Table of Contents

Modelo Conceptual para QoS en la red Felix....1
Conceptual Model (Conceptual framework)....1
View of the switch....1
Priority Queues....1
Communication and data flow....1
Topology....2
What are we NOT modeling....3
Que tenemos ahora de PhaseI....3
Tasks....4
Temas a discutir....5

Modelo Conceptual para QoS en la red Felix

Conceptual Model (Conceptual framework)

View of the switch

This is the "tipical" view of a switch, only that instead of just a queue there is **PriorityQueues** module.
Note that we model **egress queues** (and not the ingress queues) because the priority queues are implemented in egress ports. Thus, the **ROUTING** (selection of output port) should be applied BEFORE the queueing.
The **Link Transport Delay** is the one that applies the "cable delay" based on the link capacity (ej: 40 Gbps). We put it here to stress the fact that a single packet can be sent at a time and the unqueuing times are given by the link speed.

plots online: https://www.draw.io/#G0B0Spz4YE9z7_RUdOZFI5M2tQdjg

Priority Queues

PriorityQueues module view. Process for each incomming packet

We will model only egress port queues as priority queues apply to egress ports.

- **Marking**: Reading any of the packet properties, it assigns a TAG to the packet. T
- **Priority Queue Select:** It maps each TAG to one of the N queues.
- **N Priority Queues**:
    - **shared static buffer:** all the queues have a static buffer size. The total buffer size for an egress port is divided equally among all queues (thus, each priority queue has a fixed buffer size)
- **Scheduling:** WRR

Especificaciones puntuales del Brocade ICX 7250: Some details here

Communication and data flow

Felix servers and SWRODs

Some things are defined regarding the **communication** between **Felix servers and SWRODs**:

1. The **communication ratio** will be some of these options: **1)** 1 Felix -> 2 SWROD **2)** 1 Felix -> 1 SWROD **3)** 2 Felix -> 1 SWROD
2. The felix servers will running all the time (independent of the run FSM)
3. They will use a **publish/subscribe** architecture: Felix servers publish which channels they provide and SWROD subscribe to channels. Once the subscriptions are made, the Felix servers will forward the fragments to the SWRODs which are subscribed (PUSH of the data). There will be no application level ACKs (yes the TCP ACKs)
4. Felix servers will **receive fragmets** from the (32?) GBT links at **100kHZ**
5. Felix servers will **gather fragments** and send them once there is **1MB** of data to be sent (**NOTA:** validar con Jorn!)

Control network

There is a big interest in NOT having a separate network for the control traffic (like there is not for the HLT network). That is the reason why there is a need to implement QoS to guarantee bandwidth limits.

The control traffic flows are the on here

There is no way at this point in time to foresee how the control traffic pattern and intensity will be. The best idea is to interpolate with the traffic from today in the HLT farm (which is not completely undestood yet).

Topology

At this point there are several things which are **not defined yet**:

1. # of Felix and SWROD servers
2. Thus the topology can change a lot and is not well-defined at all
3. The underlying technology will most probably be Ethernet, but Inifinband is also being considered.

Below there is an example idea of a possible topology

What are we NOT modeling

1. Ingress Ports: we will only model egress ports, as the priority queues are implemented in egress ports.
2. Loss-less mode: porque depende de la interaccion entre ingress/egress queues, y en caso de alcanzarse la capacidad de las ingress queue se envia un pause frame al siguiente hop. (no tenemos modelado y no se van a usar pause frames en TDAQ)
3. UC and MC queues: ¿Que son? En total serian 16 priority queues: 8 UC y 8 MC. Solo modelamos una de esas
4. Strict Priority Scheduling (SPS): "Does not serve any WRR queue until the SPS is empty"
5. Configurable buffer limits: Brocade has a shared buffer for all priority queues. But it allows to configure more/less percentage of the buffering capacity to certain queues. To start with, we will model only equal capacity in all queues
6. Rate limiting: Brocade has a module which applies rate limiting which we will not model
7. SFOs: where not included in the previous model and would require programming the SFO and PU message passing.

Que tenemos ahora de Phasel

https://twiki.cern.ch/twiki/bin/view/Main/PhaseISimpleTopology

TODOs:

- (DONE) Perform some tests to verify the model, and compare with JMT.
- (Semi-DONE) Think of a mechanism to read flow parameters from the configuration.
  We are reading each distribution parameters as usual from the config (nothing special for flows)
- Reading logging and debug setting from configuration is making initialization slow (8s). This is because it tries to read ~200 variables from Scilab. Need to think of a way to avoid this time consuming reading of variables (¿Is there a way to know all defined variables in Scilab from PowerDEVS?).

