ElasticTree: Saving Energy in Data Center Networks

Brandon Heller⋆ , Srini Seetharaman† , Priya Mahadevan⋄ , Yiannis Yiakoumis⋆ , Puneet Sharma⋄ , Sujata Banerjee⋄ , Nick McKeown⋆ ,

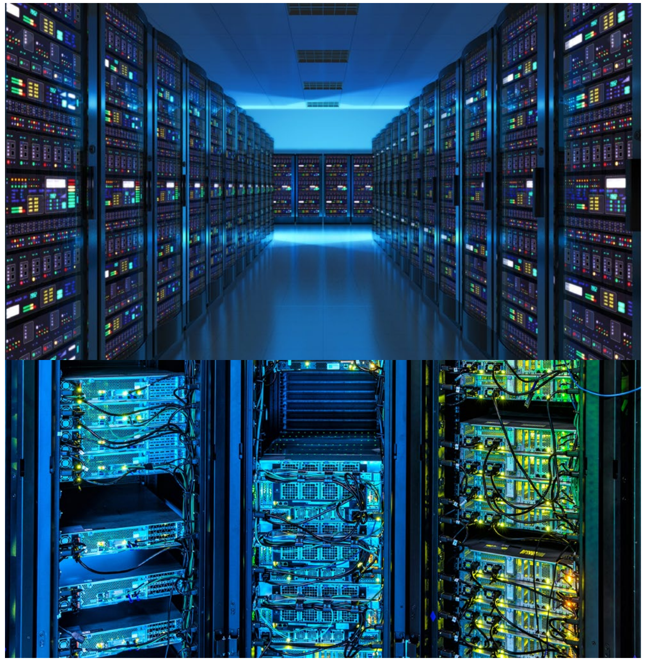
Nsdi. Vol. 10. 2010.

22.12.06.
Cloud Computing
Sohee Kim

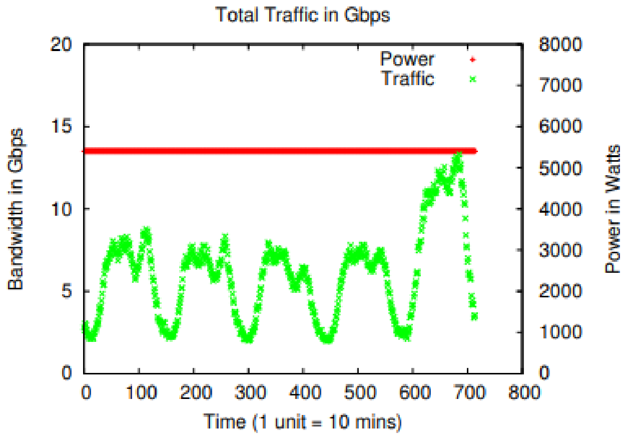
[PDF] Elastictree: Saving energy in data center networks.
B Heller, S Seetharaman, P Mahadevan, Y Yiakoumis… - Nsdi, 2010 - usenix.org
Networks are a shared resource connecting critical IT infrastructure, and the general practice is to always leave them on. Yet, meaningful energy savings can result from improving a network's ability to scale up and down, as traffic demands ebb and flow. We present ElasticTree, a network-wide power 1 manager, which dynamically adjusts the set of active network elements—links and switches—to satisfy changing data center traffic loads. We first compare multiple strategies for finding minimum-power network subsets across a range of …

[PDF] usenix.org

1. Introduction



- Data centers consume huge amounts of energy
- Most efforts to **reduce energy consumption** in Data centers is focused on servers and cooling, which account for about 70% of a data center's total power budget
- **Networking elements** are responsible for 10~20% of power usage in a data center
  - 3 billion kWh in 2006[US] by networking elements -> rising
  - This paper focus on **reducing growing energy cost** (**network power consumption)**
- Data centers are typically provisioned for peak workload
- During lower traffic, most of the network components are idle, still using power

