Software Defined Networking at Scale

Bikash Koley
on behalf of Google Technical Infrastructure

BTE 2014

Software Defined Networking at Scale

Bikash Koley
on behalf of Google Technical Infrastructure

~~Software Defined~~ Networking at Google

Bikash Koley
on behalf of Google Technical Infrastructure

Google | google.com/datacenters

Software Defined Networks require Software Defined Operations

Google made great progress in SDN data and control plane

It is time to transform the management plane with the industry!

Google

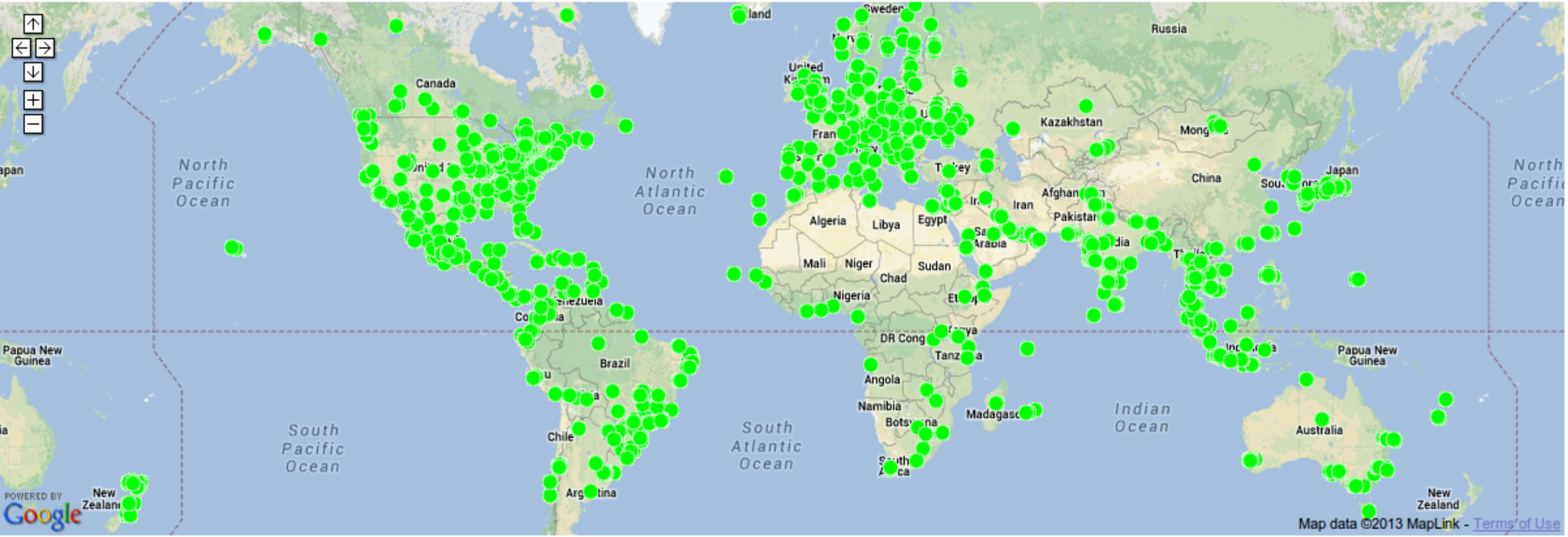
