Situación actual de OBSModules

Javier Armendáriz Silva

Índice

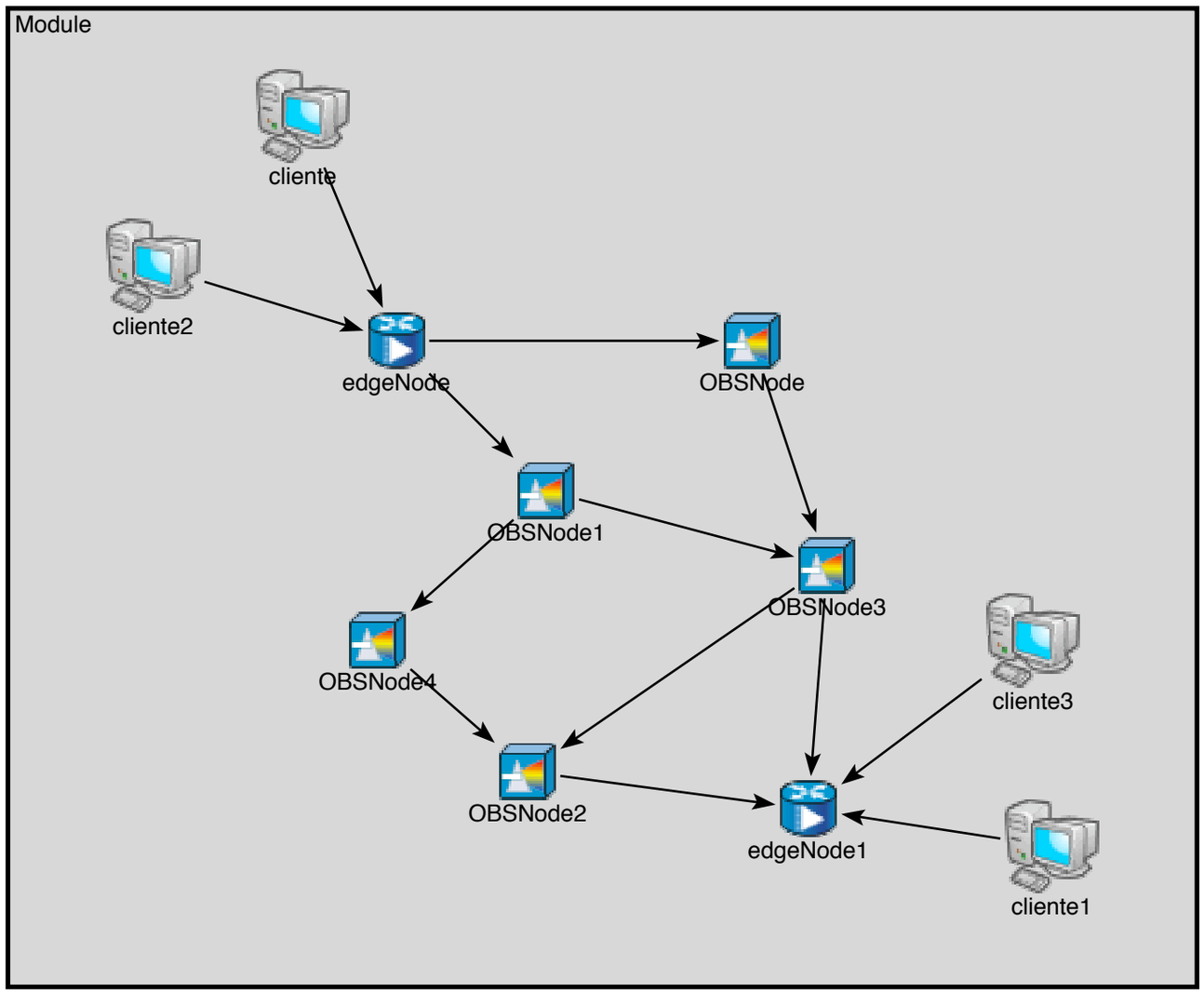
- Repaso de la tecnología OBS
- OBSModules
  - RouterOBS
  - SwitchOBS
- perdidasOBS
- Conclusiones

Optical Burst Switching (OBS)

