Eliminating Adverse Control Plane Interactions in Independent Network Systems

Matthew K. Mukerjee

Computer Science PhD Thesis Defense

May 1st, 2018
Network Control

Routing

CDN server selection

VM migration

Congestion Control
Network Control

Coordination

Routing

CDN server selection

Coordination

VM migration

Congestion Control
Control Plane
Routing: “figure out” best path (periodically computed)

Data Plane
Forwarding: data transmission (done per-packet)

Source -> Which way? -> Destination
Distributed OSPF

Centralized SDN

Controller
**OSPF**

- Quick failure response
- Bad at performance optimization

**SDN**

- Good at performance optimization
- Slow failure response
Distributed OSPF

Quick failure response

Bad at performance optimization

Centralized SDN

Good at performance optimization

Slow failure response
Bad CDN server selection → ISP paying for costly routes
User decisions

TCP decisions

Application decisions

Issues?

Issues?

Issues?
Categorizing Control Coordination

App TE + ISP TE

Expensive

Cheap

BGP + BGP

Expensive

Cheap

Coflow (App + DC scheduling)

BGP

OSPF ISP A

OSPF ISP B

OSPF ISP C

Internet-scale Routing (BGP + OSPF)
Categorizing Control Coordination

- **Reaction**
  - App TE + ISP TE

- **Transparency**
  - Coflow (App + DC scheduling)

- **Priority Ranking**
  - BGP + BGP

- **Hierarchical Partitioning**
  - Internet-scale Routing (BGP + OSPF)
Control Coordination

Scenario: Layering
Etalon

Scenario: Admin
VDX

Scenario: Scalability
VDN

App TE + ISP TE
Reaction

BGP + BGP
Priority Ranking

Coflow
Transparency

Internet-scale Routing
Hierarchical Partitioning
Control Coordination

- Scenario: Layering
  - Etalon
- Scenario: Admin
  - VDX
- Scenario: Scalability
  - VDN

- App TE + ISP TE
  - Reaction
- BGP + BGP
  - Priority Ranking

- Coflow
  - Transparency
- Internet-scale Routing
  - Hierarchical Partitioning
Difficult to scale datacenters with demand

Higher Bandwidth + Higher Port Count ≠ CMOS limits...

Use circuits to build bigger + faster networks!

Reconfigurable Datacenter Networks (RDCNs)
**Circuit Switch Design**

How do you physical build it?

**Network Scheduling**

How do you make use of it?

**End-to-End Challenges**

What existing things break?
RDCN switch design

Challenge:

Workloads

ToR Switch

Server 1
Server 2
... Server M
Rack 1

App Demand

ToR Switch

Server 1
Server 2
... Server M
Rack 2

ToR Switch

Server 1
Server 2
... Server M
Rack N

Packet Network

Packet Switch

Circuit Switch
RDCN switch design

Challenge: Workloads

App Demand

Packet Switch

Packet Network

Circuit Switch

ToR Switch

Server 1

Server 2

Server M

Rack 1

Rack 2

Rack N
Rack 1

Packet Switch

Rack 2

Circuit Switch

Rack N

1 —> 2

1 —> 3

1 —> N

2 —> 3

2 —> 6

2 —> N

2 —> 5

N —> 4

N —> 1

N —> 2

N —> 7

Rack 1

Rack 2

Rack N

RDCN scheduling
RDCN scheduling

Packet Switch

Circuit Switch

1 —> 2
1 —> 3
1 —> N
1 —> 2
2 —> 3
2 —> 6
2 —> N
2 —> 5
N —> 4
N —> 1
N —> 2
N —> 7

RDCN Scheduling Algorithm (e.g., Solsticce)
RDCN scheduling

- Packet Switch
- Circuit Switch

Rack 1
 Packet Switch
 Rack 2
 Circuit Switch
 Rack N

1 —> 2
1 —> 3
1 —> N
1 —> 2
2 —> 3
2 —> 6
2 —> N
N —> 4
N —> 2
N —> 1
N —> 7
2 —> 5

RDCN Scheduling Algorithm (e.g., Solstice)