Warehouse Scale Computers

Source: Google, 2012

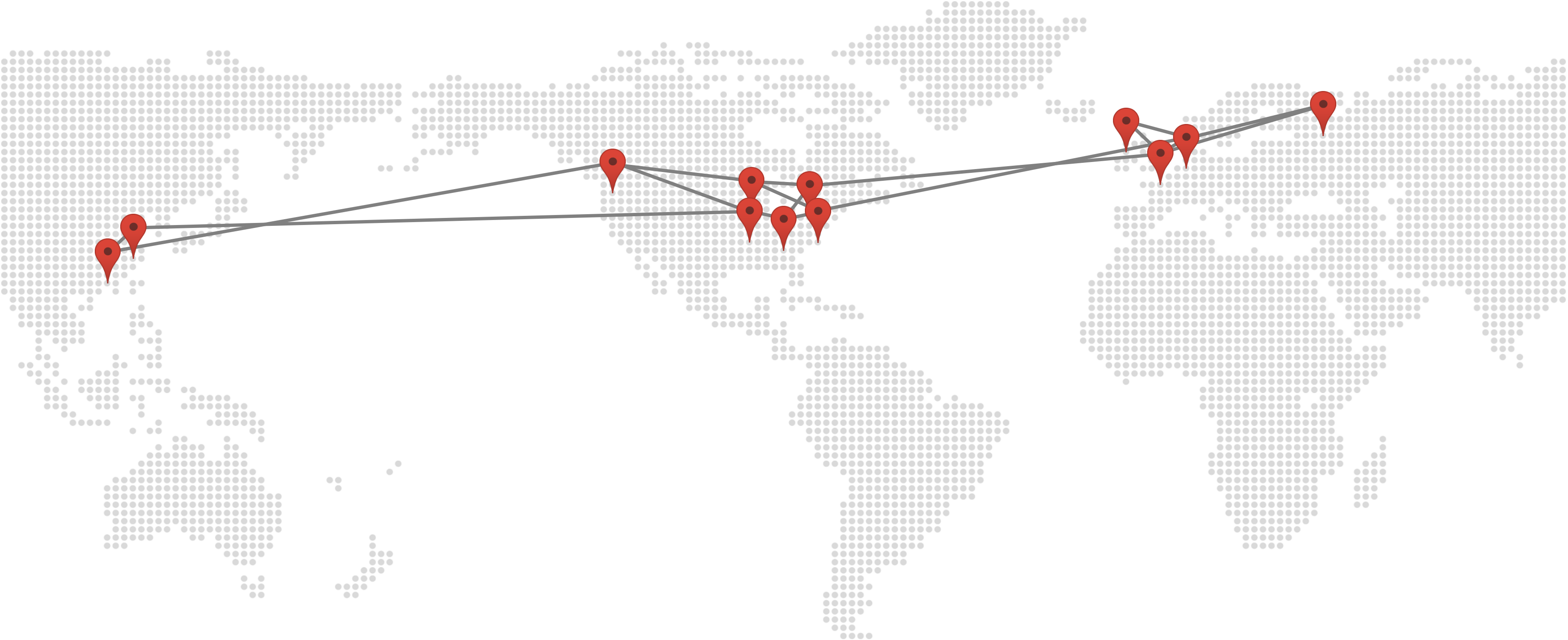
100 Billion

searches per month on google.com

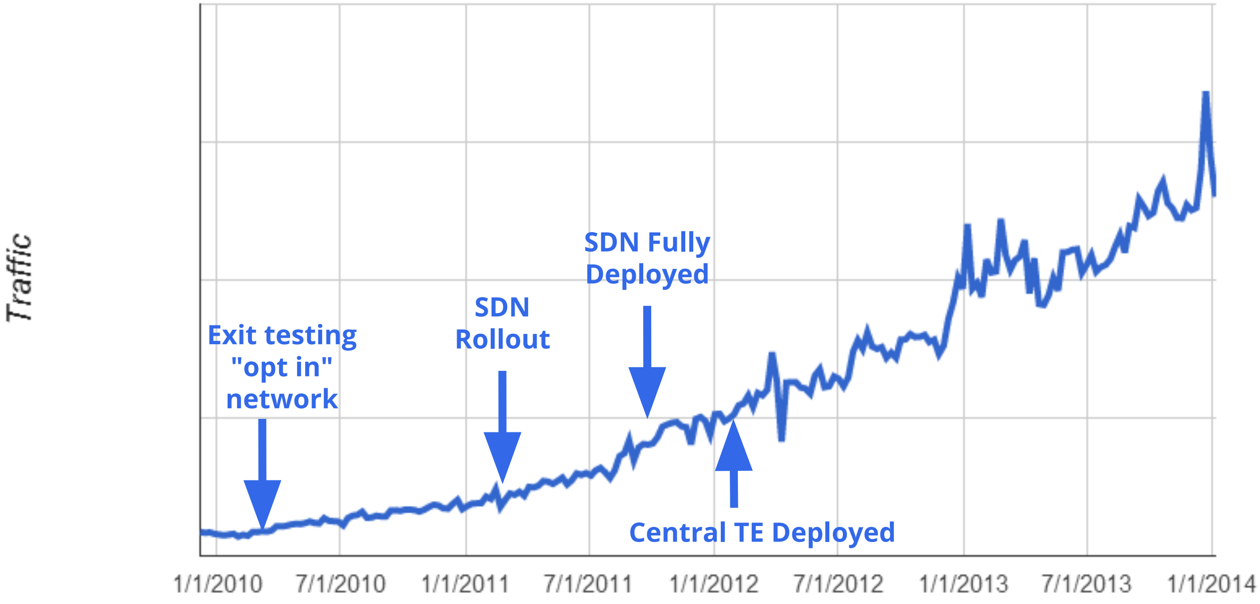
Google's Global CDN



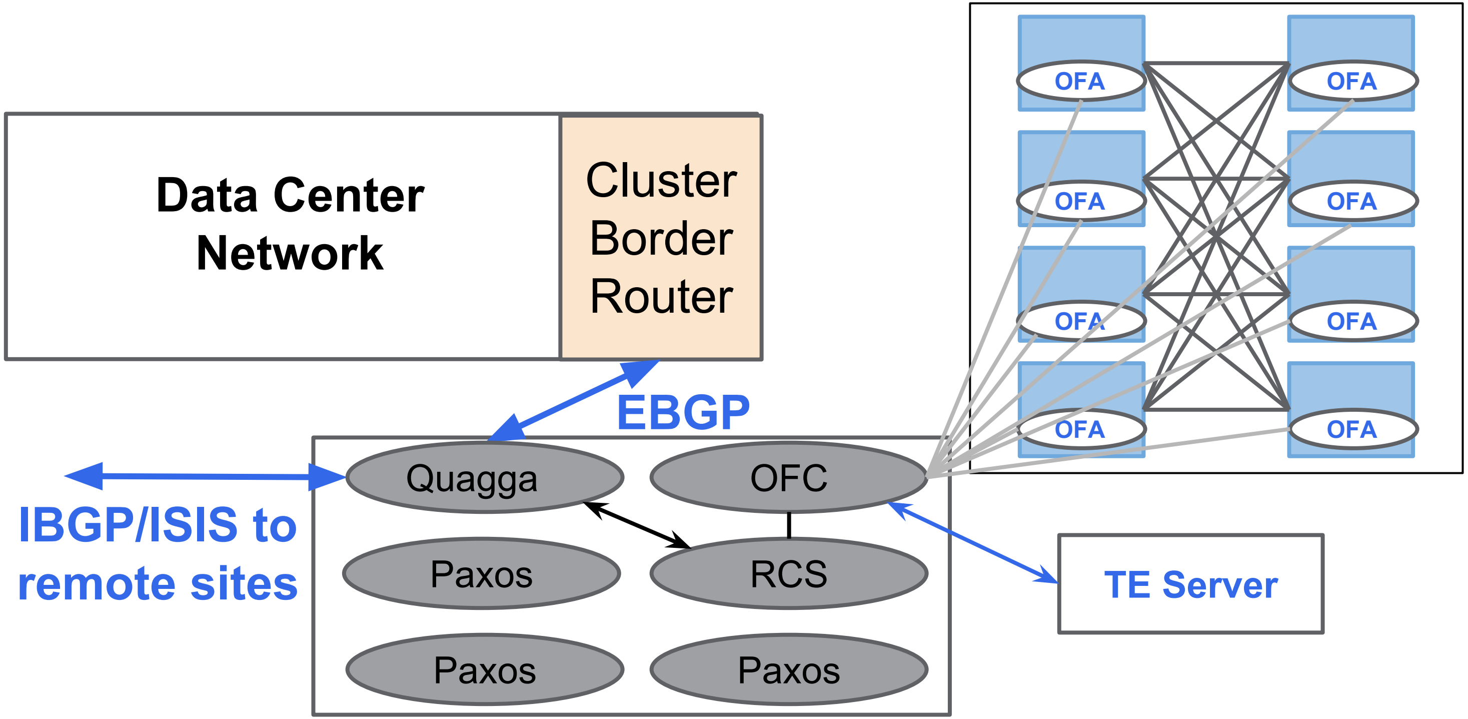
B4: Software Defined inter-Datacenter WAN



