Elastic Tree: Saving Energy in Data Center Networks

Brandon Heller, David Underhill, Srinivasan Seetharaman, Nick McKeown

Presented By:-
Aditya Kumar Mishra

Introduction

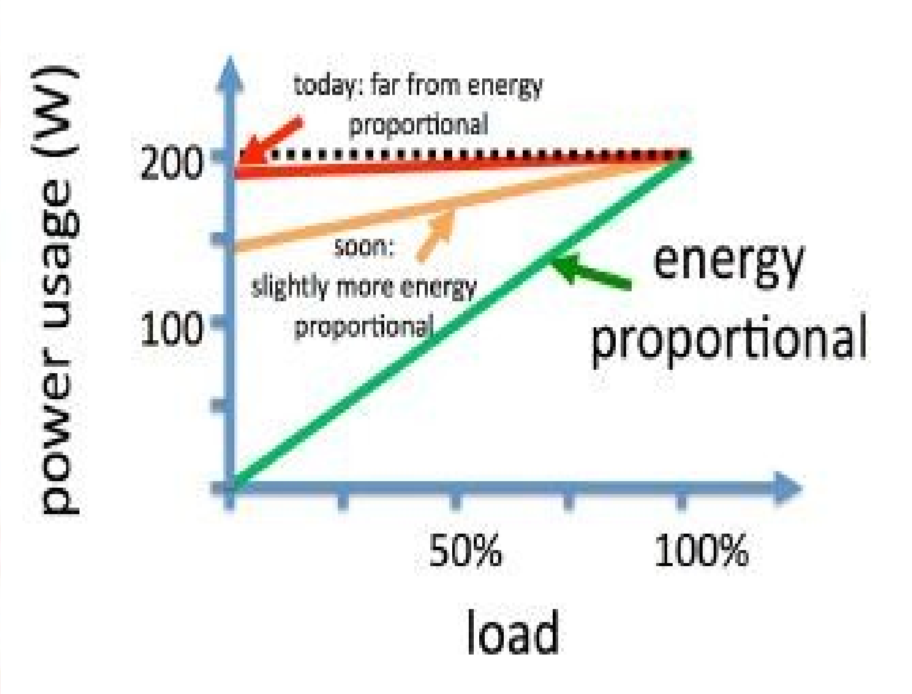
- Currently, most efforts focused at optimizing energy consumption at servers
- Network consumes 10-20% of Data center power

Introduction (Contd)

Try and minimize two things

- Energy consumed by network components
- Number of active components

Energy Proportionality



- If each component is energy proportional, we don't need to minimize the number of active components

Elastic Tree approach

- **Input:** Network topology and traffic matrix
- **Decide**, how to route packets to minimize energy
- **After rerouting**, power down all possible links and switches
- **Balance** performance and fault tolerance