For Circuit Switch

For Packet Switch
RDCN scheduling

For Circuit Switch

Rack 1

Packet Switch

Circuit Switch

Rack 2

Rack N

For Packet Switch

Rack N

Rack 1

2 —> 3

1 —> 2

1 —> N

N —> 4

2 —> 6

N —> 2

N —> 1

N —> 7

2 —> 5

RDCN Scheduling Algorithm (e.g., Solstice)
RDCN scheduling

Circuit Switch Schedule:

<table>
<thead>
<tr>
<th>Rack 1</th>
<th>Rack 2</th>
<th>...</th>
<th>Rack N</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 ➔ 2</td>
<td>2 ➔ 3</td>
<td>...</td>
<td>N ➔ 4</td>
</tr>
<tr>
<td>300μs</td>
<td>20μs</td>
<td></td>
<td>180μs</td>
</tr>
</tbody>
</table>

Packet Switch Schedule:

Rack 1 Packet Switch

<table>
<thead>
<tr>
<th>Rack 1</th>
<th>Rack 2</th>
<th>...</th>
<th>Rack N</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 ➔ 3</td>
<td>2 ➔ N</td>
<td>...</td>
<td>N ➔ 1</td>
</tr>
<tr>
<td>180μs</td>
<td>20μs</td>
<td></td>
<td>518μs</td>
</tr>
</tbody>
</table>

RDCN Scheduling Algorithm (e.g., Solstice)

For Circuit Switch:

1 ➔ 2
1 ➔ N
1 ➔ 3
2 ➔ 6
2 ➔ N
N ➔ 4
N ➔ 2
N ➔ 1
N ➔ 7
2 ➔ 5

For Packet Switch:

N ➔ 7
2 ➔ 5
Contributions

End-to-End Challenges

Challenge:
- BW Fluct.
  Solution: Dynamic Buffer Resizing

Challenge:
- Demand Estimation
  Solution: Endhost-based Estimation

Challenge:
- Workloads
  Solution: App-specific Modification

Etalon, an RDCN Emulator
Overview

End-to-End Challenges

Challenge:
Demand Estimation
Solution: Endhost-based Estimation

Challenge:
Workloads
Solution: App-specific Modification

Challenge:
BW Fluct.
Solution: Dynamic Buffer Resizing

Etalon, an RDCN Emulator
TCP and rapid bw fluctuations

Challenge: BW Fluct.
Solution: Dynamic Buffer Resizing

Sender → ToR Queue → Circuit Switch → High BW → ToR Queue → Receiver

Packet Switch

Low BW

Time (μs)
TCP and rapid bw fluctuations

Challenge: BW Fluct.

Solution: Dynamic Buffer Resizing

Sender → ToR Queue → Packet Switch (Low BW) → ToR Queue → Circuit Switch (High BW) → Receiver
TCP and rapid bw fluctuations

Challenge: BW Fluct.
Solution: Dynamic Buffer Resizing

Packet Switch

Low BW

High BW

Sender

ToR Queue

Circuit Switch

ToR Queue

Receiver
TCP and rapid bw fluctuations

Challenge: BW Fluct.
Solution: Dynamic Buffer Resizing

Packet Switch

High BW

Low BW

Sender

ToR Queue

ToR Queue

Receiver

Time (μs)

BW
TCP and rapid bw fluctuations

Challenge: BW Fluct.
Solution: Dynamic Buffer Resizing

Packet Switch

High BW

Circuit Switch

Sender → ToR Queue → Packet Switch → ToR Queue → Receiver

Time (μs)

BW
TCP and rapid bw fluctuations

Challenge: BW Fluct.
Solution: Dynamic Buffer Resizing

Sender -> ToR Queue -> Packet Switch -> Circuit Switch -> ToR Queue -> Receiver

High BW

Low BW

Time (μs)

BW
TCP and rapid bw fluctuations

Challenge: BW Fluct.

Solution: Dynamic Buffer Resizing

Packet Switch

Sender

ToR Queue

Circuit Switch

High BW

ToR Queue

Receiver

Low BW

Graph showing time (μs) vs. bandwidth (BW) with fluctuations indicated.
TCP and rapid bw fluctuations

Challenge: BW Fluct.
Solution: Dynamic Buffer Resizing

What we want

What we get
TCP and rapid bw fluctuations

Challenge: BW Fluct.