- Tecnología de conmutación óptica
  - Combinación de OPS y WRN
    - Flexibilidad de una red de conmutación de paquetes
  - Establece caminos virtuales como en WRN
    - Mayor simplicidad en los nodos

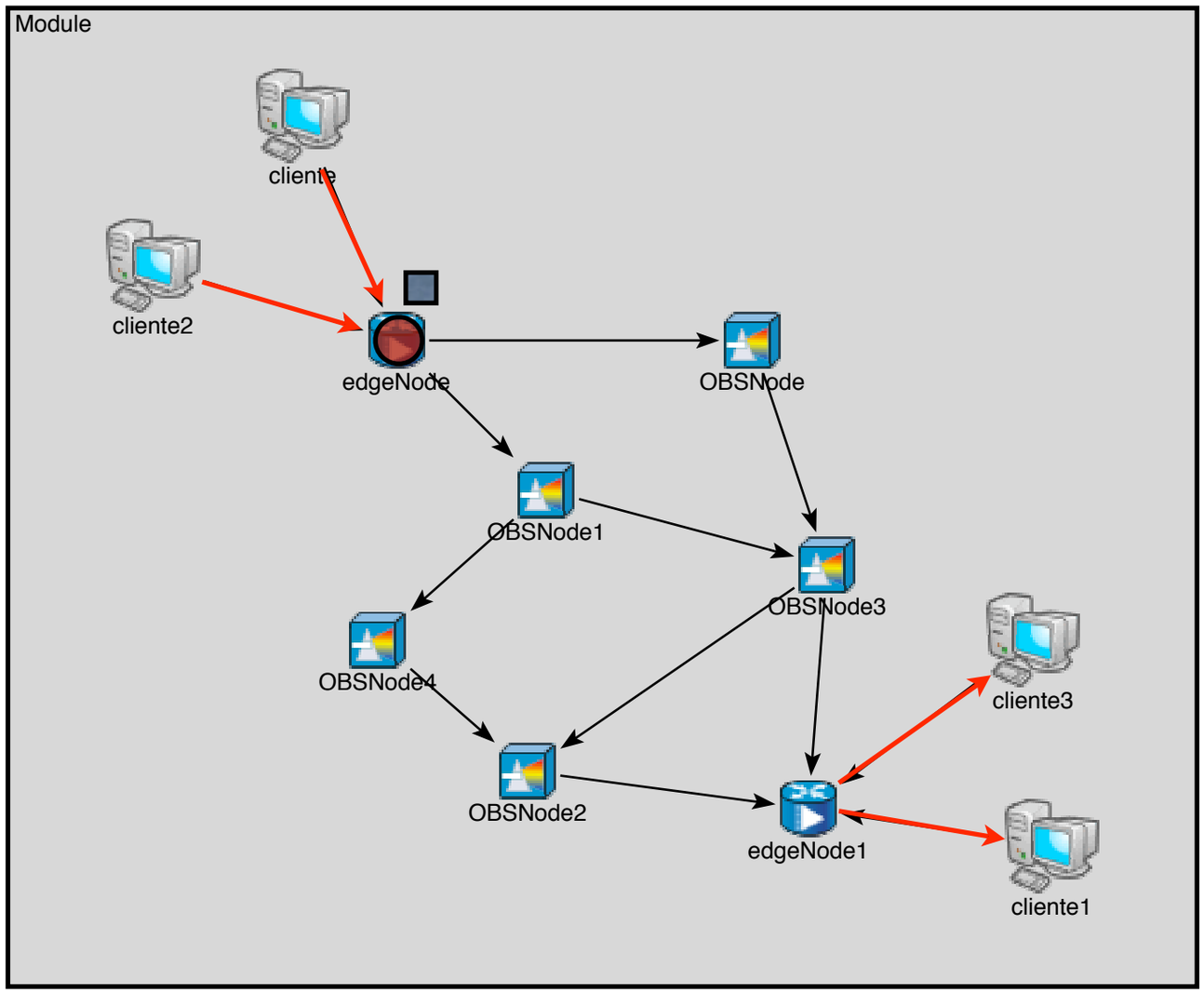
Arquitectura de una red OBS

- Nodos frontera
- Switches intermedios



Transparente para los clientes. Ellos no tienen porque saber que sus paquetes viajan a través de una red óptica.