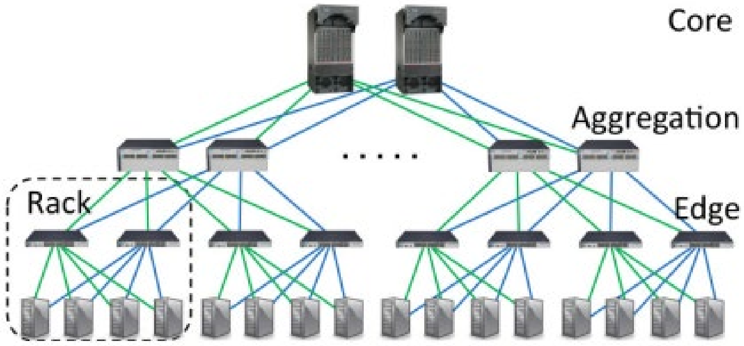


- Traffic collected from 292 servers hosting an E-commerce application over a 5 days period
- The traffic peaks during the day and falls at night
- Even though traffic varies significantly with time, the associated switches draw a constant power

1. Introduction

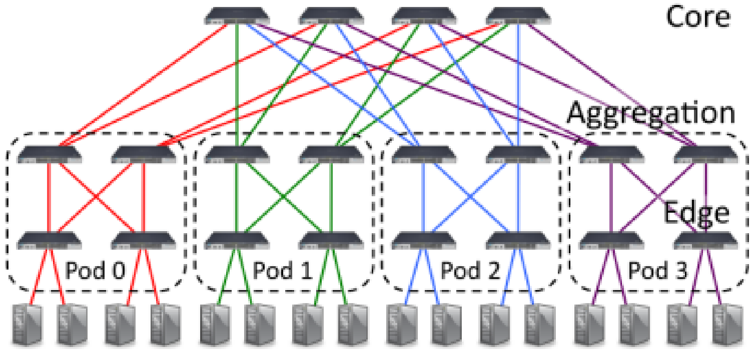
❖ Data Center Networks

✓ **Typical Data Center Network (2N)**



✓ **Fat tree**

➢ Built from a large number of richly connected switches and can support any communication pattern.



- In a typical DCN(2N tree) - One failure can cut the effective bisection bandwidth in half, while two failures can disconnect servers.
- Richer, mesh-like topologies (like the fat-tree) handle failures more gracefully
  - With more components and more paths, the effect of any individual component failure becomes manageable
- This can be used in improving energy efficiency, by dynamically varying the number of active network elements
- It can be thought as a control knob to tune between energy efficiency, performance and fault tolerance

1. Introduction

❖ Energy Proportionality

- Unfortunately, today's network elements are **not energy proportional**
  - Fixed overheads such as fans, switch chips, and transceivers waste power at low loads.
- However, maximum efficiency can be realized by a combination of improved components and improved component management.

⇒ Choice : To manage today's non energy-proportional network components more intelligently

⇒ The Strategy is simple

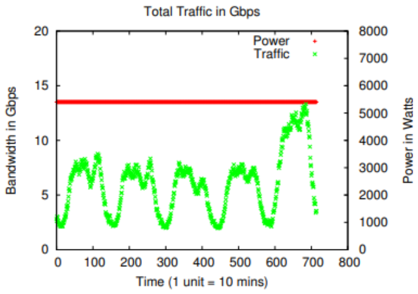
  ⇒ Turn off the links and switches that we don't need to keep available only as much networking capacity as required

➢ **ElasticTree** is a network-wide energy optimizer that continuously monitors data center traffic conditions

✓ It can reduce the energy consumed by a data center network by up to 60% by **simply turning off unneeded links and switches**

✓ The challenge is to save energy without affecting performance or fault tolerance in a way that scales to data centers with a hundred thousand nodes

✓ **Goal**

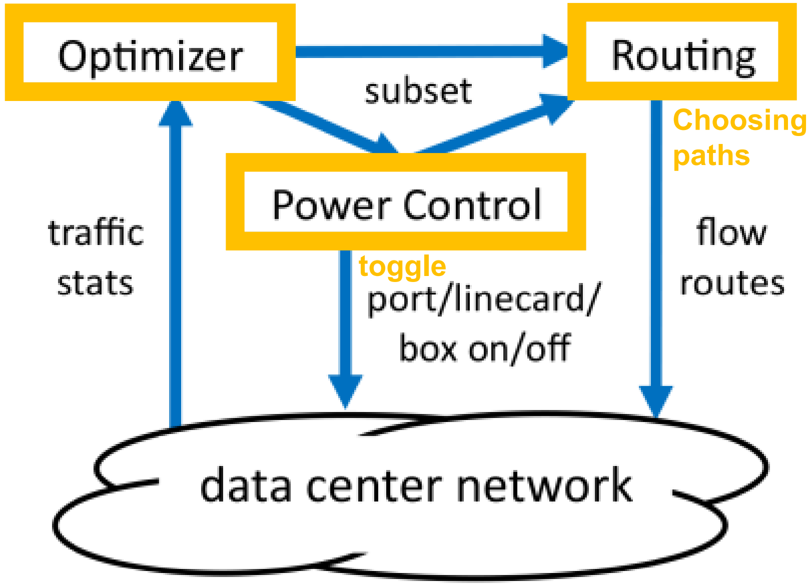


Make this graph look like this!