Data Center Networks

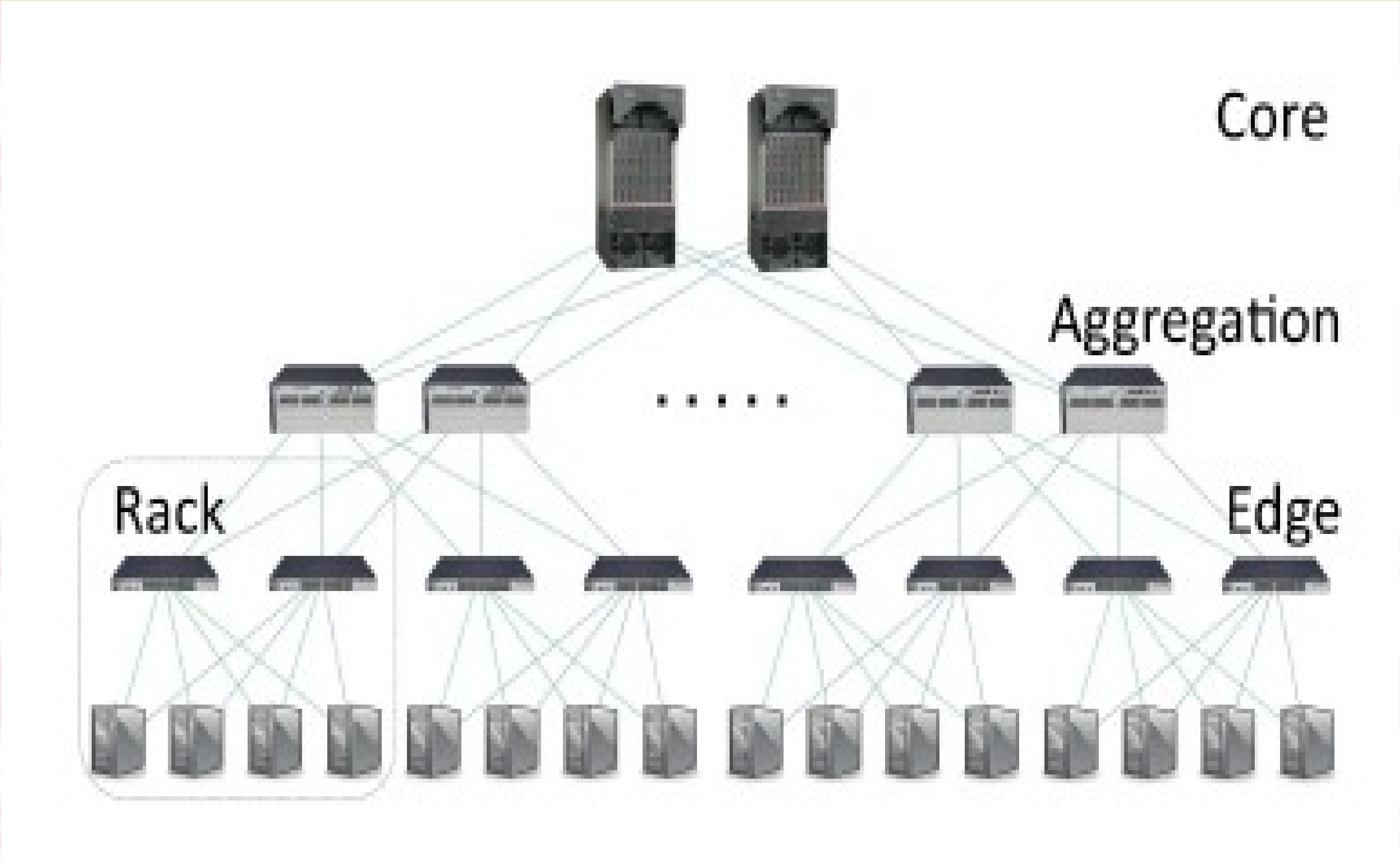
Data Center Networks

- Are **big:** Scale to over 100000 servers and 3000 switches
- Are **structured:** Employ regular tree like to-pologies with simple routing
- Are **cost-sensitive**

Typical Data Center Network

- Often built using 2N topology
- Every server connects to two edge switches
- Every switch connects to two higher layer switch and so on

