

Minuta 16/07 - Paper CACIDI CAC

Monday, July 16, 2018 9:52 PM

- 1) Investigar cómo se implementa CAC en Asteriks
- 2) Comparación entre CAC tradicional y modelo CAC SDN (*)
- 3) Planteo de arquitectura propuesta
 - a. Módulo 1: Conteo de sesiones concurrentes en Asteriks y API
 - b. Módulo 2: Comunicación entre API ASTKS y API Controlador
 - c. Módulo 3: Desarrollo de la lógica CAC en el controlador
 - d. Módulo 4: Comunicación entre aplicación CAC y API Controlador
 - e. Módulo 5: Comunicación entre API Controlador y OpenFlow

(*) Resaltar ventajas y desventajas del modelo SDN. Trabajar sobre argumento de mayor interoperabilidad y vendor agnóstico

Que comandos OpenFlow deberíamos utilizar para cortar un flujo de comunicación?

Objetivo: Implementar mecanismo de Call Admission Control (CAC) utilizando northbound APIs y OpenFlow sobre plataforma SIP. Tomar indicadores de BW, Jitter y Packet Loss como parámetros para medir performance y realizar comparaciones sobre una arquitectura no SDN. El desarrollo contemplará modelo de APIs, controlador y comunicación southbound interface con OpenFlow (1.5 y 1.6) [<https://www.opennetworking.org/wp-content/uploads/2014/10/openflow-switch-v1.5.1.pdf>].

Herramientas a utilizar:

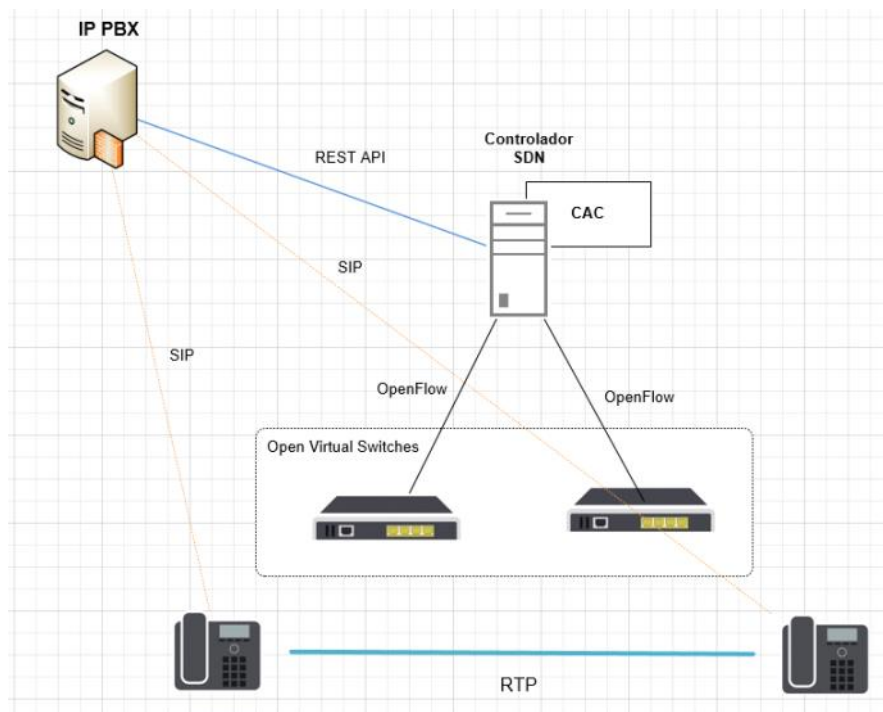
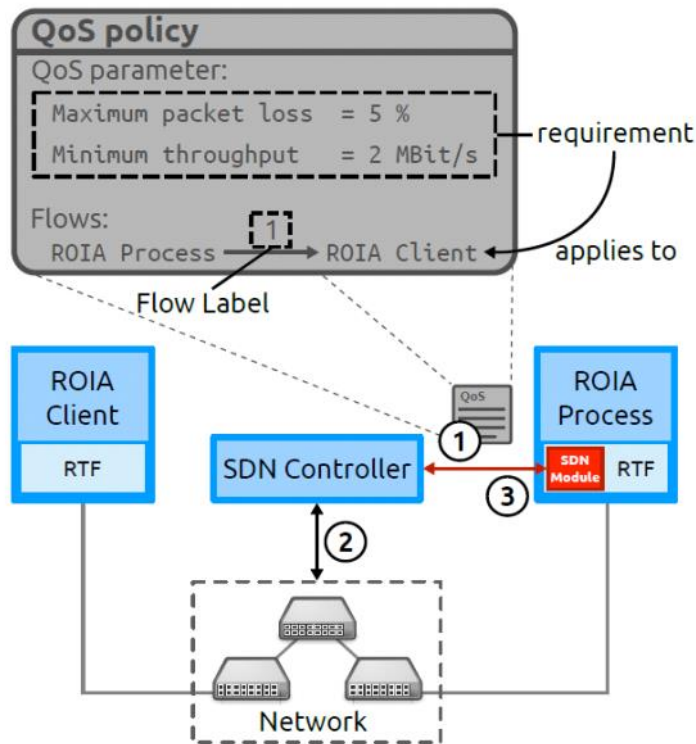
- GNS3 (<https://gns3.com/>)
- Mininet (<http://mininet.org/>)
- Open Virtual Switch (<https://www.openvswitch.org/>)
- Onos (<https://onosproject.org/>)
- Floodlight (<http://www.projectfloodlight.org/floodlight/>)
- Python/Java para desarrollo de API y aplicación en controlador

