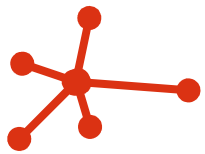
upna Universidad Pública de Navarra
Nafarroako Unibertsitate Publikoa

Módulos OBS

J. Armendariz, F. Espina, M. Izal, D. Morato
felix.espina@unavarra.es



GRSST Grupo de Redes, Sistemas y Servicios Telemáticos
http://www.tlm.unavarra.es

Contenido

- [5] Introducción OCS-OPS-OBS
- [1] Objetivos
- [7] Edge Node
- [8] Core Node
- [1] Módulos extras
- [1] Ejemplo

OCS

- Optical Circuit-Switching (OCS)
  - Se establecen circuitos entre los routers que necesitan comunicarse (estaticos o bajo demanda)
  - Los nodos del nucleo manejan muchas λs pero de forma pasiva (no hay limitacion electrónica)
  - Poca flexibilidad (caidas de enlace, cambios de trafico...)

OPS

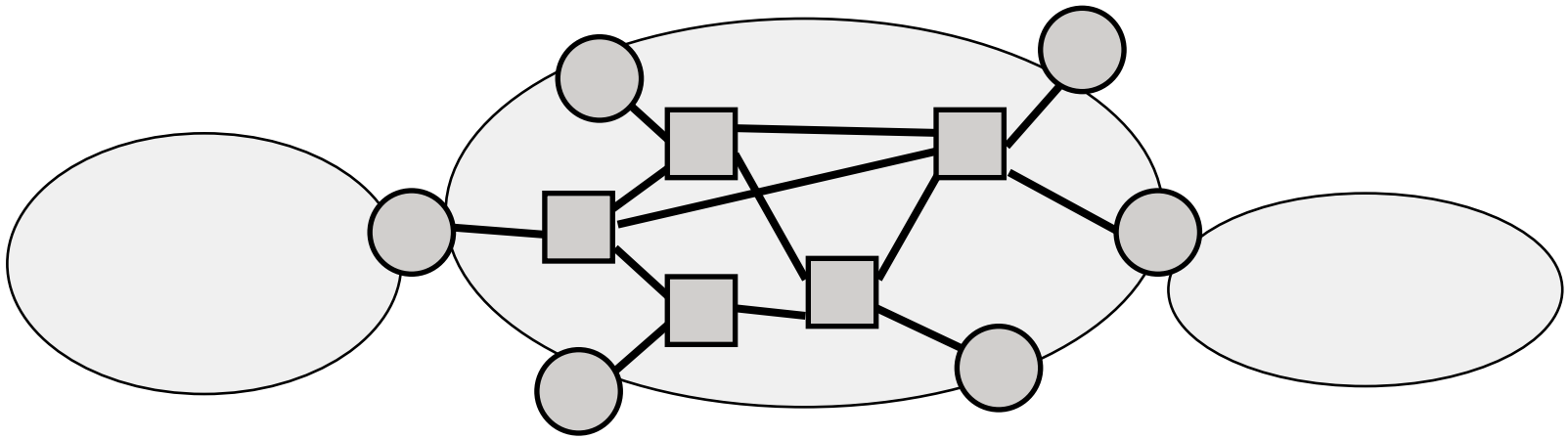
- Optical Packet Switching (OPS)
  - Es el ideal, cuando se pueda hacer OBS dejara de tener sentido?
  - Problemas
  - Muy dificil hacer colas. En optica integrada no hay memoria, solo lineas de retardo
  - Los conmutadores opticos tienen un tiempo de conmutación (ms?, ns?)