History of B4 WAN



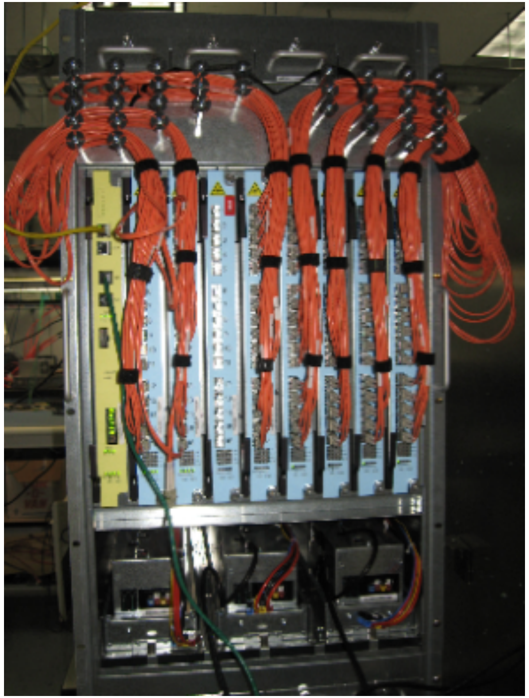
Mixed SDN Deployment



- Ready to introduce new network function virtualization (NFV)

B4: SDN Equipment

- The only way to get well defined control and data plane APIs on a routing HW at that time was to build it ourselves
  - Built from merchant silicon
  - OpenFlow support
  - Does not have all features
  - Multiple chassis per site
  - Fully centralized software controlled



Why SDN?

- SDN ⇏ Cheap Hardware
- SDN = programmatic decomposition of **control**, **data** and **management** planes
- Well defined APIs ⇒ fundamentally easier operational model
- Separation of **control** and **data** planes ⇒ much higher uptime
- Network function virtualization ⇒ new functions rolled out in days (vs years)

Google

Why SDN?