Solution: Dynamic Buffer Resizing

Sender

ToR Queue

Packet Switch

Latency

SMALL

ToR Queue

Receiver

Circuit Switch

High BW

Low BW

BW

Time (μs)
TCP and rapid bw fluctuations

Challenge: BW Fluct.
Solution: Dynamic Buffer Resizing

Sender → ToR Queue → Packet Switch → ToR Queue → Receiver

Latency

BIG

Circuit Switch

High BW

Low BW

Time (μs)

BW
Challenge: BW Fluct.

Solution: Dynamic Buffer Resizing

TCP and rapid bw fluctuations

Sender

ToR Queue

Packet Switch

Low BW

High BW

Circuit Switch

Time (μs)

BW

How Early?

Sender

ToR Queue

Circuit Switch

ToR Queue

Receiver

Bandwidth

BIG

How Early?
Static buffers provide good circuit util or latency

Challenge: BW Fluct.

Solution: Dynamic Buffer Resizing

Table:

<table>
<thead>
<tr>
<th>Buffer size (packets)</th>
<th>AVG. Circuit Utilization</th>
</tr>
</thead>
<tbody>
<tr>
<td>SMALL 4 packets</td>
<td>16%</td>
</tr>
<tr>
<td>SMALL 8 packets</td>
<td>25%</td>
</tr>
<tr>
<td>SMALL 16 packets</td>
<td>39%</td>
</tr>
<tr>
<td>SMALL 32 packets</td>
<td>61%</td>
</tr>
<tr>
<td>SMALL 64 packets</td>
<td>92%</td>
</tr>
<tr>
<td>SMALL 128 packets</td>
<td>99%</td>
</tr>
</tbody>
</table>

Low utilization
Static buffers provide good circuit util or latency.

Challenge: BW Fluct.

Solution: Dynamic Buffer Resizing

<table>
<thead>
<tr>
<th>Buffer size (packets)</th>
<th>Avg. circuit utilization</th>
<th>Buffer size (packets)</th>
<th>Median latency (μs)</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>16</td>
<td>48</td>
<td>SMALL</td>
</tr>
<tr>
<td>8</td>
<td>25</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>39</td>
<td>32</td>
<td></td>
</tr>
<tr>
<td>32</td>
<td>61</td>
<td>64</td>
<td></td>
</tr>
<tr>
<td>64</td>
<td>92</td>
<td>128</td>
<td></td>
</tr>
<tr>
<td>128</td>
<td>99</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

High Latency

Buffer size (packets)

Median latency (μs)
Challenge:
BW Fluct.

Solution:
Dynamic Buffer Resizing

Buffer resize provides good circuit util **and** latency

![Graph showing average circuit utilization vs. early buffer resize time (μs)]

- Avg. circuit utilization
- Early buffer resize (μs)
  - 0 μs: 39
  - 200 μs: 52
  - 400 μs: 56
  - 600 μs: 65
  - 800 μs: 79
  - 1000 μs: 88
  - 1200 μs: 100
  - 1400 μs: 100
Buffer resize provides good circuit util and latency.

Challenges:
- BW Fluct.

Solutions:
- Dynamic Buffer Resizing

2x increase in utilization

Buffer resize provides good circuit util and latency.
Overview

End-to-End Challenges

Challenge: Demand Estimation
Solution: Endhost-based Estimation

Challenge: BW Fluct.
Solution: Dynamic Buffer Resizing

Challenge: Workloads
Solution: App-specific Modification

Etalon, an RDCN Emulator
Difficult to schedule workloads

Challenge: Workloads
Solution: App-specific Modification
Difficult to schedule workloads

Solution: App-specific Modification

Challenge: Workloads
Schedule:

Config 1:
- A → B
- B → C
- C → D
- D → A
- RECONFIG DELAY

Config 2:
- A → C
- B → D
- C → A
- D → B
- RECONFIG DELAY

Config 3:
- A → D
- B → A
- C → B
- D → C
- RECONFIG DELAY
Challenge: Workloads

Solution: App-specific Modification

reHDFS

Schedule:

A -> C
B -> D
C -> A
D -> B

Config 1
reHDFS reduces tail latency

9x decrease in write time

Challenge: Workloads

Solution: App-specific Modification

CDF (%) vs. HDFS write completion time (ms)

HDFS vs. reHDFS
Control Coordination

Scenario: Layering
Etalon

Scenario: Admin
VDX

Scenario: Scalability
VDN

App TE + ISP TE
Reaction
BGP + BGP
Priority Ranking

Coflow
Transparency
Internet-scale Routing
Hierarchical Partitioning
Control Coordination

Scenario: Admin
VDX

Scenario: Scalability
VDN

App TE + ISP TE
Reaction
BGP + BGP
Priority Ranking

Internet-scale Routing
Hierarchical Partitioning

Coflow
Etalon

Scenario: Transparency
Layering
Traditional Content Delivery

Legend:

- Content Provider (CP)
- CDN
- Client
Changing Content Delivery

Legend:

- Content Provider (CP)
- Content
Brokered Content Delivery

Legend:
- Content
- Control

Content Provider (CP)

CDN

CDN

Broker

Client

Client

Client
Brokered Content Delivery

Easier for CPs to meet performance and cost goals
Brokered Content Delivery

Content Provider (CP)

Brokers select “best” CDN for clients to minimize cost and meet performance goals

Legend:
- → Content
- → Control

Broker

Client B

Client B

Client B
Brokered Content Delivery

How do brokers and CDNs impact each other? (this talk)
Contributions

• Identify challenges that brokers and CDNs create for each other by analyzing data from both

• Examine the design space of CDN-broker interfaces

• Evaluate the efficacy of different designs
CDN Cost and Pricing

Legend:
- Content

Internal Costs: **Bandwidth** (mostly)

Content Provider (CP) → CDN → Client
Do bandwidth costs differ across geographic regions?

Legend:
- **Content**

Internal Costs:
- **Bandwidth** (mostly)
CDN Cost / Byte Delivered

30x

difference in cost per byte between the most expensive and least expensive countries
CDN Internal Cost
CDN Internal Cost

CDN X
$ $$

CDN Y
$ $

CDN X
$ $
CDN External Price

CDN Pricing

Content Provider (CP)

 CDN Y

 CDN X

 CDN X

 $$$

 $$

 Client

 Client

 Client

 Client

 Client

 Client

 CDN Y

 CDN X

 Client

 Client

 Client

 Client

 CDN X

 $
CDN External Price

**CDN Pricing**

Content Provider (CP)

CDN X

CDN Y

CDN Y makes money, CDN X loses money
Do we see traffic patterns like this at the country level?
Country Level Traffic

% Used in Country

Country (Anonymized)
<table>
<thead>
<tr>
<th>Country (Anonymized)</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
<th>13</th>
<th>14</th>
<th>15</th>
</tr>
</thead>
<tbody>
<tr>
<td>% Used in Country</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>75</td>
<td>50</td>
<td>25</td>
<td>25</td>
<td>25</td>
<td>25</td>
</tr>
</tbody>
</table>

Flat pricing makes CDN profits unpredictable with brokers.

Country 8 costly —> CDN B loses money!
Country 7 cheap —> CDN A profits!
Contributions

- Identify challenges that brokers and CDNs create for each other by analyzing data from both.
- Examine the design space of CDN-broker interfaces.
- Evaluate the efficacy of different designs.
Brokered Delivery Today

Legend:
- **Content**
- **Control**
Brokered Delivery Today

- Content Provider (CP)
- CDN
- CDN
- Broker
- Client
- Client
- Client
Brokered Delivery Today

Content Provider (CP)

CDN

CDN

Broker

Client

Client

Client
Brokered Delivery Today

Content Provider (CP)

CDN

CDN

Broker

Client

Client

Client
Brokered Delivery Today

Content Provider (CP)

CDN

Broker

Client
Brokered Delivery Today

Content Provider (CP)