2. ELASTIC TREE

❖ ElasticTree

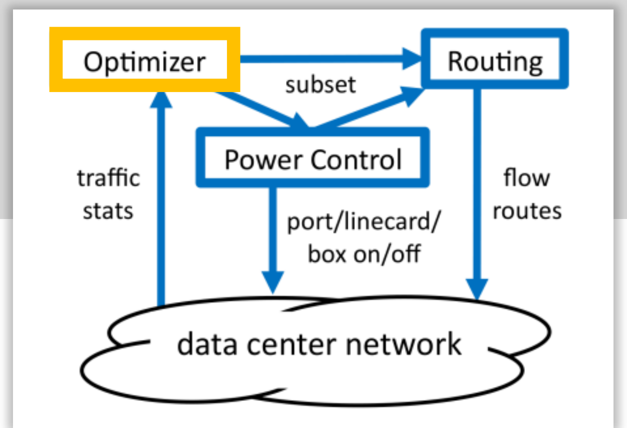
- It is a network-wide power manager, which dynamically adjusts the set of active network elements-links and switches- to satisfy changing data center traffic loads
- It consists of three logical modules : Optimizer, Routing, Power Control



✓ **System Diagram**

1. **Optimizer**
   - To find the minimum power network subset which satisfies current traffic conditions
   - Inputs : topology, traffic matrix, a power model for each switch, desired fault tolerance properties
   - Outputs : a set of active components to both the power control and routing modules
2. **Routing**
   - Choosing paths for all flows then pushes routes into the network
3. **Power Control**
   - Toggles the power states of ports, line cards, and entire switches

2. ELASTIC TREE



❖ Optimizers

Type	Quality	Scalability	Input	Topo
Formal	Optimal [3]	Low	Traffic Matrix	Any
Greedy	Good	Medium	Traffic Matrix	Any
Topo-aware	OK	High	Port Counters	Fat Tree

Table 2: Optimizer Comparison

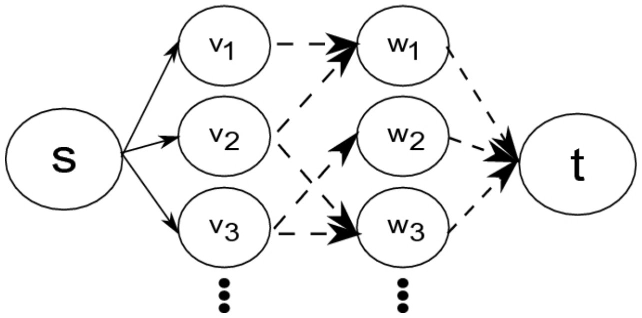
➢ A range of methods to compute a minimum-power network subset in Elastic Tree

- Role is to find minimum-power network subset which satisfies current traffic conditions
- There are 3 different methods for computing a minimum power network subset
  1. **Formal Model**
  2. **Greedy-Bin Packing**
  3. **Topology-aware Heuristic**
- Each method achieves different tradeoffs between **scalability**(확장성) and **optimality**(최적성)
- Methods can be further improved by considering a data center's traffic history

2. ELASTIC TREE

Type	Quality	Scalability	Input	Topo
Formal	Optimal [3]	Low	Traffic Matrix	Any
Greedy	Good	Medium	Traffic Matrix	Any
Topo-aware	OK	High	Port Counters	Fat Tree

