Paper Summary
HA 2.a

Damodar Sahasrabudhe

School of Computing, University of Utah

September 8, 2015

1 Title of paper: B4: Experience with a Globally-Deployed Software Defined WAN

Paper discussed in this summary is "B4: Experience with a Globally-Deployed Software Defined WAN"[1].

1.1 First pass information

1. *Category:* Paper discusses implementation of SDN for WAN used by Google.

2. *Context:* Paper is in area of SDN implementation. It discusses about Design, Implementation, Evaluation and learning lessons.

3. *Assumptions:* Basic assumption in behind implementation is that networking requirements for Googles data centers data centers will not change in future. These requirements are - massive bandwidth requirements deployed to a modest number of sites, elastic traffic demand that seeks to maximize average bandwidth, and full control over the edge servers and network.

4. *Contributions:*

   (a) Paper and implementation's huge success is technology demonstrator for Open Flow and SDN.

   (b) B4 links to run at near 100

   (c) Even if SDN fails, servers can still communicate by legacy routing mechanisms adds more stability.

   (d) Allows available bandwidth to be used 'elastically' for different applications as and when needed

   (e) Establishes 'priority' routing for control plane, in case of network failure

1.2 Second pass information

Summary:

- Introduction and Background: Paper starts with explaining how traditional WAN architecture uses overprovisioning for reliable service and later explains unique requirements for Google's network connecting data centers - Elastic bandwidth demands, Moderate number of sites with very high bandwidth requirement, ability to prioritize applications and saving cost overheads needed for bandwidth overprovisioning, fault tolerance, scalability.

- Design: B4 has 3 layered architecture - switch hardware layer with simple rules used for forwarding traffic, Site Controller layer containing Open Flow Controllers and the Global layer housing Traffic Engineering Servers and gateways. Each of the layer is explained in subsequent sections.

1. Switch Design: Google built there own hardware switches with peculiar characteristics to suit the requirements. With SDN transmission rates can be adjusted. It reduces packet drops thereby avoiding need for very large buffers in switch. As number of data centers are limited, switches did not need large forwarding tables. Google custom made switches that would provide low level control to software through interfaces. Use of high-radix switches further reduced number of switches needed. Switches are connected by Clos topology. OpenFlow Agent would send any packet, destined outside of ingress chip, to 'spine'.
2. Network Control: Paxos algorithm is used to find out 'leader' in this distributed system. When a majority of the Paxos servers detect a failure, they elect a new leader among the remaining set of available servers. NIB contains the current state of the network with respect to topology, trunk configurations, and link status (operational, drained, etc.).
3. Routing: Quagga stack is selected for routing using BGP/ISIS. The custom component "Routing Application Proxy (RAP)" is used to interchange : route updates, flow of routing protocol packets between Quagga and switches and vice a versa. RAP translates from RIB entries forming a network-level view of global connectivity to the low-level hardware tables used by theOpenFlow data plane.
4. Traffic Engineering (TE): TE ensures fair allocation of bandwidth among competing applications. Bandwidth Enforcer also aggregates bandwidth functions across multiple applications. "Tunnel Group Generation" allocates bandwidth to FGs based on demand and priority. It allocates edge capacity among FGs according to their bandwidth function such that all competing FGs on an edge either receive equal fair share or fully satisfy their demand. A bottleneck edge is not further used for TG generation, and traffic is bypassed without using bottlenecked tunnels.
5. TE Protocol and Openflow: Each site in the tunnel path maintains per-tunnel forwarding rules. Source site switches encapsulate the packet with an outer IP header whose destination IP address uniquely identifies the tunnel. The outer destination-IP address is a tunnel-ID rather than an actual destination. Each site maintains Tunnel Engineering Database (TED).TED maintains a key-value data store for global Tunnels, Tunnel Groups, and Flow Groups. TED is kept synchronized with all OFCs
6. Evaluation: As mentioned in the paper, B4 is performing beyond expectations - various graphs point that failures are reduced over the period of time. Even if a link fails, other traffic is taken over by other switches. The busiest B4 edges constantly run at near 100
7. Experience from Outage: Few lessons learned from failures - OFA should be asynchronous and multi-threaded for more parallelism, additional performance profiling and reporting show signs of impending failures, TE server server must be adaptive to failed/unresponsive OFCs when modifying TGs that depend on creating new Tunnels, Multiple, sequenced manual operations should not be involved for virtually any management operation.

1.3 Third pass information

- *Strengths:*
  - Excellent paper with technical details, graphs and explanations.
  - Technology demonstrator to showcase effectiveness of SDN.
  - Improved WAN performance at reduced cost.

  - Fault tolerance using secondary servers and automatic take over protocols ensure reliability.
  - Backward compatibility ensures seamless operation using conventional network protocol in case SDN fails.
  - Layered architecture provides ease of implementation, operation and maintenance.
  - Traffic Engineering would provide "central authority" to direct traffic.
  - Prioritization of traffic will help recovery in case of failures or bottlenecks.

- *Weaknesses:*
  - Section 4.1 regarding TE architecture should have been more elaborate.

- *Questions:* More elaboration on concepts of Tunnel, TG and FG will help better understanding.

- *Interesting citations:*
  1. Paxos [2]
  2. ECMP Hashing [3]
  3. Onix [4]
  4. Fair Allocation [5]

- *Possible improvements:*
  - For testing new configuration Google is using copies of real deployment. They can create some subnet under live network using openFlow and use it for development, research and testing.

- *Future work:* Similar "custom" implementations could be done for many organizations as per custom demands - not necessary for data centers but also for day today work. It can afford organizations reliability at reduced cost.

References

[1] B4: Experience with a Globally-Deployed Software Defined WAN

[2] Paxos Made Live: an Engineering Perspective. In Proc. of the ACM Symposium on Principles of Distributed Computing (New York, NY, USA),ACM, pp.

[3] ECMP Hashing - Thaler, D. Multipath Issues in Unicast and Multicast Next-Hop Selection. RFC 2991, IETF, 2000.

[4] Onix: a Distributed Control Platform for Large-scale Production Networks. In Proc. OSDI (2010)

[5] Upward Max Min Fairness. In INFOCOM (2012)