CDN

Broker

Client
Brokered Delivery Today
Brokered Delivery Today

Latency & loss Measurements

ISP, device type, location, …
Brokered Delivery Today

Which cluster to receive from

Which CDN to use
Brokered Delivery Today
Example

CDN Pricing

Content Provider (CP)

CDN X

CDN Y

Client

CDN X

CDN Y

Client

Client

Client

Client

Client

Client

CDN Pricing

CDN X

CDN Y

Client

Client

Client

Client

Client
Example
Example
CDN X can compete with other CDNs across regions
Evaluation

• Simulator using data from a broker & CDN, as well as public data from 13 other CDNs

• CDN data provides cluster locations, cluster-to-client performance, delivery costs, etc.

• Broker data provides client locations, request distributions, etc.
Evaluation Takeaways

• Today’s world (Brokered) is pretty broken (performance can be better; most CDNs lose money on brokered video delivery)

• Marketplace (VDX) fixes this by exposing clusters and cost
Control Coordination

Scenario: Admin
VDX
App TE + ISP TE
Reaction

Scenario: Scalability
VDN
BGP + BGP
Priority Ranking
Internet-scale Routing
Hierarchical Partitioning

Coflow
Etalon
Transparency
Layering
Control Coordination

Scenario: Admin
VDX

Scenario: Scalability
VDN

Reaction
App TE + ISP TE
BGP + BGP
Priority Ranking

Internet-scale Routing
Hierarchical Partitioning

Transparency
Etalon
Coflow
Scenario: Layering
Control Coordination

Scenario: Scalability

VDN

App TE + ISP TE

Reaction

Internet-scale Routing

Hierarchical Partitioning

BGP + BGP

Priority Ranking

Scenario: Admin

Some Information Sharing

VDX

Coflow

Etalon

Transparency

Scenario: Layering

Full
Control Coordination

Scenario: Scalability

VDN

App TE + ISP TE
Reaction
Internet-scale Routing
Hierarchical Partitioning

BGP + BGP
Priority Ranking
Scenario: Admin

VDX

Some Information Sharing

Coflow
Transparency
Scenario: Layering

Etalon
Full
Live Video is Becoming Wildly Popular

- Commercial sports streams
- User-generated streams
Live Video is Becoming Wildly Popular

• Commercial sports streams
  • **Single** World Cup stream = 40% global Internet traffic
• User-generated streams (e.g., Twitch)
  • Users watch **150b min of live video per month**
  • Amazon buys Twitch for ~~$1 Billion~~
CDN Live Video Delivery Background

Video Sources

Reflector Clusters

Edge Clusters

Legend

Requests:
- Video 1
- Video 2

Responses:
- Video 1
- Video 2

Video Requests

HTTP GET

HTTP RESPONSE
CDN Live Video Delivery Background

- **Video Sources**
  - Video Sources A and B

- **Reflector Clusters**
  - Reflector Clusters C and D

- **Edge Clusters**
  - Edge Clusters E, F, and G

- **Clients**
  - Clients H, I, and J

- **Link Cost**
  - Link Costs: A to C (25), C to E (20), C to F (15), C to G (300)

- **Link Capacity**
  - Link Capacities: A to C (3K), C to E (750), C to F (2K), C to G (1K)

- **DNS**
  - DNS for A and B

- **Data Control**
  - Data Flow from A and B to C, from C to E, F, and G, then to H, I, and J
Objective: Reasonable service quality & Minimal delivery cost
Problems with CDNs Today

Service Quality

- Simulation using Conviva traces, modeling user-generated content

Delivery Cost (per request)
- Simulation using Conviva traces, modeling large sports events

Average Bitrate (Kbps)

- CDN: 2.0x
- Optimal: 1.0x
Problems with CDNs Today

Service Quality

Delivery Cost

Not Fine-Grained

Slow DNS Updates

Videos aggregated into large groups

Can’t push updates

DNS entries get cached

Optimal CDN

Optimal

CDN

2.0x

1.0x

Quantitative

Qualitative
Solution?

Service Quality

Not Fine-Grained

Centralization!