OBS

- Optical Burst Switching
- (Conmutación óptica de ráfagas)
  - Intermedio: establecer un circuito sólo para la duración de un conjunto de paquetes (ráfaga o burst)

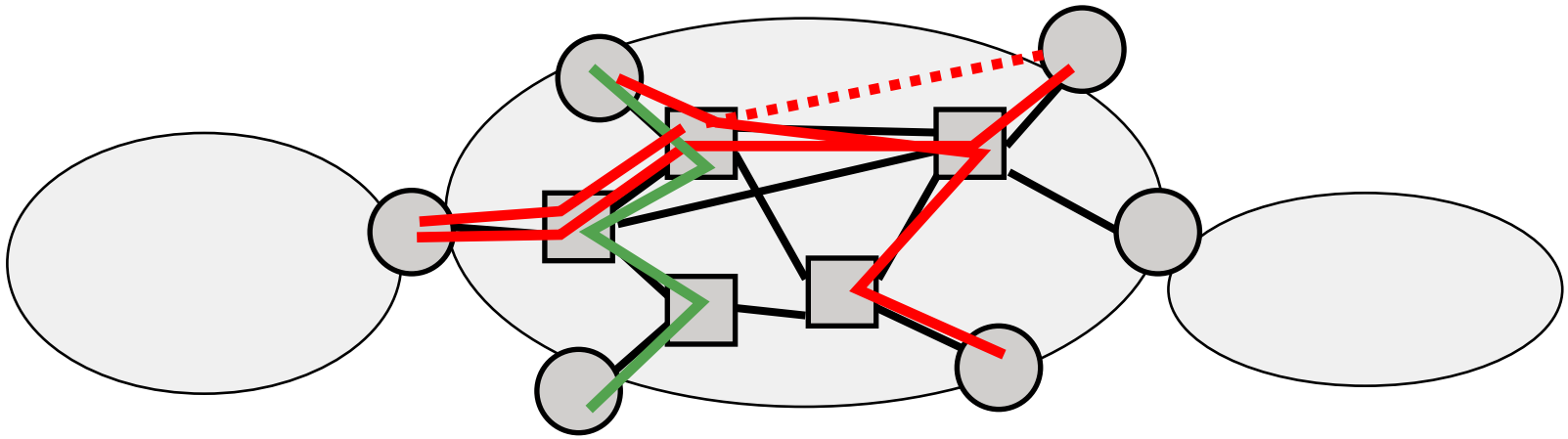
OBS

- Existen 2 modelos:
  - Redes con nodos completos: tráfico entrada/salida + tráfico óptico en tránsito
  - Redes con
    - Edge Nodes: tráfico entrada/salida (ensamblado/desensamblado ráfagas ópticas)
    - Core Nodes: tráfico óptico en tránsito



OBS

- Los Edge nodes van formando ráfagas y enviándolas
- El núcleo simplemente programa los conmutadores para que cuando llegue cada ráfaga tenga un camino
  - El BCP de cada rafaga indica en que tiempo despues llegara la ráfaga (En algunas versiones tambien la duración)
  - Si no es posible planificar una rafaga se pierde sin que el origen se entere (perdidas en bloque)

