Inter-Datacenter WAN with centralized TE using SDN and OpenFlow

Background

Google offers many different services - Google Web Search, Google+, Gmail, YouTube, Google Maps - that are core to Google's mission to organize the world's information and make it universally accessible and useful. In order to serve a global user base with speed and high availability, large volumes of data need to be moved from one region to another making these applications/services very WAN-intensive.

When we look at the management, cost and performance of infrastructure, economies of scale have worked out well for storage and compute components but not necessarily for the WAN. This is for a variety of reasons such as non-linear complexity and interaction of networking devices, manual configuration and management and non-standard vendor configuration APIs.

We require a WAN where economies of scale deliver cost efficiency, higher performance, better fault tolerance and manageability. We need to manage the WAN as a fabric and not as a collection of individual boxes.

Overview of Google's SDN solution

Google's WAN is organized as two backbones – an Internet facing (I-scale) network that carries user traffic and an internal (G-scale) network that carries traffic between datacenters. These two backbones have very different requirements and traffic characteristics. It is the G-scale network in which Google has deployed an OpenFlow powered Software Defined Networking (SDN) solution.

"SDN is a pragmatic approach to reducing the complexity and management of networks. While it is still early to declare success, Google's experience with SDN and OpenFlow in the WAN show that it is ready for real world use."
— Urs Hölzle, Fellow and SVP of Technical Infrastructure, Google

When Google started this effort there was no network device available that had OpenFlow support and could meet our scale requirements - so Google built its own network switch from merchant silicon and open source routing stacks with OpenFlow support. The list of features developed was minimal but sufficient for Google's use case.

Each site is comprised of multiple switch chassis to provide scalability (multiple terabits of bandwidth) and fault tolerance. Sites are connected together and multiple OpenFlow controllers communicate with the switches using OpenFlow. Multiple controllers ensure that there is no single point of failure.

On this WAN fabric we built a centralized traffic engineering (TE) service. The service collects real-time utilization metrics and topology data from the underlying network and bandwidth demand from applications/services. With this data, it computes the path assignments for traffic flows and then programs the paths into the switches using OpenFlow. In the case of changing demand or network events, the service re-computes path assignments and reprograms the switches.

We have the above SDN solution deployed in the G-scale WAN today where it is serving datacenter-to-datacenter traffic.

Network utilization up to 95%. This is unheard of in the industry.

We started this effort in Jan 2010 and by early 2012 all our datacenter backbone traffic was being carried by the OpenFlow powered network. Centralized TE was rolled out globally in two months.

Benefits of SDN

- **Unified view of the network fabric** With SDN we get a unified view of the network, simplifying configuration, management and provisioning.
- **High utilization** Centralized traffic engineering provides a global view of the supply and demand of network resources. Managing end-to-end paths with this global view results in high utilization of the links.
- **Faster failure handling** Failures whether it be link, node or otherwise are handled much faster. Furthermore, the systems converge more rapidly to target optimum and the behavior is predictable.
- **Faster time to market/deployment** With SDN, better and more rigorous testing is done ahead of rollout accelerating deployment. The development is also expedited as only the features needed are developed.
- **Hitless upgrades** The decoupling of the control plane from the forwarding/data plane enables us to perform hitless software upgrades without packet loss or capacity degradation.
- **High fidelity test environment** The entire backbone is emulated in software which not only helps in testing and verification but also in running "what-if" scenarios.
- **Elastic compute** Compute capability of network devices is no longer a limiting factor as control and management resides on external servers/controllers. Large-scale computation, path optimization in our case, is done using the latest generation of servers.

Challenges

- **OpenFlow protocol** The OpenFlow protocol is in its infancy and is bare bones. However, as our deployment shows, it is good enough for many network applications.
- **Fault tolerant OpenFlow controllers** To provide fault tolerance, multiple OpenFlow controllers must be provisioned. This requires handling master election and partitions between the controllers.
- **Partitioning functionality** It is not very clear what functionality should reside in the network devices and what should reside in external controllers. Configuration of functionality resident in the network device remains an open question.
- **Flow programming** For large networks, programming of individual flows can take a long time.

Summary

Google's datacenter-to-datacenter WAN successfully runs on an SDN and OpenFlow enabled network. It is the largest production network at Google. SDN and OpenFlow have improved manageability, performance, utilization and cost-efficiency of the WAN.