[Liu, Xi et. al. A Case for a Coordinated Video Control Plane. SIGCOMM 2012]
Outline

Problems with Live Video Today

Centralized Control

Distributed Control

Hybrid Control

Putting it all Together
Motivating Centralized Optimization
Motivating Centralized Optimization

- **Video Sources**
- **Reflector Clusters**
- **Edge Clusters**
- **Clients**

The diagram illustrates the flow of data and control between various components. The network is shown with nodes labeled A to J, each representing different points in the system. Arrows indicate the direction of data transfer, and numbers denote link capacities.

Congestion is indicated at certain points, suggesting areas where optimization is needed. The diagram highlights the importance of centralized optimization in managing network congestion and improving efficiency.
Motivating Centralized Optimization

- **Video Sources**
- **Reflector Clusters**
- **Edge Clusters**
- **Clients**

The diagram illustrates the flow of data and control signals between different components, highlighting the link capacity and DNS management. The nodes A, B, C, D, and so on, represent various system elements such as video sources, reflector clusters, edge clusters, and clients.
Motivating Centralized Optimization

Needs global view to coordinate videos and network resources

Video Sources

Edge Clusters

Clients

Reflector Clusters

Clients Link Capacity

DNS

500
300
750
700
300
200
200
300
500
700
750
Unfortunately… No Free Lunch

Experiments on EC2 nodes with a centralized controller at CMU across the Internet
Outline

Centralized Control

Distributed Control

Slow join times

Problems with Live Video Today

Hybrid Control

Putting it all Together
Alternate Approach: Distributed

Legend

Data Requests:
- Video 1

Responses:
- Video 1

Link Capacity
Alternate Approach: Distributed

Build “distance-to-video” tables at each cluster, for each video
Alternate Approach: Distributed

**Legend**

Data Requests:
- Video 1

Responses:
- Video 1

**Link Capacity**

**Video Sources**

**Reflector Clusters**

**Edge Clusters**

**Clients**

**Central Controller**

DISTANCE AT CLUSTER F

VIDEO 1: VIA C: 2; (B, 1K) VIA D: 1; (D, 800)

PICK SHORTEST PATH WITH ENOUGH CAPACITY
Alternate Approach: Distributed

Distributed decisions fast (ms) but sub-optimal

Video Sources

Edge Clusters

Clients

Legend

Data Requests:

Video 1

Responses:

Video 1

DISTANCE AT CLUSTER F

VIDEO 1:

VIA C: 2; (B, 1K)

VIA D: 1; (D, 800)

PICK SHORTEST PATH WITH ENOUGH CAPACITY
Alternate Approach: Distributed

Combine approaches? "Hybrid Control"

Legend

Data Requests:
- Video 1
- Link Capacity

Responses:
- Video 1

Distances at Cluster F:
- Video 1: via C: 2; (B, 1K)
- via D: 1; (D, 800)

Pick shortest path with enough capacity
Combining Approaches: **VDN**

- **Video Sources**
- **Reflector Clusters**
- **Edge Clusters**
- **Clients**

**Legend**
- **Data Requests:**
  - Video 1
  - Data Requests: 2K, 3K, 800
- **Responses:**
  - Video 1
  - Responses: 2K, 3K, 800

**Central Controller**

**The Internet**

**High Latency**

**Clients:**
- **H:**
- **I:**
- **J:**

**Video Sources:**
- **A:** 800
- **B:** 2K
- **C:** 1K
- **D:** 2K

**Reflector Clusters:**
- **C:** 2K
- **D:** 800

**Edge Clusters:**
- **E:** 500
- **F:** 300
- **G:** 700
- **H:** 500
- **I:** 300
- **J:** 800
- **Central Controller:**

**The Internet:**

**High Latency**

**Combining Approaches:**

**VDN**
Combining Approaches: VDN

Legend

Data Requests:
- Video 1

Responses:
- Video 1

Video Sources

Reflector Clusters

Edge Clusters

Clients

The Internet
Combining Approaches: **VDN**

**Legend**
- **Data Requests:**
  - Video 1: 2K
  - Video 2: 3K
- **Responses:**
  - Video 1: 800
  - Video 2: 800
- **Control Traffic:**
  - Video 1: 800