Virtual Network

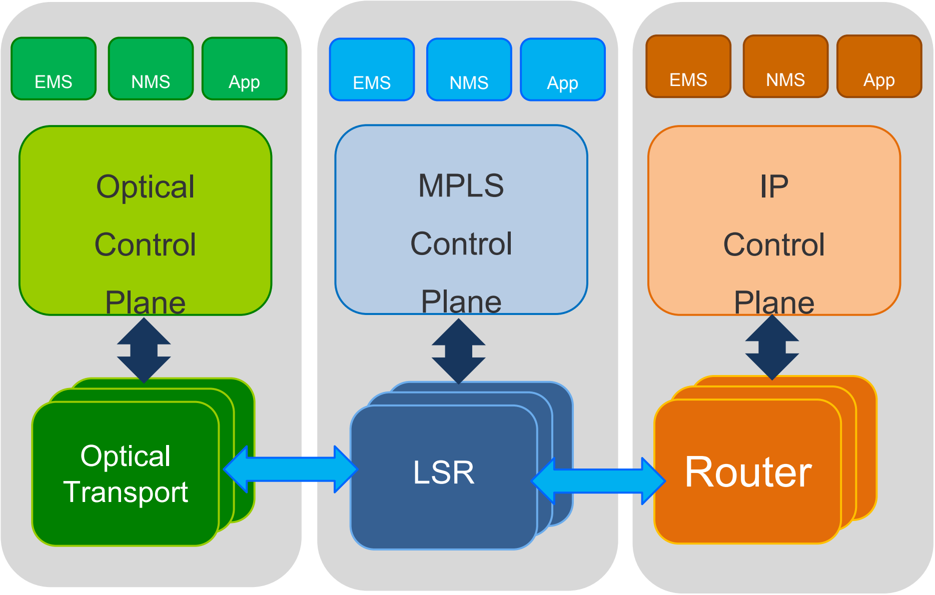
⇔

Physical Network

- SDN ⇏ Cheap Hardware
- SDN = programmatic decomposition of **control**, **data** and **management** planes
- Well defined APIs ⇒ fundamentally easier operational model
- Separation of **control** and **data** planes ⇒ much higher uptime
- Network function virtualization ⇒ new functions rolled out in days (vs years)