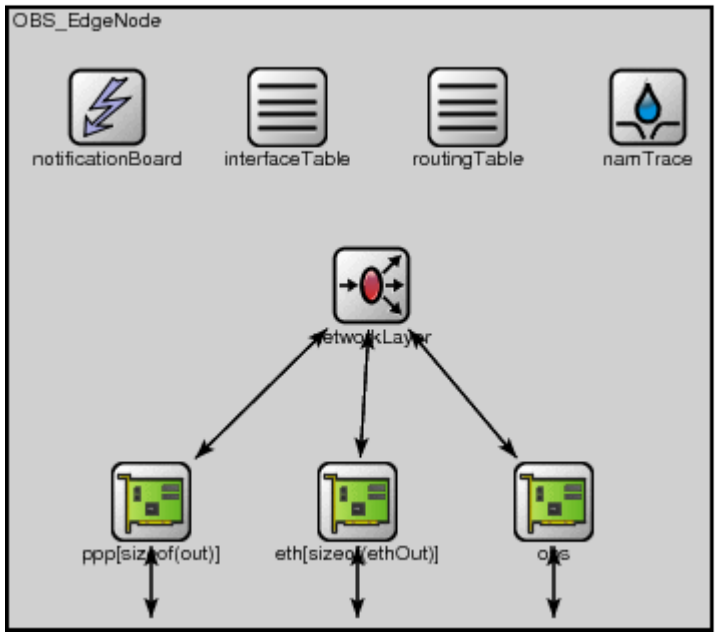


Objetivos

- Hacer investigación sobre OBS
- Testbeds??
  - Existen pocas
  - No tenemos recursos materiales ni financieros
- Simulaciones
  - Existen implementaciones?? Algunas
  - Son accesibles?? Pocas
  - Son buenas?? NO
- Objetivo: desarrollar los módulos de simulación necesarios en OMNeT++ para poder hacer investigación sobre OBS.

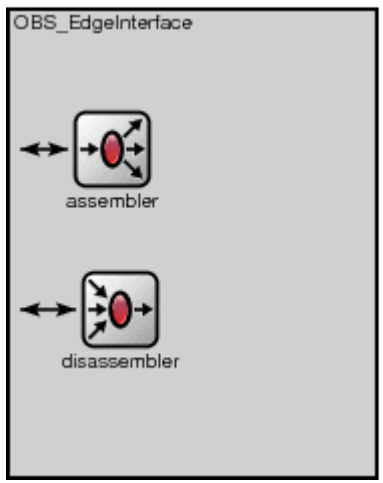
Edge Node

- Edge Nodes: tráfico entrada/salida (ensamblado/desensamblado ráfagas ópticas)
- Router OMNeT++/INET (basado en él) + interfaz OBS
- Params: numLambdas, routingFile



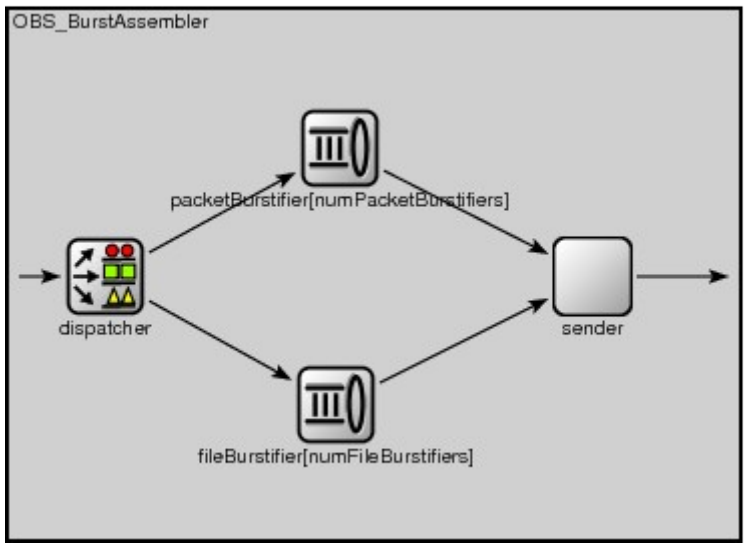
Edge Node - Interface

- La interfaz OBS tiene 2 submódulos:
  - Assembler: convierte el tráfico de entrada en ráfagas ópticas y las inyecta en la red óptica
  - Disassembler: convierte las ráfagas ópticas que salen de la red óptica en tráfico de salida
- Params: numLambdas (, colourIn, colourOut)



Edge Node - Assembler

- Se compone de 3 submódulos simples en cadena:
  - Dispatcher
  - PacketBurstifier // FileBurstifier
  - Sender
- Params: numLambdas, dispatcherRules, numPacketBurstifiers, numFileBurstifiers (, colours)



Edge Node - Assembler

- Dispatcher:
  - Responsable de decidir según reglas del EdgeNode a qué burstifier enviar el tráfico de entrada => diferenciación de tráfico, QoS
  - Reglas: IP origen/destino, Port origen/destino, Protocolo

```
# TCP (protocol 6) packet with http traffic from 10.0.3.3
destPort 80 srcAddr 10.0.3.3 protocol 6
# XMPP TCP traffic (port 5222) from 10.0.1.1 to 10.0.2.1, and the same
    with destAddr 10.0.2.3
destAddr 10.0.2.1 srcAddr 10.0.1.1 destPort 5222
srcAddr 10.0.1.1 destAddr 10.0.2.3 destPort 5222
```

