SPAIN:
High BW Data-Center Ethernet with Unmodified Switches

Jayaram Mudigonda, HP Labs
Mohammad Al-Fares, UCSD

Praveen Yalagandula, HP Labs
Jeff Mogul, HP Labs
Internet-facing applications:
E-Mail, Web Servers, etc.
DC Trends

Information Explosion

HPC Applications

Application Consolidation

Virtualization
DC Trends

Datacenter Fabric

Shuffle phase of Map – Reduce

Internet
DC Trends

Internet

High bisection bandwidth

Shuffle phase of Map – Reduce
DC Trends

Internet

Flat Network
DC Fabric Goals

- High bisection BW
- Flat network
- Low-cost
Ethernet: a good choice

Commodity ➔ Inexpensive

Speeds:

- 10G is here
- 40G/100G soon

Flat-addressing

Self-configuring
But wait…
Spanning Tree Protocol (STP) makes Ethernet hard to scale!
Spanning Tree Protocol (STP)

Root

Bandwidth bottleneck

Unused links
Proposal 1: High-port core switch

A common current approach
Expensive Core Switch

High BW or Multiple Links
Proposal 2: L3

IP Subnetting

VL2 [SIGCOMM’09]
L3 routers

Expensive
No non-IP protocols
Proposal 3: Modify switches (HW/SW)

TRILL [IETF]
SEATTLE [SIGCOMM’08]
PortLand [SIGCOMM’09]

Not deployable today!
SPAIN

Unmodified L2 switches

Multi-pathing

Arbitrary topologies
SPAIN Approach

Multi-pathing
via VLANs
+
End-host driver
to spread load
Multi-pathing via VLANs

Default VLAN

A

C

B

D
Multi-pathing via VLANs
SPAIN

Unmodified L2 switches
Multi-pathing via VLANs
Arbitrary topologies
Minor End-host mods

Low-cost
High-BW
DC Fabric
Today!
Outline

Introduction

SPAIN Components

Offline computation

End-host driver

Evaluation

Summary
Outline

Introduction

SPAIN Components

Offline computation

End-host driver

Evaluation

Summary
Offline Computation

Steps:
1. Discover topology
2. Compute paths
3. Layout paths as VLANs
Discover topology

SNMP Queries
Compute paths

Goal: leverage redundancy; improve reliability

Challenges: large graphs; more paths → more resources
Compute paths

Only consider paths between edge-switches; Modified Dijkstra’s; Prefer edge-disjoint paths
VLAN Layout

Simple scheme: Each Path as VLAN
But…

IEEE 802.1Q:

VLAN ID = 12 bits

⇒ 4096 VLANs!
VLAN Layout

Simple scheme: Each Path as VLAN

Scales to only few switches
VLAN Layout

Our approach: 1 VLAN for a set of paths
Challenge: Minimize VLANs

NP-Hard for arbitrary topologies
VLAN Layout

Heuristics:

1. Greedy path packing
2. Parallel graph-coloring
VLAN Layout

# VLANs = 4
Outline

Introduction

SPAIN Components

Offline computation

End-host driver

Evaluation

Summary
SPAIN End-host Driver
SPAIN End-host Driver
SPAIN End-host Driver

Flow Table
A → B, 1 : RED
A → B, 2 : BLUE

Flow Table
Challenges

Link & switch failures
Pathological flooding
Interoperability
Host mobility
Load-balance
End-host state
Failures

Flow Table
A → B : RED

Flow Table
Pathological Flooding

Does not know the location of B

Flow Table
A → B : RED

Flow Table
B → A : GREEN
Solution:

Chirping
Chirping

Knows the location of B

A

Flow Table
A→B : RED

B

Flow Table
B→A : GREEN
Chirping

Flow Table
A → B: RED

Flow Table
B → A: GREEN
Evaluation

Simulations

Real testbed
Simulations

Topologies:
CiscoDC

Core switches

Aggregation modules
\( m = 2 \)