❖ Optimizers

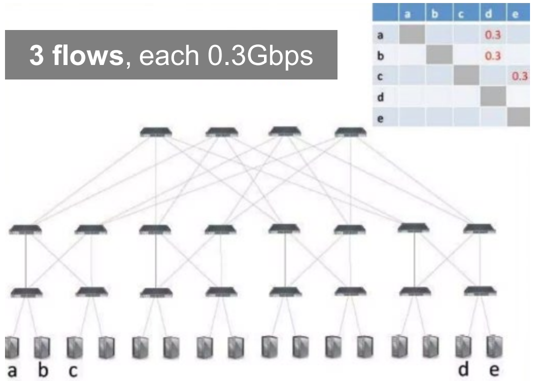


1. **Formal Model**
   - Extension of the standard multi-commodity flow (MCF) problem with additional constraints which force flows to be assigned to only active links and switches
   - The constraints include link capacity, flow conservation and demand satisfaction
   - Minimize $\sum(Link + Switch\ Power)$
     - The optimization goal is **to minimize the total network power, while satisfying all constraints**
   - The model outputs a subset of the original topology, plus the routes taken by each flow to satisfy the traffic matrix.

Type	Quality	Scalability	Input	Topo
Formal	Optimal [3]	Low	Traffic Matrix	Any
Greedy	Good	Medium	Traffic Matrix	Any
Topo-aware	OK	High	Port Counters	Fat Tree

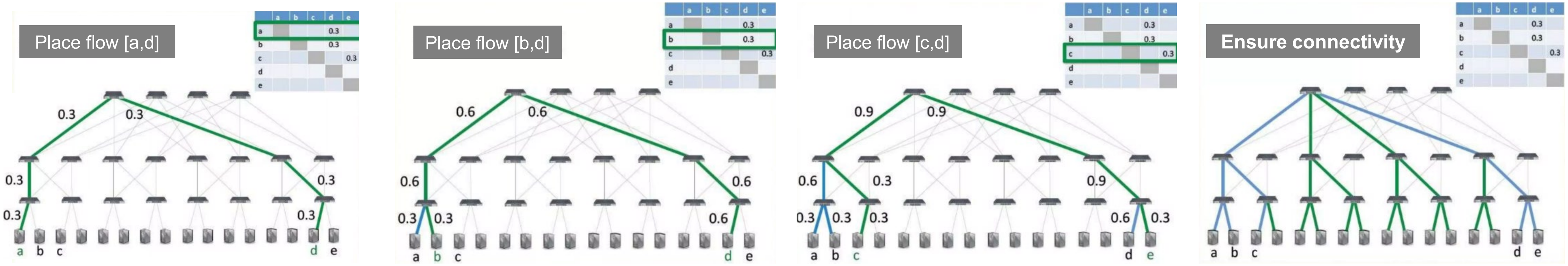
2. ELASTIC TREE

❖ Optimizers



2. Greedy Bin-Packing

- The greedy bin-packing heuristic improves on the formal model's scalability
- Greedy bin-packer evaluates possible paths and chooses the leftmost one with sufficient capacity.
- Repeated for all flows
- Solutions within a bound of optimal are not guaranteed, but in practice, high quality subsets result.



Heller, Brandon, et al. "Elastictree: Saving energy in data center networks." Nsdi. Vol. 10. 2010.

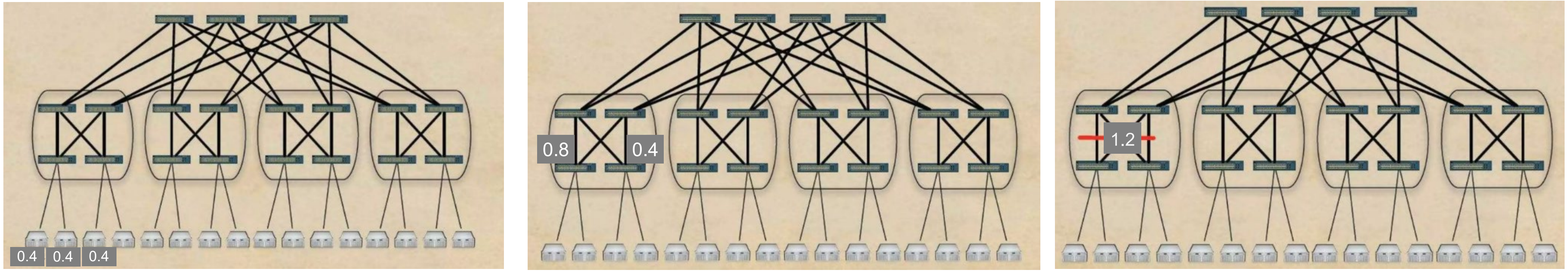
2. ELASTIC TREE

Type	Quality	Scalability	Input	Topo
Formal	Optimal [3]	Low	Traffic Matrix	Any
Greedy	Good	Medium	Traffic Matrix	Any
Topo-aware	OK	High	Port Counters	Fat Tree

