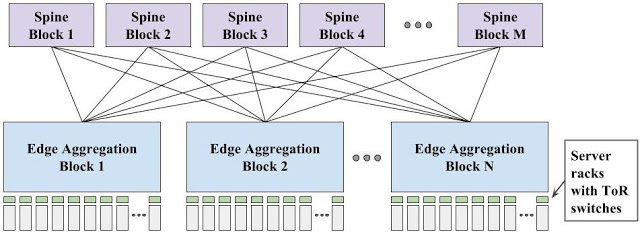
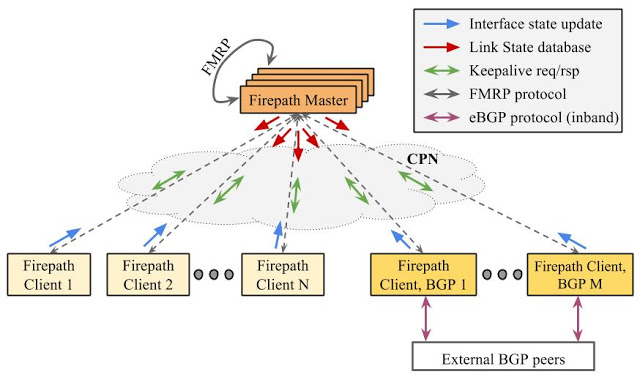
Pursuing Google’s mission of organizing the world’s information to make it universally accessible and useful takes an enormous amount of computing and storage. In fact, it requires coordination across a [warehouse-scale computer](http://www.morganclaypool.com/doi/abs/10.2200/S00516ED2V01Y201306CAC024). Ten years ago, we realized that we could not purchase, at any price, a datacenter network that could meet the combination of our scale and speed requirements. So, we set out to build our own datacenter network hardware and software infrastructure. Today, at the [ACM SIGCOMM conference](http://conferences.sigcomm.org/sigcomm/2015/), we are presenting a [paper](http://static.googleusercontent.com/media/research.google.com/en/pubs/archive/43837.pdf) with the technical details on five generations of our in-house data center network architecture. This paper presents the technical details behind a [talk](https://www.youtube.com/watch?v=FaAZAII2x0w) we presented at [Open Network Summit](http://opennetsummit.org/conference/) a few months ago.  
  
From relatively humble beginnings, and after a misstep or two, we’ve built and deployed five generations of datacenter network infrastructure. Our latest-generation Jupiter network has improved capacity by more than 100x relative to our first generation network, delivering more than 1 petabit/sec of total *bisection bandwidth*. This means that each of 100,000 servers can communicate with one another in an arbitrary pattern at 10Gb/s.  
  
Such network performance has been tremendously empowering for Google services. Engineers were liberated from optimizing their code for various levels of bandwidth hierarchy. For example, initially there were painful tradeoffs with careful data locality and placement of servers connected to the same top of rack switch versus correlated failures caused by a single switch failure. A high performance network supporting tens of thousands of servers with flat bandwidth also enabled individual applications to scale far beyond what was otherwise possible and enabled tight coupling among multiple federated services. Finally, we were able to substantially improve the efficiency of our compute and storage infrastructure. As quantified in this [recent paper](http://static.googleusercontent.com/media/research.google.com/en/pubs/archive/43438.pdf), scheduling a set of jobs over a single larger domain supports much higher utilization than scheduling the same jobs over multiple smaller domains.  
  
Delivering such a network meant we had to solve some fundamental problems in networking. Ten years ago, networking was defined by the interaction of individual hardware elements, e.g., *switches*, speaking standardized protocols to dynamically learn what the network looks like. Based on this dynamically learned information, switches would set their forwarding behavior. While robust, these protocols targeted deployment environments with perhaps tens of switches communicating between between multiple organizations. Configuring and managing switches in such an environment was manual and error prone. Changes in network state would spread slowly through the network using a high-overhead broadcast protocol. Most challenging of all, the system could not scale to meet our needs.  
  