  - Params: numQueues (=numPacketBurstifiers del padre), rulesFile, writeScalars

Edge Node - Assembler

- PacketBurstifier:
  - El módulo que hace de burstifier
  - Tiene el esquema de ensamblado:
    - Timer
    - Size
    - Num IP packets
  - y los tiempos de Offset para los BCPs
  - Params: timeout, maxSize, numPackets, minOffset, maxOffset
  - + Params: minSizeWithPadding, overflowLastPacket, cteHeader, varHeader, writeScalars
  - Param encaminamiento OBS: destLabel

Edge Node - Assembler

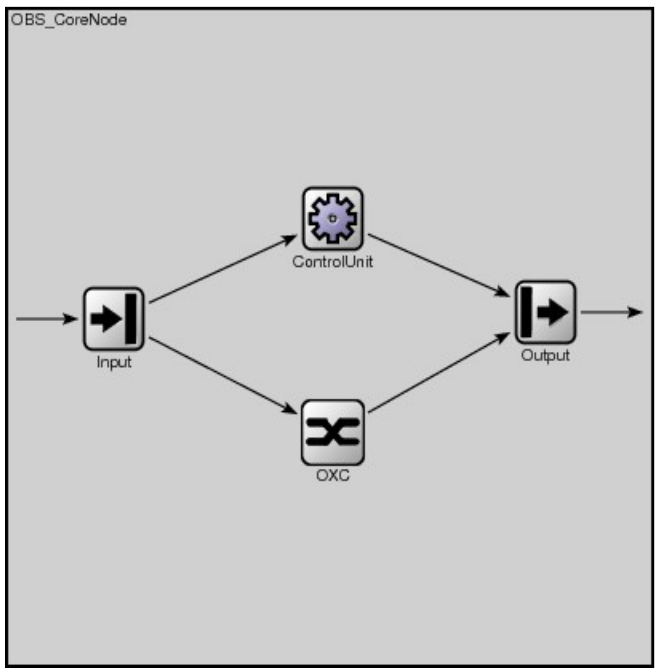
- Sender:
  - Encola y envía las ráfagas y los BCPs según el esquema de planificación implementado
  - Actualmente tiene el más simple: Horizon
    - Busca primera lambda libre, coloca ahí ráfaga, e intenta enviar BCP antes
  - Es la única parte del Edge Node que necesita conocer número de lambdas y capacidad por lambda.
  - Params: numLambdas, dataRate, BCPSize
  - Params memoria finita: maxSchedBurstSize, maxSchedBurstElems
  - + Params: guardTime, testing

Edge Node - Disassembler

- Convierte las ráfagas ópticas que salen de la red óptica en tráfico de salida
- Es necesario que le llegue el inicio/fin de ráfaga para que considere que ha llegado una ráfaga
- Actualmente no hace nada con el tráfico BCP => se podría usar para control de errores
- Params: (, colours)

Core Node

- Core Nodes: tráfico óptico en tránsito => conmutado óptico
- Se compone de 4 submódulos: input, ControlUnit, OXC y Input
- Params: numInNodes, numOutNodes, lambdasPerNodes, OEConversionDelay, EOConversionDelay, BCPProcessingDelay, dataRate, routeTableFile



Core Node - Input

- De cada fibra/enlace entrante redirige la lambda de control al ControlUnit y las lambdas de datos al OXC
- Params: numNodes, lambdasPerNode
- Supone que la primera lambda de cada fibra/enlace es la de control
- LambdasPerNode: este parámetro en el EdgeNode indicaba las lambdas de datos, aquí indica las lambdas totales (datos+control)

