Integrating Network Management for Cloud Computing Services
What is Cloud Computing

- Deliver applications as services over Internet
- Run applications in datacenters
- Lower capital and operational expenses
Cloud Services are Growing

- Public cloud for consumer service
  - Amazon Web Services
  - Microsoft Azure
  - ...
Cloud Services are Growing

- Public cloud for consumer service
  - Amazon Web Services
  - Microsoft Azure
  - …

- Enterprises out-source IT service
  - Storage: Box, …
  - Analytics: Salesforce, …
  - Productivity: Office365, …
Quality of Cloud Service Depends on Network Quality

Datacenter

- App
- OS

Servers

- Routing
- Hardware

Network Devices

Enterprise Network

ISP
Improve Network Quality

- The old way cannot keep up with the growth of cloud services
  - Deploying more devices with higher bandwidth
Improve Network Quality

• The old way cannot keep up with the growth of cloud services
  • Deploying more devices with higher bandwidth

• The key is proper management of network resources
Examples of Network Management Solutions
Examples of Network Management Solutions

Traffic Engineering
Examples of Network Management Solutions

- Traffic Engineering
- Load Balancing
Examples of Network Management Solutions

- Traffic Engineering
- Load Balancing
- Link Corruption Mitigation
Examples of Network Management Solutions

- Traffic Engineering
- Load Balancing
- Link Corruption Mitigation
- Network Device Configuration
Examples of Network Management Solutions

- Traffic Engineering
- Load Balancing
- Link Corruption Mitigation
- Network Device Configuration
- Device Power Control
Examples of Network Management Solutions

- Traffic Engineering
- Load Balancing
- Link Corruption Mitigation
- Network Device Configuration
- Device Power Control
Problems of Current Network Management

• Disjoint management of network components

• Low-level interfaces with network devices
Disjoint Management Systems

• Division of labor in pre-cloud era
Disjoint Management Systems

- Division of labor in pre-cloud era
Disjoint Management Systems

• Division of labor in pre-cloud era
Disjoint Management Systems

- Division of labor in pre-cloud era

Servers
  - App
  - OS

Traffic Engineering
  - Routing
  - Hardware

ISP

Network Devices

Users

Server Provisioning
Datacenter Breaks Balance

- Coordination of server & network:
  - More applications are built as multi-tier distributed systems
  - Intra-datacenter traffic is new majority
Datacenter Breaks Balance

- Coordination of server & network:
  - More applications are built as multi-tier distributed systems
  - Intra-datacenter traffic is new majority

- Infrastructure evolution speeds up:
  - Network architecture changes (e.g., FatTree)
Disjoint Mgmt. is Bottleneck

Because:

• Cloud service providers have stakes in all three management areas:
  • Server, infrastructure, traffic
Disjoint Mgmt. is Bottleneck

Because:

- Cloud service providers have stakes in all three management areas:
  - **Server, infrastructure, traffic**

- Great opportunity exists for consolidation
  - *Google and Microsoft on SoftWAN*
Yet Another Problem: Low-level Device Interaction

- Hardware vendors differentiate with specialized devices
- Network operation:
  - intensively uses vendor-specific APIs
  - heavily depends on experiences
Cloud Service is Different

- Much more devices in datacenters than traditional networks
  - Automation is a must
Cloud Service is Different

• Much more devices in datacenters than traditional networks
  • Automation is a must

• Commodity devices instead of specialized hardware
  • Homogeneity is preferred
Cloud Service is Different

• Much more devices in datacenters than traditional networks
  • Automation is a must

• Commodity devices instead of specialized hardware
  • Homogeneity is preferred

• Lower vendor dependence pays off (MS & Amazon on SoftLB)
Promising Yet Limited SDN

• Great way of automating traffic management with high-level programming paradigms

• Yet literature focus on just ‘traffic’
Summary of Problems

• This dissertation solves:

• *Disjoint management of server, infrastructure, and traffic*

• *Low-level device interaction for broader scope of network management*
Practical Approach
Practical Approach

Identify opportunity of integration
Practical Approach

Identify opportunity of integration

Simple and expressive programming abstraction
Practical Approach

Identify opportunity of integration

Simple and expressive programming abstraction

Efficient and scalable system for execution
Practical Approach

1. Identify opportunity of integration
2. Simple and expressive programming abstraction
3. Deployment in real world
4. Efficient and scalable system for execution
Practical Approach

- Identify opportunity of integration
- Simple and expressive programming abstraction
- Deployment in real world
- Efficient and scalable system for execution
# Contributions