❖ Optimizers

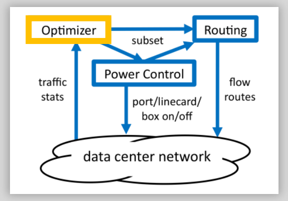
3. Topology-Aware Heuristic

- Leverages regularity of fat tree topology to quickly find network subsets
- Unlike the other methods, it does not compute the set of flow routes, and assumes perfectly divisible flows.
- Intuition : “An edge switch doesn’t care which aggregation switches are active, instead, how many are active”



Heller, Brandon, et al. "Elastictree: Saving energy in data center networks." Nsdi. Vol. 10. 2010.

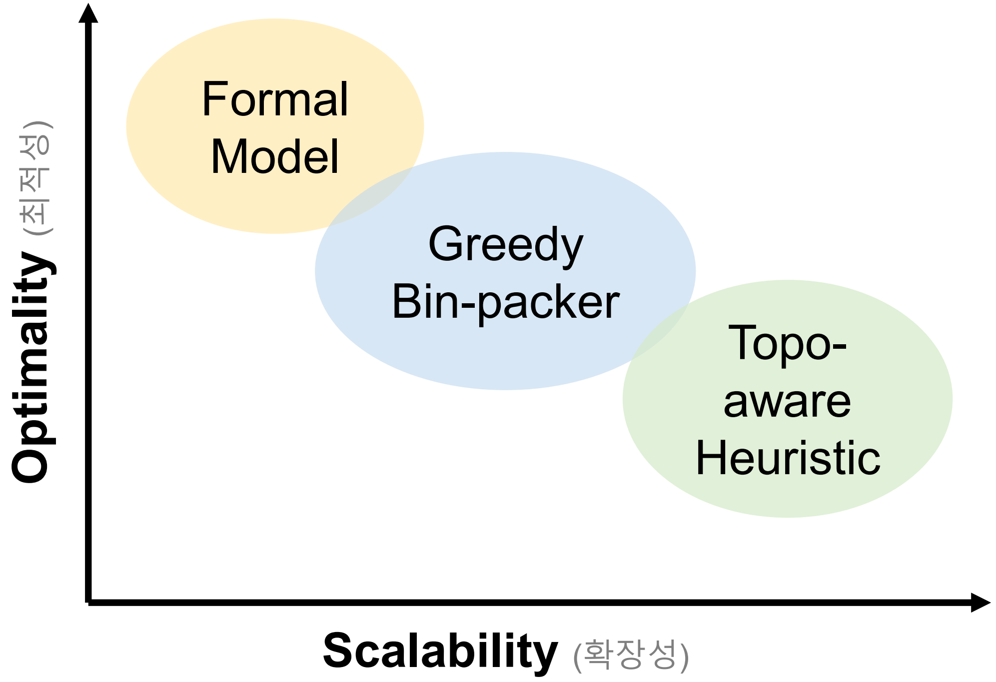
2. ELASTIC TREE



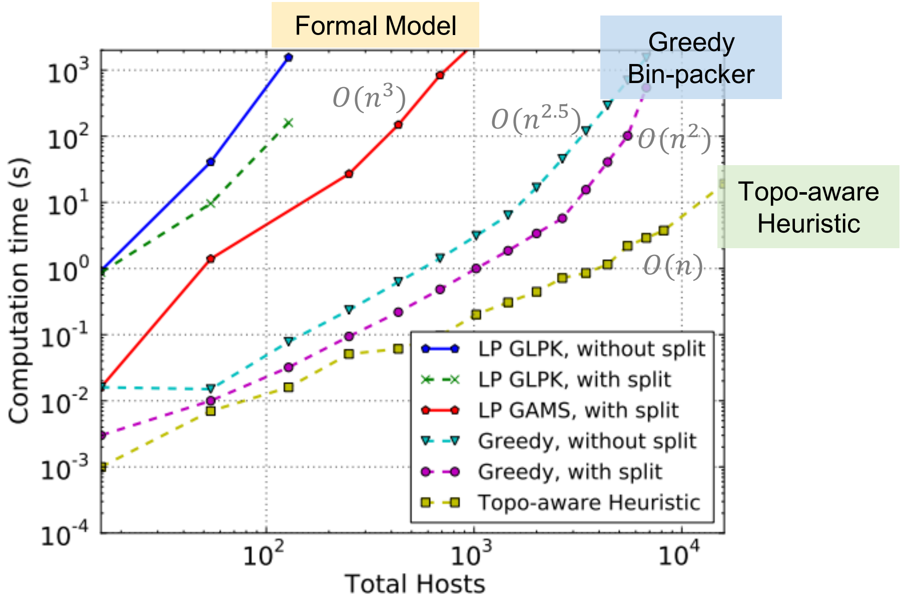
Type	Quality	Scalability	Input	Topo
Formal	Optimal [3]	Low	Traffic Matrix	Any
Greedy	Good	Medium	Traffic Matrix	Any
Topo-aware	OK	High	Port Counters	Fat Tree

❖ 3 Optimizers

- Outputs a network topology subset, used by the control software



▪ Scalability



➢ The topology-aware heuristic approach is not fundamentally unscalable, especially considering that the number of operations increases linearly with the number of hosts.

3. POWER SAVING ANALYSIS

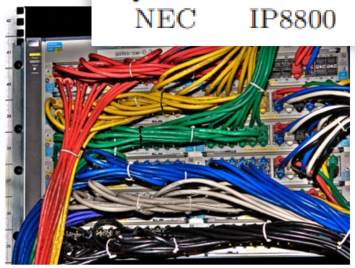
$$\%\text{ of original network power} = \frac{\text{Power consumed by ElasticTree} \times 100}{\text{Power consumed by original fat tree}}$$

== the overall power saved by turning off switches and links

- Small tests performed on prototypes
- Larger networks tested through simulations
- Considered power usage:
  - Number of switches powered on
  - Number of ports enabled on them
- Ignored power usage:
  - Running servers hosting ElasticTree modules
  - Cooling components:
    - additional energy for cooling servers
    - decreased energy for cooling switches

✓ Fat Tree Configuration

Vendor	Model	k	Virtual Switches	Ports	Hosts
HP	5400	6	45	270	54
Quanta	LB4G	4	20	80	16
NEC	IP8800	4	20	80	16