OBS: Funcionamiento

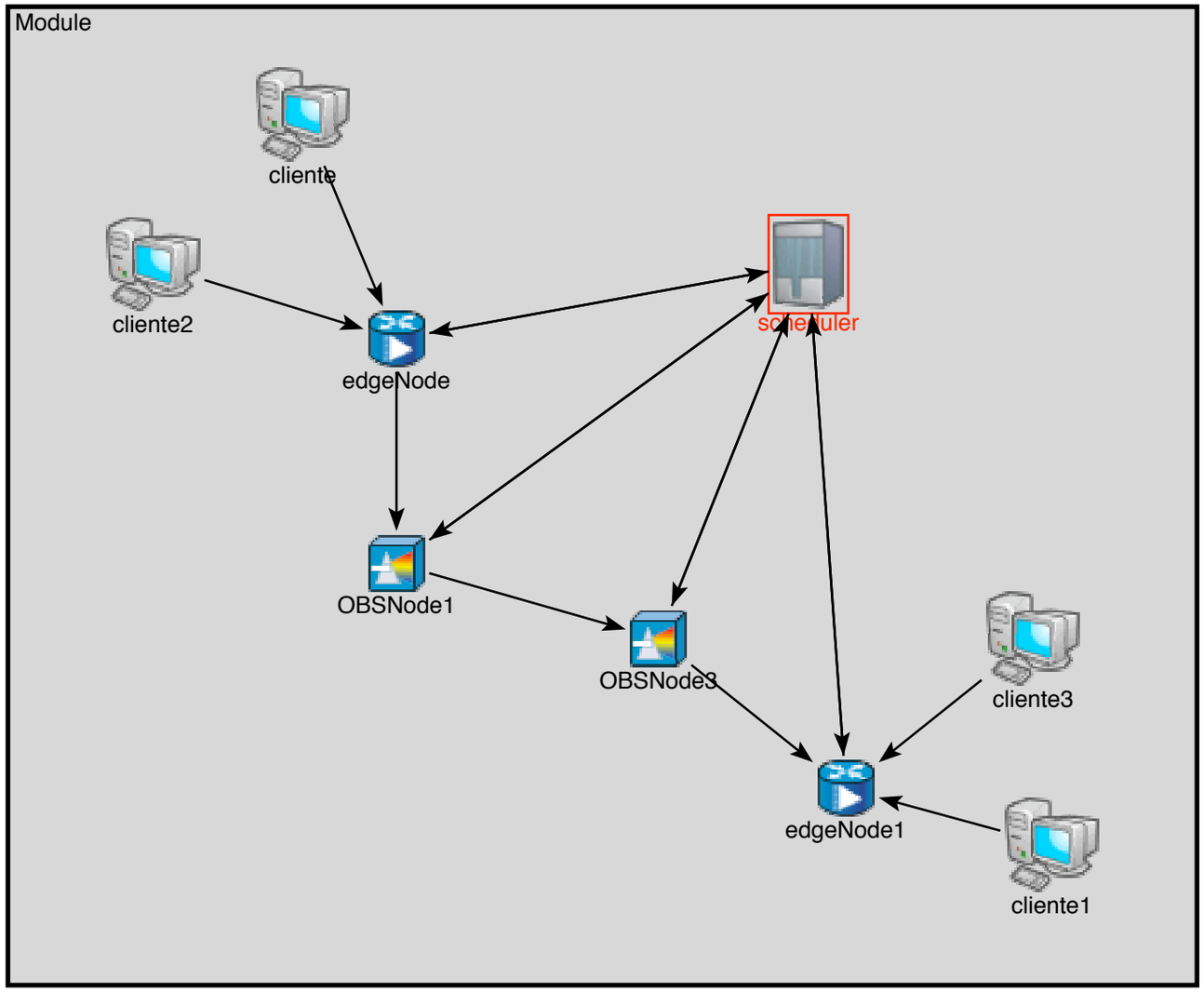


El círculo rojo representa una ráfaga.
El cuadrado azul es un mensaje de control.