<table>
<thead>
<tr>
<th>Project</th>
<th>Publication</th>
<th>Deployment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Statesman</td>
<td>SIGCOMM’14</td>
<td>Microsoft Azure</td>
</tr>
<tr>
<td>HONE</td>
<td>JNSM, Vol. 23, 2015</td>
<td>Overture/Verizon</td>
</tr>
<tr>
<td>Sprite</td>
<td>SOSR’15</td>
<td>In process with OIT</td>
</tr>
</tbody>
</table>
Statesman: Safe Datacenter Traffic/Infrastructure Management

Datacenter

- App
- OS

Servers

Network Devices

- Routing
- Hardware

Enterprises

ISP

Other ISPs
Hone: End-host/Network Cooperative Traffic Management

Datacenter

Servers

Network Devices

App

OS

Routing

Hardware

Other ISPs

ISP

ISP

Enterprises
Sprite: Direct Control of Entrant ISP for Enterprise Traffic

Datacenter

- App
- OS
- Routing
- Hardware

Servers

Network Devices

Enterprises

ISP

Other ISPs
What Follows

- Brief explanation of each project
- Open issues
- Related work
- Q&A
Statesman:
Integrating Network Infrastructure Management
Problem for Cloud Providers

- Multiple mgmt. solutions coexist
  - for traffic and infrastructure mgmt.
Problem for Cloud Providers

• Multiple mgmt. solutions coexist
  • *for traffic and infrastructure mgmt.*
• Complexity is the main problem
Problem for Cloud Providers

- Multiple mgmt. solutions coexist
  - for traffic and infrastructure mgmt.
- Complexity is the main problem
  - Development
    - Scale & heterogeneity of devices
Problem for Cloud Providers

- Multiple mgmt. solutions coexist
  - for traffic and infrastructure mgmt.
- Complexity is the main problem
  - Development
    - Scale & heterogeneity of devices
  - Coordination
    - Conflicts and safety violations
Conflict
Conflict

Link-corruption-mitigation adjusts traffic away from Core1
Conflict

Link-corruption-mitigation adjusts traffic away from Core1

TE tunes traffic among links to Core1, 2
Safety Violation

Core 1 2
Agg A  Agg B
ToRs
Safety Violation

Link-corruption-mitigation shuts down faulty Agg A
Safety Violation

Core 1

Agg A

2

ToRs

Agg B

Link-corruption-mitigation shuts down faulty Agg A

Firmware-upgrade schedules Agg B to upgrade
The Statesman System

- Network operating system
- Common layer to consolidate traffic and infrastructure management
- Resolve conflicts & safety violations
The Statesman System

- Network operating system
  - Common layer to consolidate traffic and infrastructure management
  - Resolve conflicts & safety violations

- Core techniques:
  - Network state & three-view workflow
  - State dependency model
  - Scalable & robust system
Three-view Workflow with Network State

- Observed State
- Proposed State
- Target State
Three-view Workflow with Network State
Three-view Workflow with Network State

What we see from the network
Three-view Workflow with Network State

Observed State

Proposed State

Target State

What we see from the network

What we want the network to be

Solutions
Three-view Workflow with Network State

- **Observed State**: What we see from the network
- **Proposed State**: What we want the network to be
- **Target State**: What can be actually done on the network

---

**Solutions**

**Statesman**
State Dependency for Conflict Detection

Device

Link

Device
State Dependency for Conflict Detection

PowerState

Device

Link
State Dependency for Conflict Detection

- FirmwareVersion
- PowerState

Device

Link
State Dependency for Conflict Detection

Device

ConfigurationState
FirmwareVersion
PowerState

Link
State Dependency for Conflict Detection

- ConfigurationState
- FirmwareVersion
- PowerState

Device

ConfigurationState

AdminState

Link
State Dependency for Conflict Detection

- RoutingState
- ConfigurationState
- FirmwareVersion
- PowerState
- ConfigurationState
- AdminState

Device

Link
State Dependency for Conflict Detection

- PathState
- RoutingState
- ConfigurationState
- FirmwareVersion
- PowerState

Device

Link

ConfigurationState
AdminState
Statesman System

- Observed State
- Proposed State
- Target State
Statesman System

Storage Service

- Observed State
- Proposed State
- Target State
Statesman System

Storage Service

Observed State  Proposed State  Target State

Monitor
Statesman System

Storage Service

Observed State

Proposed State

Target State

Checker

Monitor
Statesman System

Storage Service

Observed State → Proposed State → Target State

Monitor → Checker → Updater
Deployment Overview

• Operational in Microsoft Azure since October 2013
• Cover 10 DCs of 20K devices
• 3 production management solutions built and running
Case: Resolve Conflict
Inter-DC TE & Firmware-upgrade

DC = Data Center
BR = Border Router
Empty (0%)  Low (1~40%)  Medium (40%~80%)  High (80%~100%)

Time Series in Minutes

Link Index
Firmware-upgrade acquires lock of BR1
TE fails to acquire lock, and moves traffic away
TE fails to acquire lock, and moves traffic away.
BR1 firmware upgrade starts
BR1 firmware upgrade starts.