Access switches per module
\( a = 2 \)
Simulations

Topologies:

CiscoDC  Fat-Tree

[Al-fares et al., SIGCOMM'08]

#ports/switch

p = 4
Simulations

Topologies:

- CiscoDC
- Fat-Tree
- HyperX

2D HyperX
k=4

HyperX
[Ahn et al. SC’09]

Fat-Tree
[Al-fares et al. SIGCOMM’08]
Simulations

Topologies:

- CiscoDC
- Fat-Tree
  - [Al-fares et al. SIGCOMM’08]
- HyperX
  - [Ahn et al. SC’09]
- B-Cube
  - [Guo et al. SIGCOMM’09]

#ports/switch (p) = 2
Levels (l) = 2
Simulations

Topologies:
- CiscoDC
- Fat-Tree [Al-fares et al. SIGCOMM'08]
- HyperX [Ahn et al. SC’09]
- B-Cube [Guo et al. SIGCOMM’09]

Metrics:
- #VLANs
- Link-Coverage
- Reliability
- Throughput
Simulations

Topologies:
- CiscoDC
- Fat-Tree [Al-fares et al. SIGCOMM’08]
- HyperX [Ahn et al. SC’09]
- B-Cube [Guo et al. SIGCOMM’09]

Metrics:
- #VLANs
- Link-Coverage
- Reliability
- Throughput
## Num. of VLANs

<table>
<thead>
<tr>
<th>System</th>
<th>#switches</th>
<th>#VLANs</th>
</tr>
</thead>
<tbody>
<tr>
<td>CiscoDC (8,8)</td>
<td>146</td>
<td>38</td>
</tr>
<tr>
<td>Fat-Tree (48)</td>
<td>2880</td>
<td>576</td>
</tr>
<tr>
<td>HyperX (16)</td>
<td>256</td>
<td>971</td>
</tr>
<tr>
<td>B-Cube (48,2)</td>
<td>2048</td>
<td>2048</td>
</tr>
</tbody>
</table>
Throughput

CiscoDC: 2x
Fat-Tree: 24x
HyperX: 10.5x
B-Cube: 1.6x

Improvement over STP
OpenCirrus Experiments
OpenCirrus Testbed

CORE SWITCH (CS)

RACK SWITCH (RS)

10G

1G

80 blades
OpenCirrus Testbed

RACK SWITCH (RS)

CORE SWITCH (CS)

S1

S2

S3

1G

10G

80 blades
OpenCirrus Testbed

10G links that we added
OpenCirrus Testbed

4 VLANs
Shuffle-like experiment

Every server to all other servers

500 MB data transfer
Spanning Tree Protocol (STP)
Link utilization in each direction
Spanning Tree Protocol (STP)
SPAIN

No bottle-necks
Completion times

~50% reduction
Aggregate Goodput (Gbps)

<table>
<thead>
<tr>
<th>STP</th>
<th>SPAIN</th>
</tr>
</thead>
<tbody>
<tr>
<td>35.6</td>
<td>66.7</td>
</tr>
</tbody>
</table>

87% improvement
Aggregate Goodput (Gbps)

Incremental

% SPAIN hosts

0%  20%  50%  70%  100%

35.6  37.0  44.7  56.0  66.7
Single Shortest Path (SSP) SEATTLE/TRILL

All flows on RED

All flows on GREEN

SEATTLE/TRILL on unmodified switches with
Comparison with SSP

Completion Time(s)

- SSP: 513 s
- SPAIN: 431 s

16% better

Goodput (Gbps)

- SSP: 62.3 Gbps
- SPAIN: 66.7 Gbps

7% better
SPAIN Take-away

Unmodified L2 switches
Multi-pathing via VLANs
Arbitrary topologies
Minor End-host mods

Low-cost
High-BW
DC Fabric
Today!
Q&A