Jellyfish: Networking Data Centers, Randomly

Ankit Singla*, Chi-Yao Hong*
Lucian Popa+, P. Brighten Godfrey*
*UIUC  +UC Berkeley

HotCloud’11
Data center network architecture is important

- Data centers today form the backbone of cloud operations
- A good datacenter network architecture ensures
  - High bandwidth
  - Isolated services among flows
  - High freedom in workload placement
  - ...
- Fat-tree, VL2, Portland, BCube, DCell, Helios, c-through, Proteus
Incremental expansion to react to user growth

- **Incremental expansion**: Adding servers and network capacity incrementally

- Facebook has ~30,000 servers in Nov 2009 and ~60,000 by June 2010

- Much of this growth is more incremental in existing facilities (“adding capacity on a daily basis” [1])

Incremental growth difficult in current designs

Fat-tree

- Very few discrete design points
- Expansion from one fat-tree to larger fat-tree requires replacing *all* the switches
Roadmap

- **Jellyfish**: A random graph switch-level interconnect

  - **Benefits**: flexibility and efficiency
  
  - **Challenges**: routing and cabling
Jellyfish: A random regular graph at switch layer

Random

Uniform randomly selected from all regular graphs

Regular

Each node has the same degree

Graph

Nodes: Switches

Edges: Network Cables
Network Construction and Expansion

1) Randomly pick a random pair of nodes with free network port
2) Join them with an edge

The same procedure can be used to expand network incrementally
Roadmap

- **Jellyfish**: A random graph switch-level interconnect

- **Benefits**: flexibility and efficiency

- **Challenges**: routing and cabling
Jellyfish provides incremental expandability

- Expended Jellyfish has the same topological properties

![Graph showing average path length between switches vs number of servers in thousands.](image)
Jellyfish provides high resilience

- High resilience to failures
  - Regular random graph is almost surely $r$-connected
  - Link failures just produce a smaller random graph
Jellyfish has better bisection bandwidth

- Bisection bandwidth = \emph{worst-case} bandwidth between two equal-size partitions of the network

\[ \text{Normalised [0-1] bisection bandwidth} \]

\[ \text{Number of servers in thousands} \]

3-level Fat-tree with 2000 switches of 40 ports

Jellyfish; equal-cost curve [1]

Roadmap

- **Jellyfish**: A random graph switch-level interconnect

  - **Benefits**: flexibility and efficiency

  - **Challenges**: routing and cabling
Routing

Problem
- No simple routing due to lack of structure
- More difficult load balancing

Solution
- Route via equal cost paths or via longer paths
- Centralized scheduling for elephant flows; Randomization for mice flows
Cabling: Aggregating cables to minimize manual labor

- $\sqrt{n}$ clusters, each of which has $\sqrt{n}$ switches
Conclusion and future work

- Jellyfish is a random regular graph at switch layer
  - Flexible expansion
  - More efficient for the same cost

- Future work
  - Routing protocol
  - Comparing with optimal graphs
  - Quantifying cabling cost