BR1 firmware upgrade ends. Lock released.
BR1 firmware upgrade starts

TE re-acquires lock, and moves traffic back
BR1 firmware upgrade starts

TE re-acquires lock, and moves traffic back
Statesman Summary

- Programming abstraction
  - Three-view network state model

- Efficient and scalable system
  - Automatic and safe infrastructure management system

- Deployment
  - Operational in Microsoft Azure worldwide

- SIGCOMM 2014
Hone:
Combining End-host and Network for Traffic Management
Problem

- Traffic management is limited by the scope of network devices
  - Coarse granularity & limited view
The Hone System

- Bring end hosts into traffic mgmt.
The Hone System

• Bring end hosts into traffic mgmt.
• Core techniques:
  • Access of socket & transport layers
The Hone System

• Bring end hosts into traffic mgmt.

• Core techniques:
  • Access of socket & transport layers
  • Expressive three-stage programming framework
The Hone System

• Bring end hosts into traffic mgmt.

• Core techniques:
  • Access of socket & transport layers
  • Expressive three-stage programming framework
  • Efficient system with parallel execution of partitioned program
Programming Model

- Framework for each stage
- Programmable body of each stage
  - Focus on measurement and analysis
Hone System

Host

- App
- Hone Agent
- Server OS

Network
Hone System

Controller

Mgmt. Program

Hone Runtime System

Host

App

Hone Agent

Server OS

Network
Hone System
Evaluation

• Built multiple traffic management solutions on Hone

• Show ‘distributed rate limiting’ here
  • Limit total bandwidth across all instances of a tenant in a public cloud
DRL on HONE

• Host-side execution:
  • Measure
  • Calculate throughput
  • Aggregate among connections

• Controller-side execution:
  • Aggregate among hosts
  • Generate new rate-limit policies
Distributed Rate Limiting
Distributed Rate Limiting

![Graph showing throughput in MBps for Host1, Host2, and their sum. The graph indicates a drop in throughput at the time marked as 20.]

Throughput in MBps

- Host1
- Host2
- Sum

Time (in seconds)
Distributed Rate Limiting

Throughput in MBps

Time Sequ
Distributed Rate Limiting

![Graph showing distributed rate limiting with throughput in MBps over time.](image)
Distributed Rate Limiting

![Graph showing distributed rate limiting](image)

- **Y-axis**: Throughput in MBps
- **X-axis**: Time Sequence in Second
Hone Summary

• Programming abstraction
  • Access to fine-grained data in servers
  • Three-stage framework

• Efficient and scalable runtime

• Deployment
  • Integrated into product of Overture for Verizon Business Cloud

• Springer JNSM Volume 23, 2015
Sprite:
Bridging Enterprise and ISP for Inbound Traffic Control
Problem for Enterprises: Inbound Traffic Engineering

Service 1

Internet

ISP 1

ISP 2

ISP 3

Service 2

Enterprise Network
Problem for Enterprises: Inbound Traffic Engineering

- Netflix
- Internet
- ISP 1
- ISP 2
- ISP 3
- Service 2
- Enterprise Network
Problem for Enterprises: Inbound Traffic Engineering
Mechanism of Sprite

Enterprise Network

User

ISP 1

ISP 2

Other ISPs

YouTube

VoIP
Mechanism of Sprite

Enterprise Network

User

ISP 1
1.1.1.0/24

ISP 2
1.1.2.0/24

Other ISPs

YouTube

VoIP
Mechanism of Sprite

Enterprise Network

User

ISP 1

1.1.1.0/24

ISP 2

1.1.2.0/24

Other ISPs

1.1.1.0/24

1.1.2.0/24

YouTube

VoIP
Mechanism of Sprite

Enterprise Network

10.1.1.2
Src: 1.1.1.2
User

ISP 1
1.1.1.0/24

ISP 2
1.1.2.0/24

Other ISPs

YouTube

VoIP
Mechanism of Sprite

Enterprise Network

User

Src: 1.1.1.2

10.1.1.2

ISP 1

1.1.1.0/24

ISP 2

1.1.2.0/24

Other ISPs

src: 1.1.2.2

VoIP

YouTube
Challenges

• Naïve solution
  • All done by the border router
Challenges

• Naïve solution
  • All done by the border router

• Need a distributed solution
Challenges

- Naïve solution
  - All done by the border router
- Need a distributed solution
  - Control-plane scaling
Challenges

• Naïve solution
  • All done by the border router

• Need a distributed solution
  • Control-plane scaling
  • Data-plane scaling
Challenges