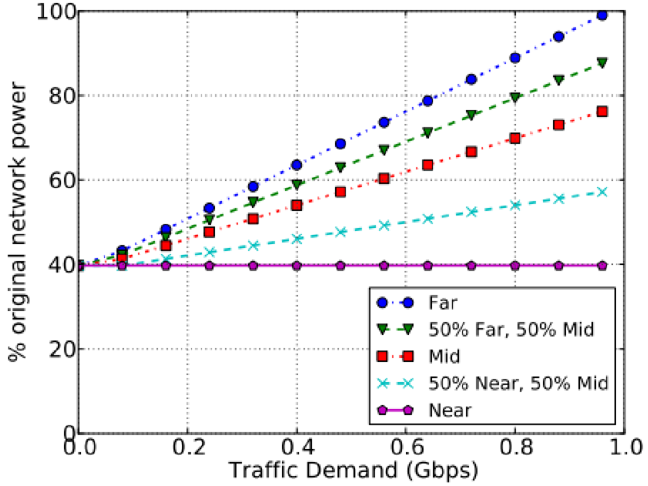


▪ Traffic Patterns

> Energy, performance and robustness

✓ Uniform Demand, Varying Locality

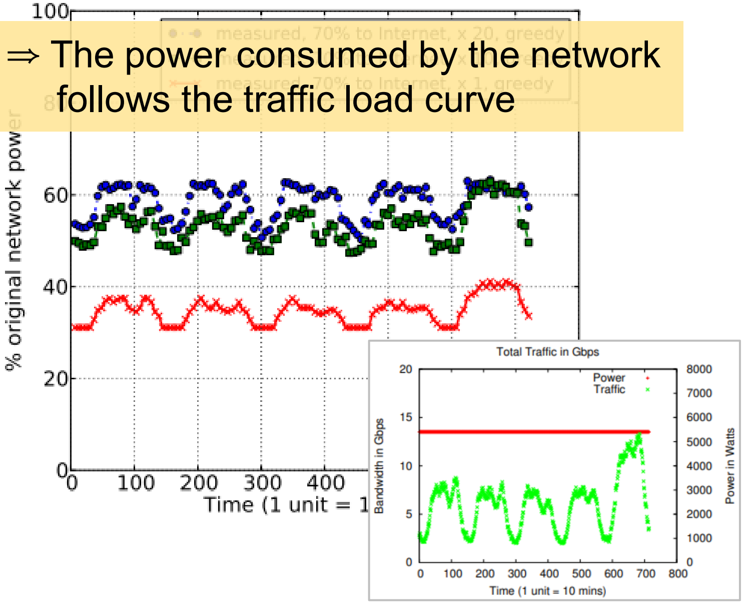
- 28K-node, k=48 fat tree
- Near = Highly localized
- Far = non-localized



➢ Power savings as a function of demand, with varying traffic locality
  ➢ Near traffic is the best-case
  ➢ Far traffic is the worst-case

✓ Simulation on real traffic data

- E-commerce website, 292 servers
- 5-day period, k=12 fat tree



⇒ The power consumed by the network follows the traffic load curve

➢ Energy savings for production data center traces
➢ 70% of the traffic leaves the data center
➢ Energy savings ranging from 25-62%

3. POWER SAVING ANALYSIS

- **Robustness Analysis**

- Network topology must be prepared for:
  - Traffic surges
  - Network failures
- Adding a minimum spanning tree (MST) to the power optimized topology enables one failure with no loss of connectivity
- Additional energy cost decreases with the size of the topology

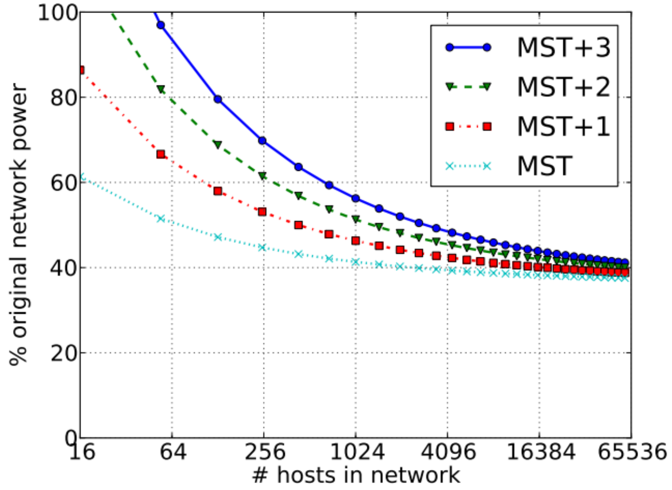


Figure 11: Power cost of redundancy

- ➢ Each +1 increment in redundancy has an additive cost, but a multiplicative benefit;
  - MST+2, the failures would have to happen in the same pod to disconnect a host.
- ➢ The added cost of fault tolerance is low

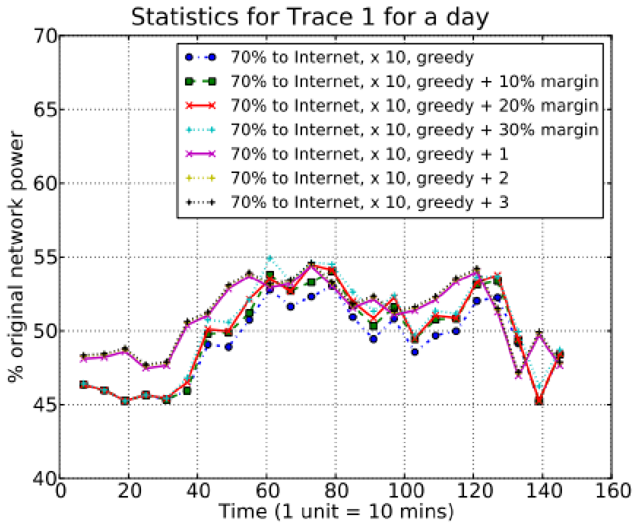


Figure 12: Power consumption in a robust data center network with safety margins, as well as redundancy. Note "greedy+1" means we add a MST over the solution returned by the greedy solver.

- ➢ The additional power cost incurred is minimal, while improving the network's ability to absorb unexpected traffic surges.