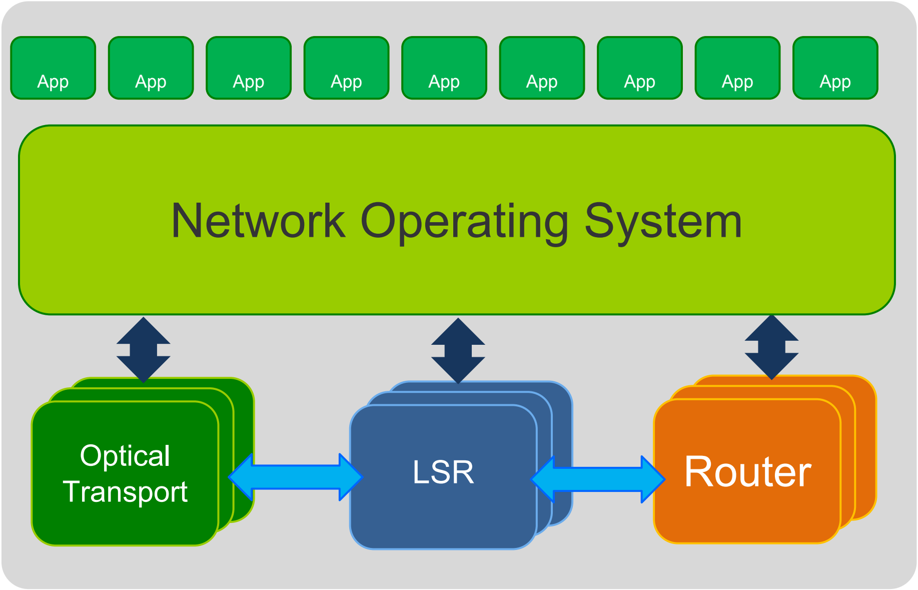
Many Networks → One Network

Layer-cake Network



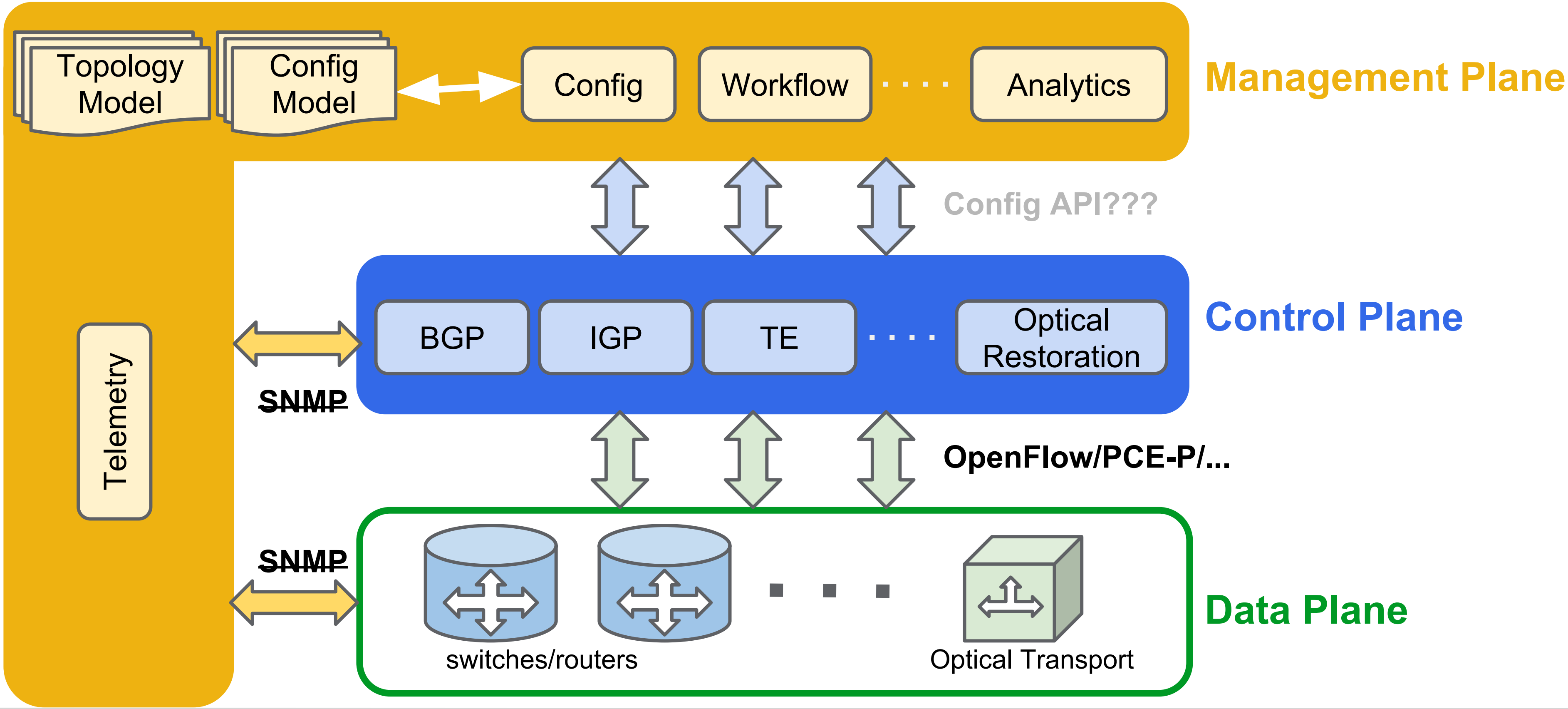
- Heterogeneous control plane
- Heterogeneous network apps
- Large inefficiencies