Possible modeling improvements:

- Evaluate to make the link (queue+server+sampler) a single atomic model to improve performance.
- Add a delay to the routers.
- It might be strage to think that the queue is inside the Link model...
- Add queueing policies to routers. This will require thinking the best way for the Router model to get the information of all queues (now inside the Link model) connected to it.
  This might be a problem because of putting the queue inside the link... maybe rethink..
- Now the sampling and logging can be set in configuration preatty easily. But with more models might become tricky. Develop "vectorial" configuration or something of the sort.
- Links are now implemented as coupled models . If you want a new link you copy&paste of a coupled model. But if we want to update the implementation of the link, we have to manually update all links one by one.
  In the future possible all hosts and router/switches will also be coupled models. It would be good to make updates to easier. Maybe the concept of 'coupled model class' that can be instanciated?

Tasks

En la charla con Rodri definimos algunas cosas sobre el modelo y las tareas.

Algunas tareas:

1. **Revivir el modelo de HLT** (el de examples/Matias/ATLAS-TDAQ). Hay que hacerlo compilar con los nuevos cambios y ver que funciona como esperamos.
    1. Esto YA NO VA MAS, es para la red Felix.
2. **Implementar priority queues** en los switches (como esta explicado arriba)
    1. En ppio utilizando n priority queues (implementacion con VDEVS)
    2. El marcado resolverlo de la manera mas simple: tal vez que las fuentes emitan con una determinada marca (flowID?)
3. **Implementar los nuevos nodos** (fuentes de paquetes). Hay que agregar los hosts, racks y switches de ONL, ctrl&config, histograms, etc.). Esto es trivial, la complejidad esta en al configuración (ver punto 5)
    1. Esto es configurar nuevos nodos segun la topologia
4. **Topologia configurable:** La topologia que se va a usar no esta nada definida, por lo que la simulacion deberia ser muy facil de cambiar a otra topologia. Hoy la topologia esta dada por como se conectan los modelos y los VDEVS.
5. **Routing:** Si la topologia va a cambiar, el routing tambien deberia ser flexible. Hoy el routing se hace haciendo cuentas en los VDEVS indexer (ej: (inIndex % nDCMs) % nROS).
6. **Configuración (de las fuentes):** ¿Como es el rate y patrón/distribución del trafico que generan esas aplicaciones?¿Con que nodos se comunican y con que frecuencia con cada uno?
   Esto no lo sabemos nosotros ni Eukeni. Hay que leerlo de las métricas de NETIS (que son muy poco programer-friendly, no es IS ni PBeast, estas para sacar valores hay que hacerlo casi a ojo). Por lo que vimos muy rapidamente son muy dispares, algunos dias no hay trafico, por ratos hay mucho uso, no todos los nodos son iguales (por ejemplo los nodos de monitoring de los moun es mucho mas activo que el de monitoring de CTP).
   Yo creo que vamos a tener que analizar esas métricas reales y ahí decidir como las metemos en el simulador. Se me ocurre que podemos tener que varios esceanarios (configuraciones) y probar algunos
7. **Busqueda de parámetros.** Una vez que esta todo modelado, hacer lo que quiere Eukeni: buscar cuales son los parámetros que hay que usar en los switches para que el trafico ande como se espera.

Temas a discutir

- ¿Como generar y definir la topologia?
    - Tomar la topologia de ONOS y generar un .pdm (o mejor el C++ directamente)?
- Cantidad de puertos: ¿Con VDEVS, con generador, o c++?
    - **con VDEVS: Cons:** Con VDEVS no se pueden hacer VDEVS acoplados. Eso trae varias "incomodidades". Por ejemplo si un switch es un modelo acoplado, donde cada puerto es un modelo acoplado que tiene varias colas. Para definir 30 switches la solucion es tener 1 modelos VDEVS de cada modelo atomico y jugar con los indices (hacer cuentas en C++). Cambiar la topologia es una patada y no es facil generarla. **Pros:** ya esta hecho
    - **con generador: Pros:** Con el generador puede generar la estructura que haga falta. **Cons:** para cambiar la implementacion de algun acoplado hay que cambiar el generador?
    - **con c++: Pros:** es facil de implementar (funciones addSwitch/ Link /Node). Es facil de generar. **Cons:** Se pierde la parte visual

Possible modeling improvements:

- Evaluate to make the link (queue+server+sampler) a single atomic model to improve performance.
- Add a delay to the routers.
- It might be strage to think that the queue is inside the Link model...
- Add queueing policies to routers. This will require thinking the best way for the Router model to get the information of all queues (now inside the Link model) connected to it.

- This might be a problem because of putting the queue inside the link... maybe rethink..
- Now the sampling and logging can be set in configuration preatty easily. But with more models might become tricky. Develop "vectorial" configuration or something of the sort.
- Links are now implemented as coupled models . If you want a new link you copy&paste of a coupled model. But if we want to update the implementation of the link, we have to manually update all links one by one.
- In the future possible all hosts and router/switches will also be coupled models. It would be good to make updates to easier. Maybe the concept of 'coupled model class' that can be instanciated?

-- MatiasAlejandroBonaventura - 2016-09-06

This topic: Main > TDAQSimulationPhaseI_QoSConceptual
Topic revision: r4 - 2016-09-09 - MatiasAlejandroBonaventura