Paradigmas en OBS

- Centralizada (Dueser y Bayvel)

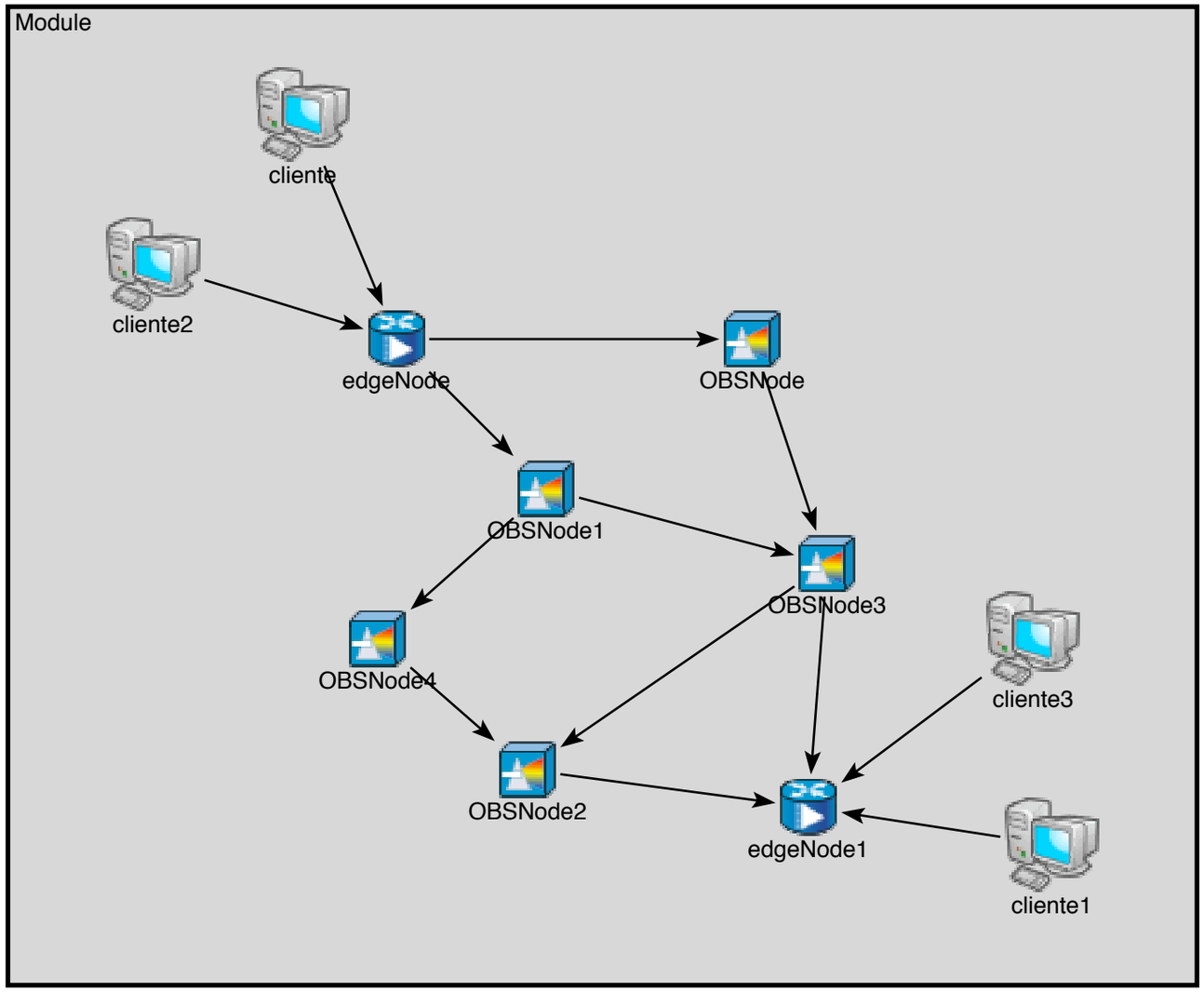
Sería mejor si dieras una descripcion de lo que hace OBS no?



El scheduler se encarga de gestionar las reservas de canal. Esto hace al sistema poco escalable pero simplifica los nodos intermedios.

