto el propio nodo **n**.
5. Añade hechos que aseguren que las topologías generadas por Alloy satisfacen las siguientes restricciones:
 - a) Cada puerto pertenece exactamente a un nodo.
 - b) La relación **conectado** es simétrica, es decir, un puerto **p1** está **conectado** con un puerto **p2** si, y sólo si, **p2** está **conectado** con **p1**.
 - c) Ningún puerto está conectado con otro puerto de su mismo nodo.
 - d) La relación **conectadoCon** relaciona cada nodo con los que son *directamente* accesibles a través de sus puertos.
 - e) Todos los switches están conectados con el controlador.
 - f) Cada host está conectado con exactamente un switch (y no está conectado ni con otro host, ni con el controlador).
 - g) Todos los hosts no están conectados con el mismo switch.
6. Genera instancias con más de dos hosts, y que contengan más hosts que switches.

Nota: Genera instancias acotando el número de átomos por signatura con valores altos (mayores que 15, run `pred for 15`).

Segunda Fase

En la segunda fase, se añaden datos a la red. Extiende el modelo de la primera fase añadiendo las siguientes signaturas, relaciones y hechos.

1. Define una signatura **abstracta** **TipoPaquete** y signaturas **FTP**, **HTTP**, **SMTP**, **Control** como sus únicas extensiones, de las que, además, en cada instancia, hay exactamente un átomo.
2. Define la signatura **Paquete** y las relaciones **tipo**, **posicion**, **origen**, **destino** que, respectivamente, asocian cada paquete con su tipo (de **TipoPaquete**), el puerto en el que se encuentran (si es que está en algún puerto), y los nodos origen y destino del paquete (que son nodos hosts).
3. Añade hechos que aseguren que:
 - a) los hosts origen y destino de cada paquete deben ser distintos.
 - b) si un paquete es de tipo **Control** entonces no puede estar en un puerto de un nodo que no sea un switch.
 - c) para cada paquete existe un *camino* en la red desde su posición actual hasta el nodo destino. Un camino es un recorrido a través de los puertos conectados en la red.

Tercera Fase

En la tercera fase, se añaden reglas y tablas a los switches de la red. Para ello, extiende el modelo de la segunda fase con las siguientes firmas, relaciones y hechos.

1. Define una firma **Regla** y relaciones **tipoPaquete**, **puertoEntrada**, **reenviar** que asocian cada regla con: (1) el tipo de paquete al que se aplica (de **TipoPaquete**), (2) el puerto de entrada por el que deben llegar los paquetes sobre los que se aplica la regla, y (3) el conjunto de puertos por los que se debe reenviar el paquete al que se aplica la regla.
2. Añade una relación **nuevaRegla**, que asocia cada paquete con una nueva regla (si es que la lleva).
3. Añade una relación **tabla** que asocia cada nodo con el conjunto de reglas que tiene activas.
4. Añade una relación **ternaria puertosAlcanzablesDesde** que asocia cada nodo switch **n**, y cada puerto de ese nodo **p**, con el conjunto de puertos remotos alcanzables, utilizando alguna de las reglas del switch, para paquetes que llegan al nodo **n** a través del puerto **p**.
5. Añade hechos que aseguren que todas las instancias satisfacen que:
 - a) ni los hosts ni el controlador tienen tablas de reglas.
 - b) cada regla pertenece a una y sólo una tabla.
 - c) el puerto de entrada y los puertos de salida de cada regla son del switch en los que se encuentra la regla.
 - d) un paquete lleva una nueva regla si, y sólo si, es de tipo **Control**.
 - e) el tipo paquete de una regla de un switch no puede ser **Control**.
 - f) el puerto de entrada de una regla no puede ser uno de los puertos por los que hay que reenviar el paquete.
 - g) [**Difícil:**] no hay ciclos en la red, es decir, las reglas de los switches deben asegurar que ningún paquete que ha visitado un switch vuelve al switch antes de llegar a su destino.

FIN DE LA PRIMERA PARTE DE LA PRÁCTICA

Se deben respetar los nombres de las firmas y relaciones del enunciado. Si algún estudiante lo ve estrictamente necesario puede añadir más firmas/relaciones, pero explicando claramente su necesidad y uso. Se pueden añadir las funciones y predicados que se vea necesario con el objetivo de simplificar los asertos.

Recuerda que esta es una práctica que debéis hacer en parejas. Los dos componentes deben participar **IGUALMENTE** en el desarrollo de la práctica, y además la documentación entregada **DEBE** ser el fruto del trabajo **EXCLUSIVO** de cada pareja. Cualquier anomalía que se detecte en este sentido se tendrá en cuenta en la evaluación de la práctica.