4. Performance

✓ **Dropped packets**

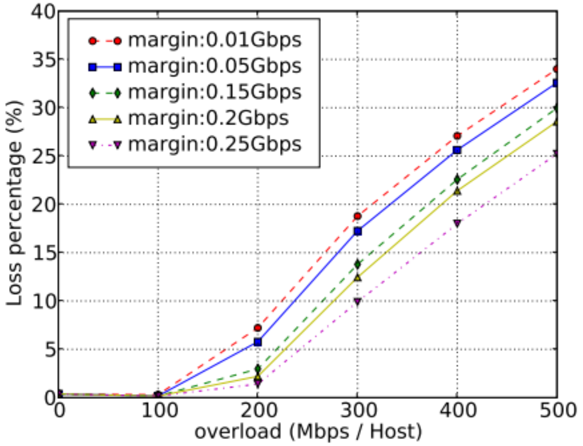


Figure 15: Drops vs overload with varying safety margins

✓ **Latency**

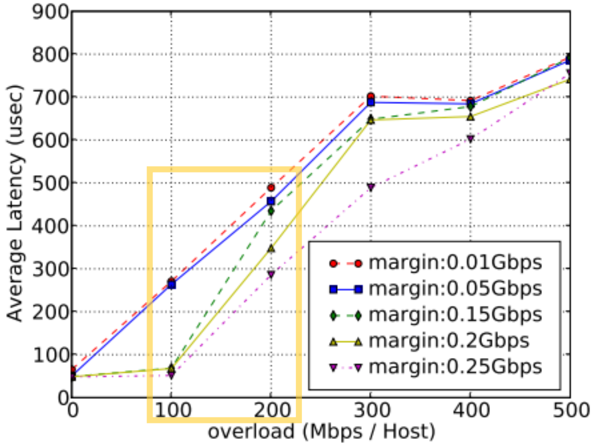


Figure 16: Latency vs overload with varying safety margins

- Safety margin : the amount of capacity reserved at every link by the optimizer
  - A higher safety margin provides performance insurance, by delaying the point at which drops start to occur, and average latency starts to degrade
- Traffic overload : the amount each host sends and receives beyond the original traffic matrix

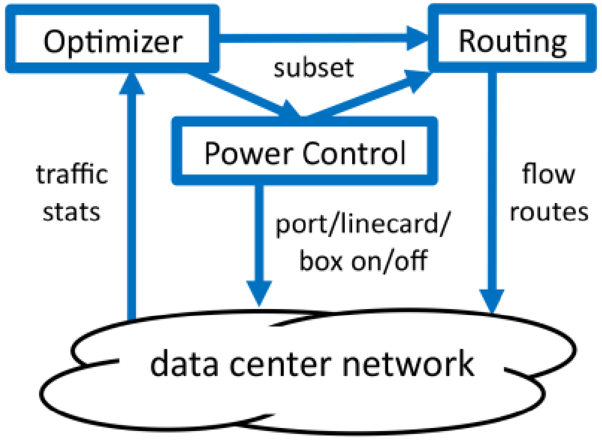
➢ No drops for small overloads (up to 100 Mbps), followed by a steadily increasing drop percentage as overload increases

➢ As expected, increasing the safety margin defers the point at which performance degrades

➢ Latency shows a trend similar to drops, except when overload increases to 200 Mbps, the performance effect is more pronounced

⇒ Given these plots, a network operator can choose the safety margin that best balances the competing goals of performance and energy efficiency.

5. Conclusion



❖ ElasticTree

- Introduces energy proportionality in today's non-energy proportional networks (which through data-center-wide traffic management and control)
- A first step toward a more efficient data center network.

- 3 Algorithms – Formal model, Greedy Bin-Packing and Topology-aware Heuristic

  ⇒ applied on E-commerce data center, and found that power consumption can be reduced
- The highly flexible system allows for balancing between performance, robustness and energy
- Initial results (covering analysis, simulation, and hardware prototypes) suggest very significant power benefits for networks with varying utilization
- ElasticTree's ability to respond to sudden increases in traffic is currently limited by the switch boot delay, but this limitation can be addressed, relatively simply, by adding a sleep mode to switches
- During periods of low to mid utilization, and for a variety of communication patterns (as is often observed in data centers), ElasticTree can maintain the robustness and performance, while lowering the energy bill

Thank You ☺