QoS in OpenStack with SDN

SDN Workshop (Group)
Philipp Aeschlimann - ICCLab Oct. 2013
I migrated our northern data center to the cloud.

But the cloud stopped working and I can't find the phone number for our cloud guy.

So...whatever.

You lost our data center?

That's one way to look at it.
What this talk is about
Where OpenStack implements SDN

- OpenStack is the software used to provide cloud functionality
- The ultimate goal in OpenStack is isolation
  - isolate resources for every tenant
- SDN is used to offer NaaS to the tenant's
  - But it can also be an implementation based on Linux VLAN
- This "can" be done with the SDN paradigm
  - The enablement "can" be OpenFlow, but also e.g. Open vSwitch
- physical layer is not addressed
What is it - NaaS
Why do we want to control the physical network

- There will be new opportunities
  - Live migration of virtual machines
  - Monitoring tasks
  - Simplify the management of network devices
  - Change the network dynamically as alternative to live migration
  - Federated SDN or bring pieces of different data-centers together
Network Layout at the ICCLab
Virtual big-switch

- Making multiple switches looking as one has several advantages
- We don't care anymore, where a cable is connected
- We can exchange devices directly
- It makes monitoring more simple
controlling the bandwidth

- QoS is always divided in two main goals:
  - service differentiation
    - how to identify traffic from youtube and a video call
  - performance assurance
    - how to actually assure the needed performance
- This brings us to SDN and OpenFlow as the enablement
  - service differentiation with OFP-match
  - performance assurance with OFP-meter-bands
controlling the bandwidth

- Is made by percentage values
  - e.g. Reserve 5% of the total Bandwidth for my traffic
- Arranged in queues but not the OFP-queues!
  - OFP-queues are too static for use
  - General idea is based on rate-limiters
  - easy to use and simple mechanism
  - rate-limit all traffic and burst to the back-plane size if more bandwidth is available
  - the common-queue is always installed and rate-limits traffic to the backplane
controlling the bandwidth

common queue

resize the common queue and add the new one
controlling the bandwidth

Priorized Traffic goes to the forwarding table

Rate-limited Traffic goes to the QoS table, afterward to the forwarding table
Future work

- Automatic resetting the backplane-size by monitoring value
- Automatic initial backplane-size by SNMP
- Experiments with different parameters in the cloud
- Using the REST-API from the QoS application by OpenStack monitoring
- Connecting RYU to the automated Framework
Questions/Discussion

cloudcomp.ch