Paradigmas en OBS

- Distribuido



OBS: Variaciones

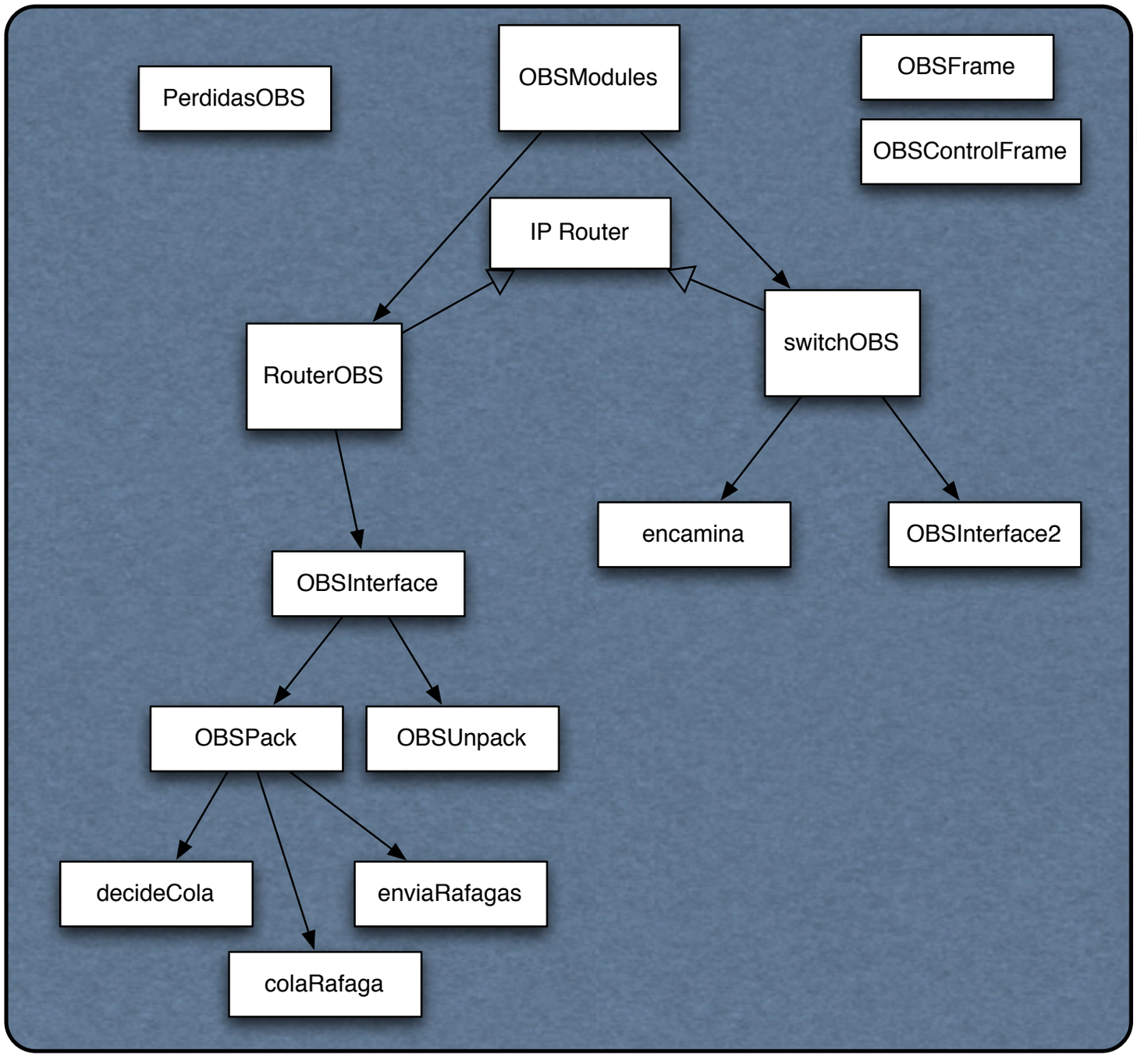
- En todas las arquitecturas mencionadas anteriormente existen variaciones. Entre otras:
  - Algoritmo de formación de ráfagas (*timer*, tamaño de ráfaga, mixto)
  - Algoritmos de reserva de canal
  - Algoritmos de *Scheduling*

OBSModules

- Implementación básica de una arquitectura OBS distribuida.
  - Formación de ráfagas por *timer* y por número máximo de paquetes
  - Indica el tiempo de llegada de la ráfaga en el mensaje de control
  - No hay *contentions*.
    - Para simular las pérdidas de ráfagas en la red OBS se ha implementado un módulo.