**Video Sources**
- A: 2K
- B: 1K
- C: 2K
- D: 800

**Reflector Clusters**
- E: 300
- F: 700

**Edge Clusters**
- G: 800

**Clients**
- H: 800
- I: 800
- J: 800

**Central Controller**

**The Internet**

**HIGH LATENCY**

---

The diagram illustrates the VDN (Video Delivery Network) approach combining video sources, reflector clusters, edge clusters, and clients. The central controller is managing data requests and responses to optimize video delivery, reducing latency. The legend provides a key for understanding the traffic and requests involved in the VDN system.
Challenges of Hybrid Control

- Forwarding loops
  - Always forward requests upwards

- State transitions
  - Versioning and “shadow FIBS”

- Avoid bad control loop interactions
Challenges of Hybrid Control

• Avoid bad control loop interactions

1. Centralized decision has priority
2. Distributed uses slack in network
Hybrid Control and Responsiveness

Experiments on EC2 nodes with a centralized controller at CMU across the Internet
Hybrid Control and Responsiveness

Experiments on EC2 nodes with a centralized controller at CMU across the Internet
Hybrid Control and Responsiveness

Experiments on EC2 nodes with a centralized controller at CMU across the Internet
Control Coordination

**Scenario:**

- **Scalability**
  - VDN
  - App TE + ISP TE
  - Internet Routing
  - Reaction
  - Hierarchical Partitioning

**Priority Ranking**

- BGP + BGP

**Transparency**

- VDX
  - Coflow
  - Etalon
  - Scenario: Layering

**Information Sharing**

- Some
- Full
Control Coordination

Scenario: Scalability
- VDN
- App TE + ISP TE
- Reaction
- Internet Routing
- Hierarchical Partitioning

Scenario: Information Sharing
- Priority Ranking
  - BGP + BGP
- Transparency
  - VDX
  - Coflow
  - Etalon
- Admin
- Layering

Some Information Sharing
Full Information Sharing
Control Coordination

- Internet Routing
- VDN
- Hierarchical Partitioning
  - Scenario: Scalability
- Priority Ranking
  - Scenario: Admin
- BGP + BGP
- VDX
- Coflow
- Etalon
- Transparency
- Scenario: Layering

App TE + ISP TE

Reaction

Some Information Sharing

Full Information Sharing
Control Coordination

- Hierarchical Partitioning
  - Scenario: Scalability

- Internet Routing
  - VDN

- BGP + BGP
  - VDX

- Priority Ranking
  - Scenario: Admin

- App TE + ISP TE
  - Reaction

- Shared resources
  - Yes/No

- Information Sharing
  - Some/Full

- Transparency
  - Coflow/Etalon

- Scenario: Layering
Control Coordination

Shared resources

Yes

(vanilla) DASH + HTTP + TCP

Reaction

App TE + ISP TE

Congestion Control

OSPF Areas + AQM

No

None

Preventive

App TE + ISP TE

OSPF Areas + AQM

CC + AQM

Some

Hierarchical

Pytheas Bohatei Klein VDN

Internet Routing

OSPF Fibbing

C3 (Conviva)

Partitioning

Scenario: Scalability

Priority Ranking

BGP + BGP

P4P

Wiser

VDX

Route Redistribution

Transparency

Coflow

Etalon

Full

Scenario: Layering

Information Sharing
Future Work

- Control theory / verification approach
- Validating VDN
- Extending VDX to multi-broker
- Principled approach to reconfigurable datacenters
- Network / endhost co-design
  - e.g., network-aware applications
Control Coordination

Hierarchical Partitioning

Internet Routing

VDN

Scalability

Scenario: Admin

Priority Ranking

BGP + BGP

VDX

None

App TE + ISP TE

Yes

Reaction

Full

Scenatio: Layering

Transparency

Coflow

Etalon

Shared resources

Information Sharing
Eliminating Adverse Control Plane Interactions in Independent Network Systems

Matthew K. Mukerjee

Computer Science PhD Thesis Defense

May 1st, 2018