Software Defined Network



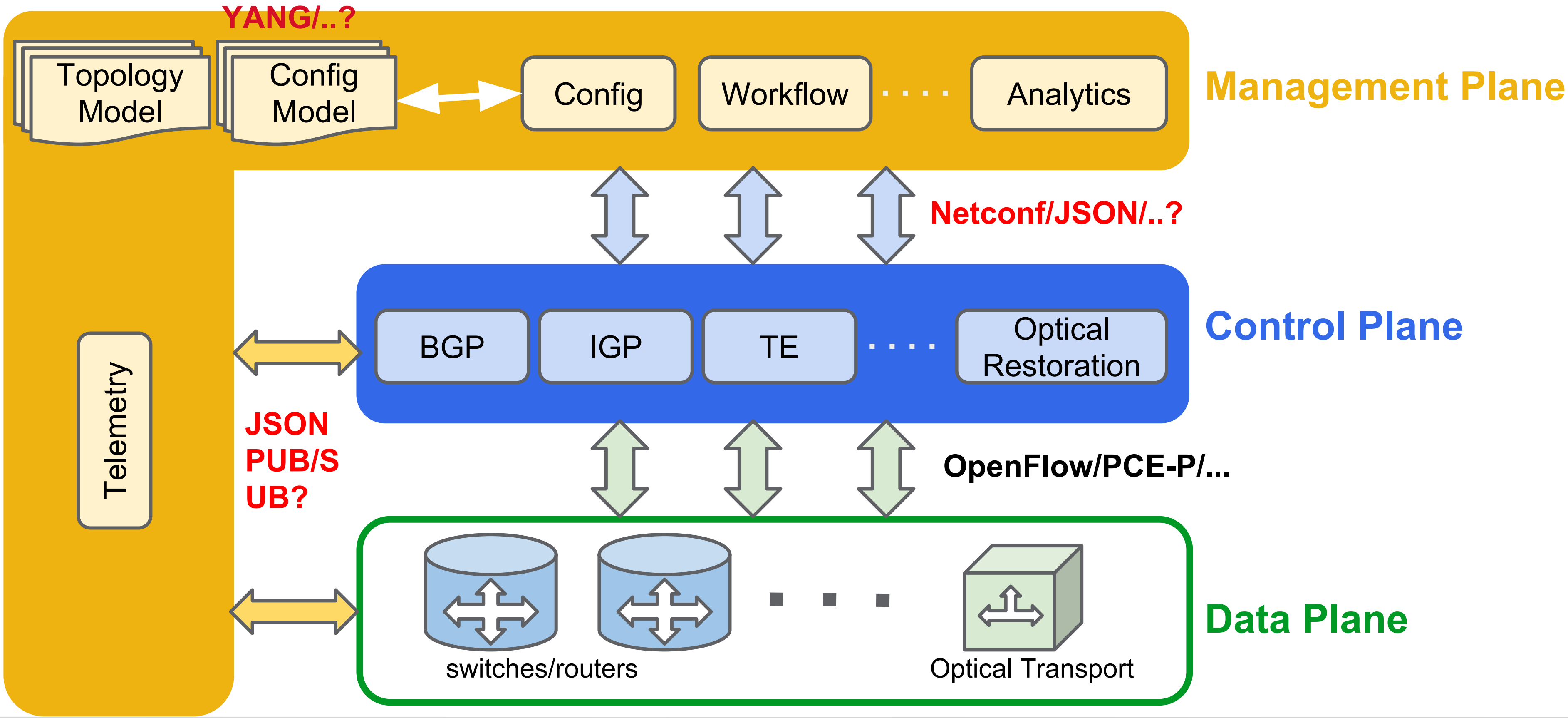
- Common network OS
- Common network apps
- Global view of network states

