Google™ A Software Defined WAN Architecture
Cloud Computing Requires Massive Wide-Area Bandwidth

- Low latency access from global audience and highest levels of availability
  - Vast majority of data migrating to cloud
  - Data must be replicated at multiple sites
- WAN unit costs decreasing rapidly
  - But not quickly enough to keep up with even faster increase in WAN bandwidth demand
WAN Cost Components

- **Hardware**
  - Routers
  - Transport gear
  - Fiber

- **Overprovisioning**
  - Shortest path routing
  - Slow convergence time
  - Maintain SLAs despite failures
  - No traffic differentiation

- **Operational expenses/human costs**
  - Box-centric versus fabric-centric views
Why Software Defined WAN

- Separate hardware from software
  - Choose hardware based on necessary features
  - Choose software based on protocol requirements
- Logically centralized network control
  - More deterministic
  - More efficient
  - More fault tolerant
- Automation: Separate monitoring, management, and operation from individual boxes
- Flexibility and Innovation

Result: A WAN that is more efficient, higher performance, more fault tolerant, and cheaper
Google's Software Defined WAN
A Warehouse-Scale-Computer (WSC) Network
Google's WAN

- Two backbones
  - I-Scale: Internet facing (user traffic)
  - G-Scale: Datacenter traffic (internal)

- Widely varying requirements: loss sensitivity, topology, availability, etc.

- Widely varying traffic characteristics: smooth/diurnal vs. bursty/bulk
G-Scale Network Hardware

- Built from merchant silicon
  - 100s of ports of nonblocking 10GE
- OpenFlow support
- Open source routing stacks for BGP, ISIS
- Does not have all features
  - No support for AppleTalk...
- Multiple chassis per site
  - Fault tolerance
  - Scale to multiple Tbps
G-Scale WAN Deployment

- Multiple switch chassis in each domain
  - Custom hardware running Linux
- Quagga BGP stack, ISIS/IBGP for internal connectivity
Mixed SDN Deployment

Data Center Network

Cluster Border Router

EBGP

IBGP/ISIS to remote sites

(not representative of actual topology)
Mixed SDN Deployment

Data Center Network

Cluster Border Router

Quagga
OFC
Paxos
Glue
Paxos

EBGP

IBGP/ISIS to remote sites
Mixed SDN Deployment

Data Center Network

Cluster Border Router

IBGP/ISIS to remote sites

EBGP

Quagga

OFC

Paxos

Glue

Paxos

OFA

IBGP/ISIS to remote sites

OFA

OFA

OFA

OFA
Mixed SDN Deployment

- SDN site delivers full interoperability with legacy sites.

**Diagram:**
- Data Center Network
- Cluster Border Router
- EBGP
- IBGP/ISIS to remote sites
- Quagga
- OFC
- Paxos
- Glue
- Paxos

**Network Components:**
- Quagga
- OFC
- Paxos
- Glue
- Paxos
Mixed SDN Deployment

- Ready to introduce new functionality, e.g., TE
Bandwidth Broker and Traffic Engineering
High Level Architecture

TE and B/w Allocation

B/W Broker

TE Server

Collection / Enforcement

SDN Gateway to Sites

SDN WAN (N sites)

Traffic sources for WAN

Control Plane

Data Plane
High Level Architecture

TE and B/w Allocation

- B/W Broker
- TE Server
- Collection / Enforcement

SDN Gateway to Sites

Traffic sources for WAN

SDN WAN (N sites)

Control Plane

Data Plane

SDN API
Bandwidth Broker Architecture

Admin Policies
Network Model
Global Broker
Global demand to TE-Server

Data Center
Site Broker
Data Center
Site Broker

Usage
Limits
(optional)
High Level Architecture

Traffic sources for WAN

B/W Broker

TE Server

Collection / Enforcement

SDN API

SDN Gateway to Sites

SDN WAN (N sites)

Control Plane

Data Plane
TE Server Architecture

Demand Matrix
\{src, dst --> utility curve \}

Abstract Path Assignment
\{src, dst --> paths and weights \}

Global Broker

Flow Manager

Path Allocation Algorithm

Path Selection

Topology Manager

Site level edges with
RTT and Capacity

Per Site Path
Manipulation Commands

TE Server

Gateway

Interface up/down
status

OFC S1

OFC Sn

Devices

Devices
High Level Architecture

Control Plane

Data Plane

TE and B/w Allocation

SDN Gateway to Sites

SDN API

B/W Broker

TE Server

Collection / Enforcement

Traffic sources for WAN

SDN WAN (N sites)
Controller Architecture

TE Server / SDN Gateway

Routing (Quagga)

Topo / routes

Tunneling App

TE ops

OFC

Flows

Flows

OFA

HW Tables

Switches in DC 1
Controller Architecture

- TE Server
- SDN Gateway
- Site 1
- Site 2
- Site 3
- non-TE (ISIS) path
- Site level TE path
Sample Utilization
Benefits of Aggregation
Convergence under Failures

Without TE: Failure detection and convergence is slower:
  * Delay 'inside' TE $\ll$ timers for detecting and communicating failures (in ISIS)
  * Fast failover may be milliseconds, but not guaranteed to be either accurate or "good"

no-TE: traffic drop $\sim$ 9 sec
with-TE: traffic drop $\sim$ 1 sec
G-Scale WAN History

Exit testing "opt in" network

SDN rollout

SDN fully Deployed

Central TE Deployed
Range of Failure Scenarios

Potential failure condition

* indicates mastership
## Trust but Verify: Consistency Checks

<table>
<thead>
<tr>
<th>TE View</th>
<th>OFC View</th>
<th>Is Valid</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clean</td>
<td>Clean</td>
<td>yes</td>
<td>Normal operation.</td>
</tr>
<tr>
<td>Clean</td>
<td>Dirty</td>
<td>no</td>
<td>OFC remains dirty forever</td>
</tr>
<tr>
<td>Clean</td>
<td>Missing</td>
<td>no</td>
<td>OFC will forever miss entry</td>
</tr>
<tr>
<td>Dirty</td>
<td>Dirty</td>
<td>yes</td>
<td>Both think Op failed</td>
</tr>
<tr>
<td>Dirty</td>
<td>Clean</td>
<td>yes</td>
<td>Op succeeded but response not yet received by TE</td>
</tr>
<tr>
<td>Dirty</td>
<td>Missing</td>
<td>yes</td>
<td>Op issued but not received by OFC</td>
</tr>
<tr>
<td>Missing</td>
<td>Clean</td>
<td>no</td>
<td>OFC has extra entry, and will remain like that</td>
</tr>
<tr>
<td>Missing</td>
<td>Dirty</td>
<td>no</td>
<td>(same as above)</td>
</tr>
</tbody>
</table>
Implications for ISPs

- Dramatically reduce the cost of WAN deployment
  - Cheaper per bps in both CapEx and OpEx
  - Less overprovisioning for same SLAs
- Differentiator for end customers
  - Less cost for same BW or more BW for same cost
- Possible to deploy incrementally in pre-existing network
  - Leveraging known techniques for delivering any new functionality
Conclusions

- Dramatic growth in WAN bandwidth requirements
  - Every 10x, something breaks
  - Existing software/hardware architectures make it impractical to deliver cheap bandwidth globally
- Software Defined Networking enables
  - Separation of hardware from software
  - Efficient logically centralized control/management
  - Innovation and flexibility
- Deployment experience with Google's global SDN production WAN
  - It's real and it works
  - This is just the beginning...
Thank you!