OBSModules

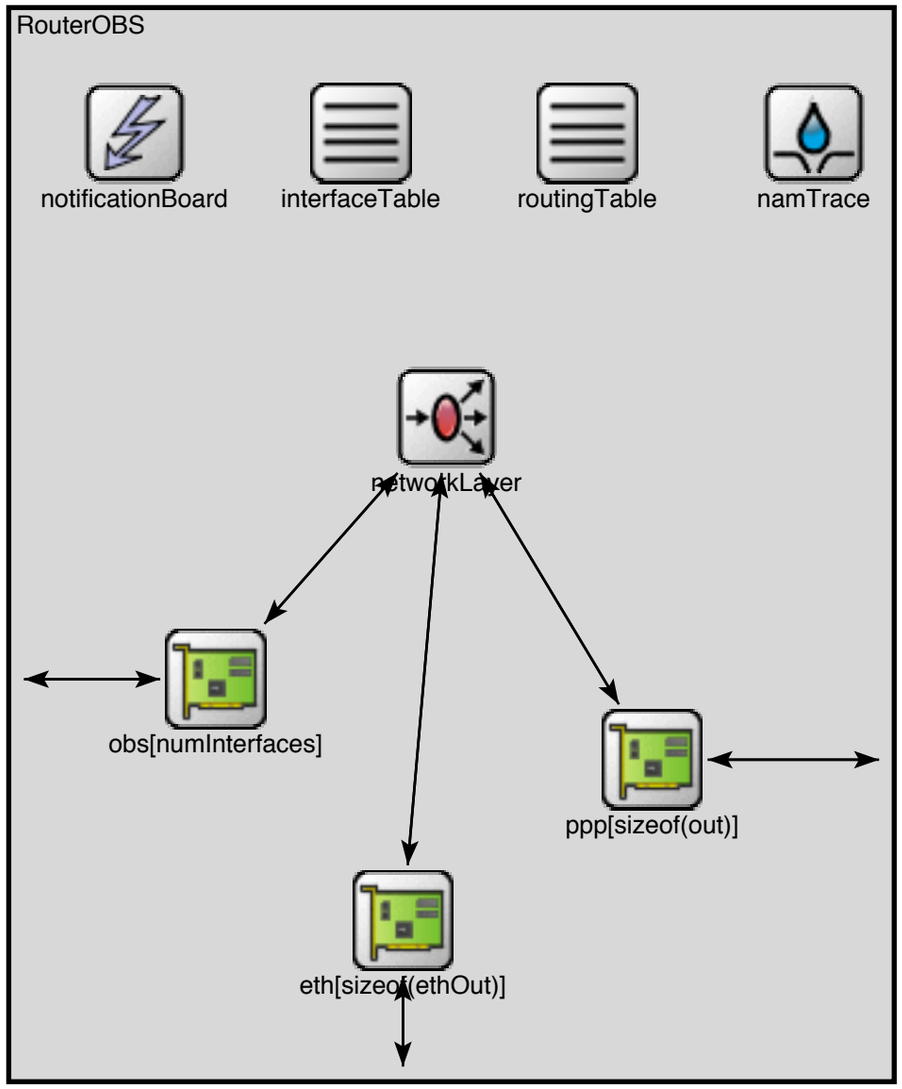
Módulo extra para simular pérdidas de ráfagas ---------->



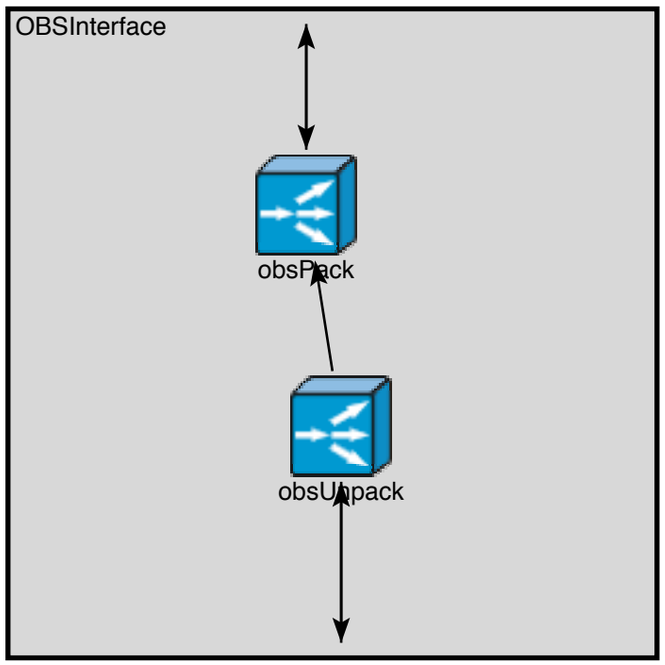
<- Mensajes (.msg)