We adopted a set of principles to organize our networks that is now the primary driver for networking research and industrial innovation, [*Software Defined Networking (SDN)*](https://www.opennetworking.org/sdn-resources/sdn-definition). We observed that we could arrange emerging merchant switch silicon around a Clos topology to scale to the bandwidth requirements of a data center building. The topology of all five generations of our data center networks follow the blueprint below. Unfortunately, this meant that we would potentially require 10,000+ individual switching elements. Even if we could overcome the scalability challenges of existing network protocols, managing and configuring such a vast number of switching elements would be impossible.

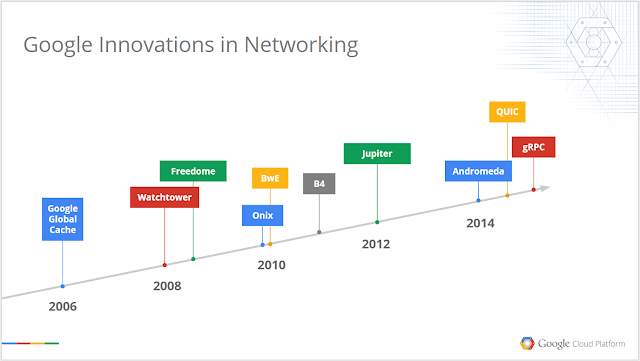
[](https://4.bp.blogspot.com/-ATbW_sscS1A/VdOXtdWTdeI/AAAAAAAAAtQ/2Il1VTSRpZM/s1600/image01.jpg)

So, our next observation was that we knew the exact configuration of the network we were trying to build. Every switch would play a well-defined role in a larger-ensemble. That is, from the perspective of a datacenter building, we wished to build a logical building-scale network. For network management, we built a single configuration for the entire network and pushed the single global configuration to all switches. Each switch would simply pull out its role in the larger whole. Finally, we transformed routing from a pair-wise, fully distributed (but somewhat slow and high overhead) scheme to a logically-centralized scheme under the control of a single dynamically-elected master as illustrated in the figure below from our paper.

[](https://1.bp.blogspot.com/-q1qsUqczFuY/VdOX019qMYI/AAAAAAAAAtY/iidETuUeUv4/s1600/image00.jpg)

Our SIGCOMM paper on Jupiter is one of four Google-authored papers being presented at that conference. Taken together, these papers show the benefits of embedding research within production teams, reflecting both collaborations with PhD students carrying out extended research efforts with Google engineers during internships as well as key insights from deployed production systems:

* Our work on [Bandwidth Enforcer](http://static.googleusercontent.com/media/research.google.com/en/pubs/archive/43838.pdf) shows how we can allocate wide area bandwidth among tens of thousands of individual applications based on centrally configured policy, substantially improving network utilization while simultaneously isolating services from one another.
* [Condor](http://static.googleusercontent.com/media/research.google.com/en/pubs/archive/43839.pdf) addresses the challenges of designing data center network topologies. Network designers can specify constraints for data center networks; Condor efficiently generates candidate network designs that meet these constraints, and evaluates these candidates against a variety of target metrics.
* Congestion control in datacenter networks is challenging because of tiny buffers and very small round trip times. [TIMELY](http://static.googleusercontent.com/media/research.google.com/en/pubs/archive/43840.pdf) shows how to manage datacenter bandwidth allocation while maintaining highly responsive and low latency network roundtrips in the data center.

[](https://1.bp.blogspot.com/-106gHg3oVXI/VdOX-vAeAJI/AAAAAAAAAtg/cNwVAim66AU/s1600/image02.png)

These efforts reflect the latest in a long series of substantial Google contributions in networking. We are excited about being increasingly open about results of our research work: to solicit feedback on our approach, to influence future research and development directions so that we can benefit from community-wide efforts to improve networking, and to attract the next-generation of great networking thinkers and builders to Google. Our focus on [Google Cloud Platform](https://cloud.google.com/) further increases the importance of being open about our infrastructure. Since the same network powering Google infrastructure for a decade is also the underpinnings of our Cloud Platform, all developers can leverage the network to build highly robust, manageable, and globally scalable services.