Core Node - Output

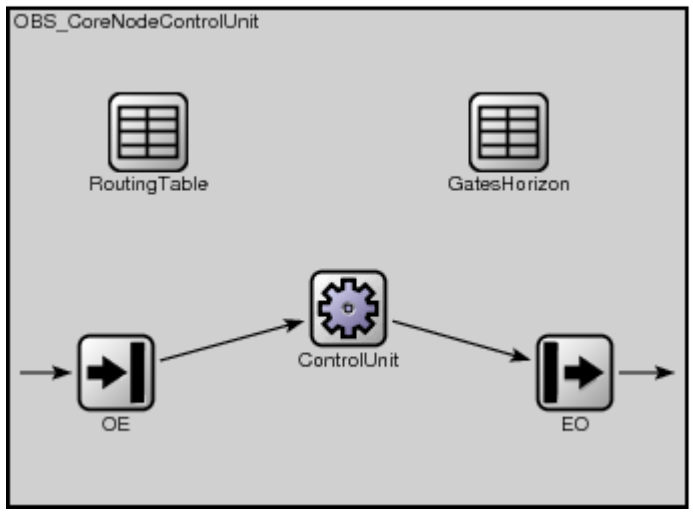
- Multiplexa en cada fibra/enlace de salida la lambda de control que viene del ControlUnit y las lambdas de datos del OXC
- Hace lo inverso de CoreNode-Input
- Params: numNodes, lambdasPerNode

Core Node - OXC

- Conmutador óptico transparente y programable
- El ControlUnit va conectando/desconectando las entradas con las salidas
- Todas las suposiciones sobre tipos de datos, señalización, capacidad de conversión óptica,... es responsabilidad del ControlUnit
- Param: switchingDelay

Core Node – Control Unit

- Se compone de 3 submodulos simples y 2 tablas de información: OE, EO, ControlUnit, RoutingTable, GatesHorizon
- Params: numOutNodes, lambdasPerNodes, routingFile
- Params procesamiento: OEConversionDelay, EOConversionDelay, processingTime, dataRate



Core Node – Control Unit

- OE: conversión BCP óptico a BCP electrónico
  - Añade información sobre fibra/enlace de entrada y tiempo de llegada
  - Param: OEConversionDelay
- EO: conversión BCP electrónico a BCP óptico
  - Crea el BCP óptico (inicio/fin)
  - Params: EOConversionDelay, dataRate
- GatesHorizon: tabla con los horizontes para poder implementar esquema Horizon
  - Params: numPorts, numLambdas => son lo mismo que numOutNodes y lambdasPerNodes de su módulo padre

Core Node – Control Unit

- RoutingTable: información de conmutación estática que se carga desde un fichero.
  - Regla: PortIn LambdaIn LabelIn PortOut LambdaOut LabelOut

```
#All bursts from port 0 will be redirected to port 2 renaming the label to 3
0 * * 2 * 3
#Bursts from port 1 and label 3 will go out to port 3 and lambda 1 preserving the
      same label
1 * 3 3 1 *
```

  - Param: routingFile
-

Core Node – Control Unit

- ControlUnit: es el módulo que decide el encaminamiento, programa el OXC y modifica el BCP electrónico.
  - Actualmente implementa el esquema más sencillo:
    - Usa info del BCP electrónico, el RoutingTable y el GatesHorizon
    - Decide si cuando llega la ráfaga alguna de las lambdas que puede usar esa ráfaga está libre y programa el OXC
    - No tiene memoria óptica, no hace reflection, total conversión óptica
  - Params: guardTime, processingTime, dataRate

Módulos extras

- DropBurst
  - Módulo para generar pérdidas en una red
  - Por ahora, pérdidas i.i.d
  - Param: dropProb
- OpticalMonitor
  - Módulo monitor => genera datos/estadísticas de las ráfagas que pasan por él
  - Está en un primer estado de implementación, la salida es:

```
* Vector 1 (Burst info)
1 <simTime> <burstifierId> <numSeq> <length> <lambda>
* Vector 2 (BCP info)
2 <simTime> <burstifierId> <numSeq>
```

  - Param:outputFile

Ejemplo

- Ejemplo...

upna Universidad Pública de Navarra Nafarroako Unibertsitate Publikoa

Sugerencias...

- ...