Esta sería la jerarquía de módulos del proyecto. La relación entre el Router y el Switch OBS no es de herencia exactamente. Ambos presentan la misma estructura del Router IP pero incluyen interfaces OBS.

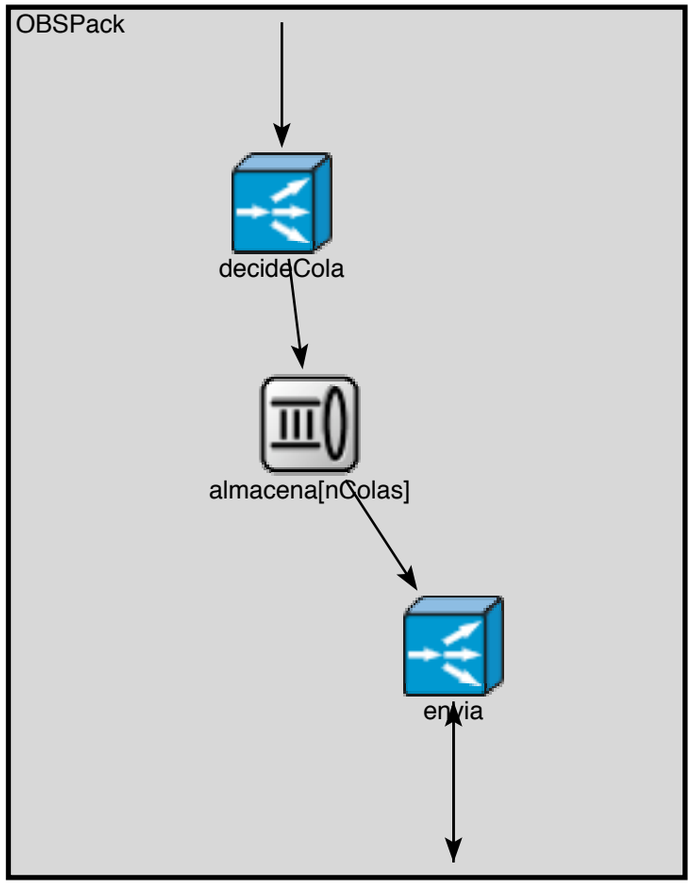
Módulo RouterOBS



OBSInterface



OBSPack



OBSPack

- Módulo compuesto.
- Almacena los paquetes recibidos por la entrada *netIn* en una cola elegida *aleatoriamente*.
- Pasado un *timer* o un tamaño mínimo de ráfaga, se ensambla y se reserva un canal para enviar la ráfaga.

DecideCola

- Recibe paquetes y los almacena en una cola para formar ráfagas.
- La cola se elige aleatoriamente.
- El array se crea en el momento de la inicialización del sistema. No se crean más ni se destruyen dinámicamente.

ColaRafaga

- Módulo simple usado en OBSPack para ir guardando paquetes de la ráfaga.
- Puede definirse el tipo de cola y la capacidad máxima.

EnviaRafagas

- Decide a qué canal irán dirigida las ráfagas que entren aquí y las redirige
  - Si todos los canales están ocupados los almacena en una cola propia
  - Si están todos libres se envía a un canal aleatorio
  - Sino se envía en el primer canal libre

cada interfaz tiene lambda canales. de los cuales lambda-1 es el canal de control

Sabemos en cada momento qué canal esta ocupado en el array ocupacionCanales.