Estuve viendo que ONOS (controlador de la ONF) ya tiene una API que permite manipular flujos e implementar QoS (<https://wiki.onosproject.org/display/ONOS/Flow+Rules>)

Papers de referencia:

- [1] Quality of Service (QoS) in Software Defined Networking (SDN) A survey
- [2] Dynamic QoS on SIP Sessions Using OpenFlow
- [3] Using Software-Defined Networking for Real Time Internet Applications

Arquitectura de referencia:



Ejercitación y testing:

Creo que sería bueno replicar algunos resultados según las prácticas de este proyecto:

- <https://github.com/yanlisa/reproducibility/wiki/Reproducing-Network-Research>

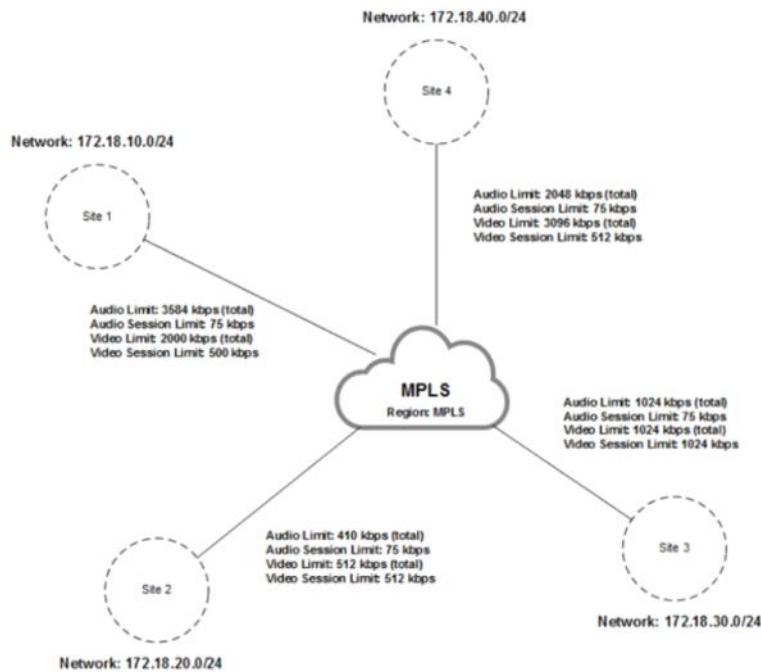
Pienso que dará expertise a la hora de controlar mininet y otro software de testbed y simulación.

Casos de Uso:

Se tratan en detalle dos casos de uso por ser los más representativos en el uso de CAC y los más implementados por proveedores de servicios. SDN jugaría un rol fundamental en este escenario para implementar CAC de forma dinámica bajo demanda de las

aplicaciones SIP que utilizan la red. Por otro lado, con este mismo mecanismo podrían implementarse tags/configuraciones sobre colas QoS que permitan priorizar flujos en tiempo real, este es una posibilidad no existente en redes no SDN.

MPLS: Se implementa Call Admission Control en un escenarios de conectividad MPLS. Multiples sitios distribuidos en distintas regiones conectados sobre una red MPLS en donde CAC debe ejecutar restricciones basandose en información de subredes origen/destino asociadas a cada sitio. Por otro lado, el vinculo que utilizará un sitio para comunicarse con el resto será compartido.



WAN: Se implementa CAC en un escenario de conectividad punto a punto o enlaces WAN entre sitios. En este caso las distintas locaciones tendrán BW específico para conectarse a cada uno de los sitios por lo que las políticas deberán contemplar estas diferencias.