• Naïve solution
  • All done by the border router

• Need a distributed solution
  • Control-plane scaling
  • Data-plane scaling

• Need a simple management interface
Three Tiers of Abstractions

Abstraction

Example
Three Tiers of Abstractions

Abstraction

Example

High-level Policy
Three Tiers of Abstractions

**Abstraction**

High-level Policy

**Example**

<BioDept, Analytic> ➞ BestLatency
Three Tiers of Abstractions

Abstraction

High-level Policy

Network-level Rule

Example

<BioDept, Analytic> ➔ BestLatency
Three Tiers of Abstractions

**Abstraction**

High-level Policy

Network-level Rule

**Example**

\(<\text{BioDept, Analytic}> \rightarrow \text{BestLatency}\)

\(<10.1.0.0/16, 184.168.1.0/24> \rightarrow \text{ISP1}\)
Three Tiers of Abstractions

Abstraction

High-level Policy

Network-level Rule

Example

<BioDept, Analytic> ➔ BestLatency

<10.1.0.0/16, 184.168.1.0/24> ➔ ISP1
Three Tiers of Abstractions

**Abstraction**

- High-level Policy
- Network-level Rule
- Per-flow Rule

**Example**

- `<BioDept, Analytic> ➔ BestLatency`
- `<10.1.0.0/16, 184.168.1.0/24> ➔ ISP1>`
Three Tiers of Abstractions

Abstraction

High-level Policy

Network-level Rule

Per-flow Rule

Example

<BioDept, Analytic> ➔ BestLatency

<10.1.0.0/16, 184.168.1.0/24> ➔ ISP1

<10.1.1.42, 184.168.1.17> SNAT to 1.1.1.2
Case: Dynamic Perf-driven Balancing

• Move users’ connections among ISPs for best per-user performance

• Live Internet experiment on AWS VPC and PEERING
  • 10 users watch movies on YouTube
User YouTube Throughput

![Graph showing average per-user throughput in Kbps over time, comparing data via ISP Gatech and ISP Clemson.]
Sprite Summary

- Direct and fine-grained inbound traffic control
- Scalable solution
  - Scaling control and data planes
- Evaluation
  - In collaboration with OIT
- SOSR’15
Open Issue #1
Statesman + Hone

- Merge Hone into Statesman
  - Joint server, traffic, infrastructure mgmt.

- Possible exploration
  - State abstraction for server data
  - Dependency of server and network states
  - Safety invariants involving servers
Open Issue #2

Transactional Statesman

• Transactional semantics for conflict resolution

• Possible exploration
  • Grouping semantics
  • Condition semantics
Open Issue #3
Hone for Multi-tenant Cloud

• Loosen the assumption of having access to the hosts’ OS

• Possible exploration
  • *Infer stats in guest OS from data in hypervisor*
Related work
# Statesman Related Work

<table>
<thead>
<tr>
<th>Work</th>
<th>What they do</th>
<th>Statesman</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDN works</td>
<td>Centralized control of flows</td>
<td>Wider spectrum of management applications</td>
</tr>
<tr>
<td>Onix</td>
<td>Single repository of network stats</td>
<td>Three-view network state model</td>
</tr>
<tr>
<td>Pyretic</td>
<td>Target at flow management</td>
<td>Wider spectrum of applications</td>
</tr>
<tr>
<td>Corybantic</td>
<td>Tight cross-application proposal evaluation</td>
<td>Loose coupling</td>
</tr>
<tr>
<td>FlowVisor</td>
<td>Virtual topology slicing</td>
<td>Network state model</td>
</tr>
</tbody>
</table>

53
## Hone Related Work

<table>
<thead>
<tr>
<th>Work</th>
<th>What they do</th>
<th>Hone</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network Exception Handler</td>
<td>Use hosts only as software switches</td>
<td>Go deeper into socket layer</td>
</tr>
<tr>
<td>Gigascope</td>
<td>Extend SQL</td>
<td>Use functional language to construct</td>
</tr>
<tr>
<td>Chimera</td>
<td>Specific application (intrusion detection)</td>
<td>Support various management solutions</td>
</tr>
<tr>
<td>MapReduce</td>
<td>Naturally parallelizable data</td>
<td>Data inherently associated with collection point</td>
</tr>
</tbody>
</table>
## Sprite Related Work

<table>
<thead>
<tr>
<th>Work</th>
<th>What they do</th>
<th>Sprite</th>
</tr>
</thead>
<tbody>
<tr>
<td>BGP AS-path prepending</td>
<td>Tune BGP configurations to influence neighboring ASes</td>
<td>Direct control from inside enterprise networks</td>
</tr>
<tr>
<td>Works on Internet re-architecture</td>
<td>Clean-slate design in routing or hosts</td>
<td>Incrementally deployable</td>
</tr>
<tr>
<td>Tunnel-based works</td>
<td>Two ends cooperate to control traffic</td>
<td>Only need the client side to act</td>
</tr>
</tbody>
</table>
Thanks!

Q&A