Anatomy of a Software Defined Network

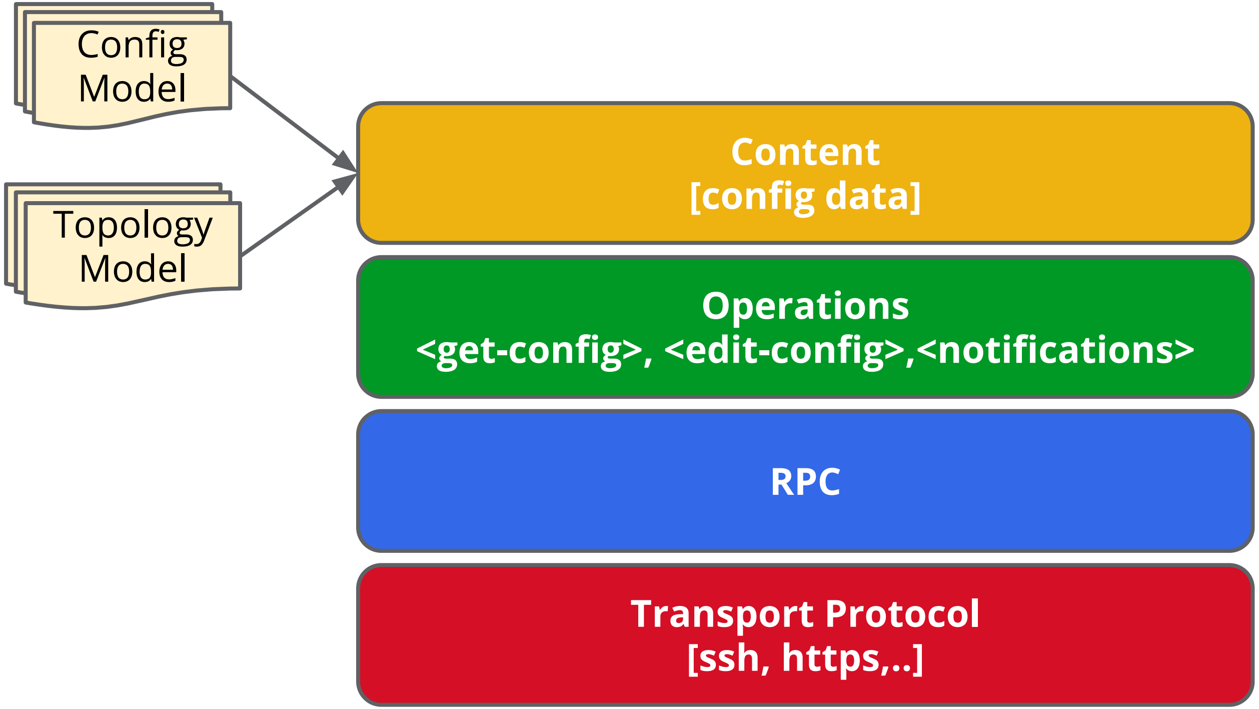


Google

Anatomy of a Software Defined Network



Software Defined Network Configuration


Config Model

Topology Model

Content
[config data]

Operations
<get-config>, <edit-config>,<notifications>

RPC

Transport Protocol
[ssh, https,..]

Towards Declarative Transactional Semantics

- **Good progress** in control plane -> dataplane APIs and protocols (OpenFlow, PCE-P..)
- **Limited progress** in management plane -> control plane protocols and APIs
  - **Netconf** (RFC 6241) is promising, need universal adoption
- **Very limited progress** in standard network data model definition
  - **YANG** as modeling language is promising
  - No vendor-neutral data model yet to describe network/device configuration
  - No standard network topology model
- **No progress** in streaming transfer of bulk-variable/data
  - SNMP is clunky and **not** that simple ☺

Towards a Common Network Model

- **Network Config model** to describe declarative configuration
    - Google is working on a rich vendor-neutral network data model described in YANG
- **Network Topology model** to describe multi-layer network topology (Layer-0 - 7)
    - Google made significant progress in structured hierarchical description of multi-layer connected graphs using protocol buffers* (aka protobuf)
- We welcome collaboration in developing common config and topology models as the basis of true software defined network operation

* http://code.google.com/p/protobuf/

Google

SDN: Beyond the Network Boundaries

- **Goal**
  - Exchange traffic optimally between provider networks (ASNs)

- **Limitations today**
  - Mutual intents of traffic exchange are expressed via BGP as *hints*
  - Suboptimal traffic exchange as the peer networks *guess* optimality

- **SDN advantage**
  - A common **network model** and a **rich pub/sub API**, **leveraging cloud**
  - **Declarative intent** expressed by an ISP:
    - e.g. *deliver 10.20.30.0/24 to Denver, 10.20.31.0/24 to San Francisc*
      do_not_deliver traffic in {Portland, Los Angeles}, avoid_congestion in topology_A, use augmented_topology_B

We welcome collaboration with the ISPs in developing programmatic traffic exchange

Questions?

bkoley@google.com