El método startTransmitting envía un msg de comienzo de ráfaga al canal indicado y programas el fin de transmisión para el instante sim_time + t_tx

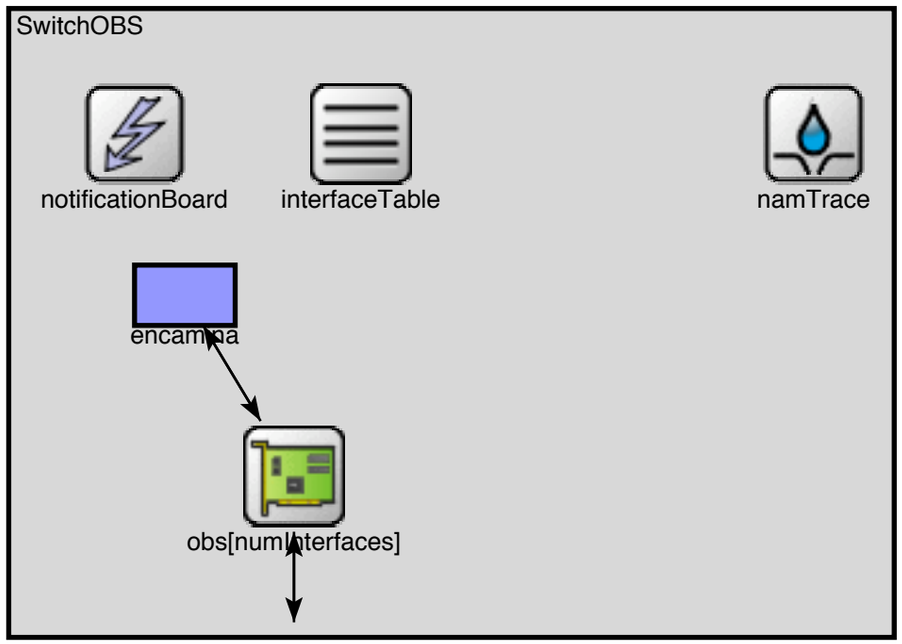
EnviaRafaga: Protocolo

- Si el canal de control está ocupado, guarda en cola
- Si no está ocupado, reserva el canal de control y envía un mensaje de reserva de canal.
- En el instante simTime() +tiempoTransmision, se libera el canal de control y se envía el mensaje de principio

OBSUnpack

- Módulo encargado de extraer los paquetes almacenados dentro de una ráfaga para enviarlos a su destinatario.
  - Empieza a desempaquetar nada más recibir el mensaje de final de ráfaga
- Envía un aviso de llegada a OBSPack por cada paquete desencapsulado

Módulo SwitchOBS



switchOBS

- Sólo dispone de interfaces OBS
  - Simplemente reenvía las ráfagas de un puerto a otro
    - Distinta a la interfaz del router. Esta es mucho más simple.
  - Enrutado por longitud de onda (*lambda*)
  - Se genera la tabla de rutas en un fichero

Encamina

- Lee la tabla de rutas a partir de un fichero (routingFile)
  - El fichero contiene un array de structs (fichero binario)
- A partir de la información de la tabla redirige un puerto de entrada hacia un puerto de salida.

routingFile

- Tiene que crearse previamente con el programa creaRutas.c (no incluido en el svn)
- Enrutamiento estático
- No existe reserva de recursos como tal ni políticas de *scheduling*

OBSInterface2

- Simplemente reenvía todos los mensajes que reciben
  - Si procede de physIn, lo envía a Encamina para que le asigne una ruta
  - Si procede de Encamina, la redirige al puerto de salida asignado

perdidasOBS

- Módulo implementado para simular las pérdidas de ráfagas que pueden ocurrir en una red OBS
  - Completamente aleatorio.
  - “Recuerda” los paquetes borrados para eliminar también los paquetes de fin de ráfaga correspondientes.

Conclusiones

- OBSModules es usable actualmente, pero presenta ciertas carencias:
  - Compilación compleja debido a OBSControlFrame
  - No implementa reserva dinámica de lambdas. No se pueden probar algoritmos de *scheduling*
  - Poco realista. Las ráfagas no se forman teniendo el cuenta el destinatario ni QoS

Nueva arquitectura

- Sería conveniente saber que funcionalidades se esperan en este módulo y cómo sería el uso deseado. (Captura de requisitos)
  - Interesa implementar la opción de un servidor centralizado de *scheduling?*
  - Soporte QoS? Multicast?