Typical Data Center Network



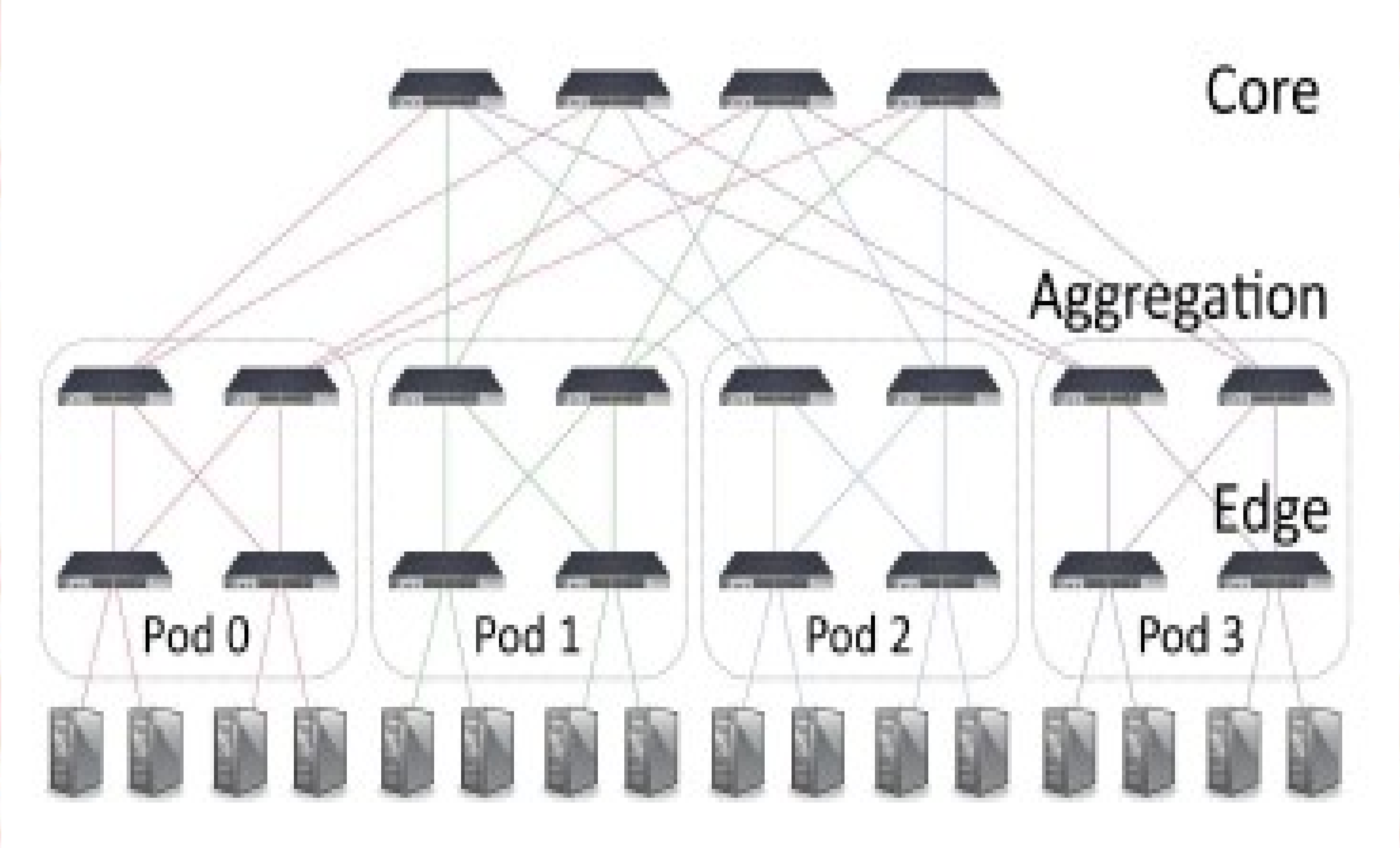
Traffic and Provisioning

- Typically provisioned for **peak load**
- At lower layers, capacity is provisioned to handle any traffic matrix
- Traffic varies
- **Daily** (more email in day than night)
- **Weekly** (More Database queries on week-days)
- **Monthly** (Higher photo sharing on holidays)
- **Yearly** (More shopping in December)

Fat Trees

- Are highly scalable
- Can be designed to support all communica-tion patterns
- Built from large number of richly interconnec-ted switches
- Provide 1:N redundancy
- ElasticTree benefits greatly from Fat Trees

Fat Tree

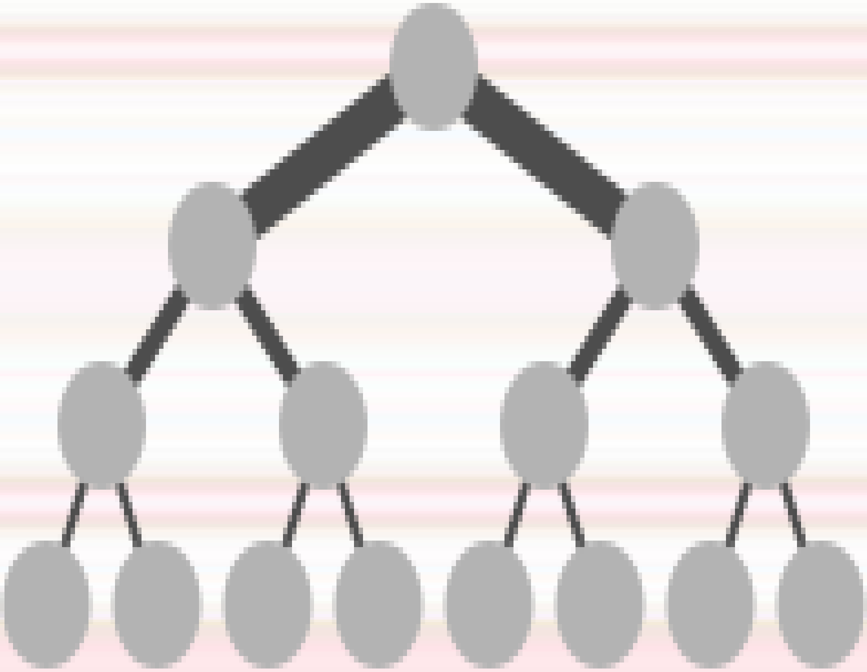


Question??

Why the name “Fat Tree”?

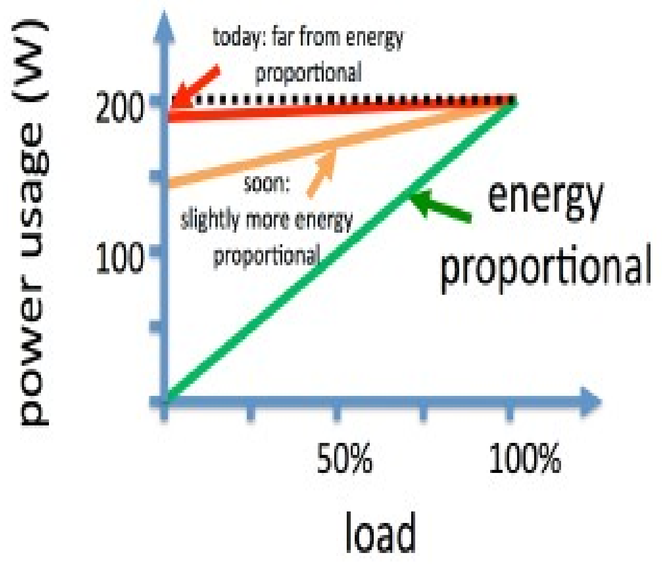
What is FAT??

- The links in a fat-tree become "fatter" as one moves up the tree towards the root.



Power consumption of Switches

Switch Configurations (48 ports)	Model A power in W	Model B power in W	Model C power in W
Idle, no active ports	151	133	76
All ports active, no traffic	184	170	97
All ports active, traffic at 1 Gbps	195	175	102



Workload Management in a Data Center

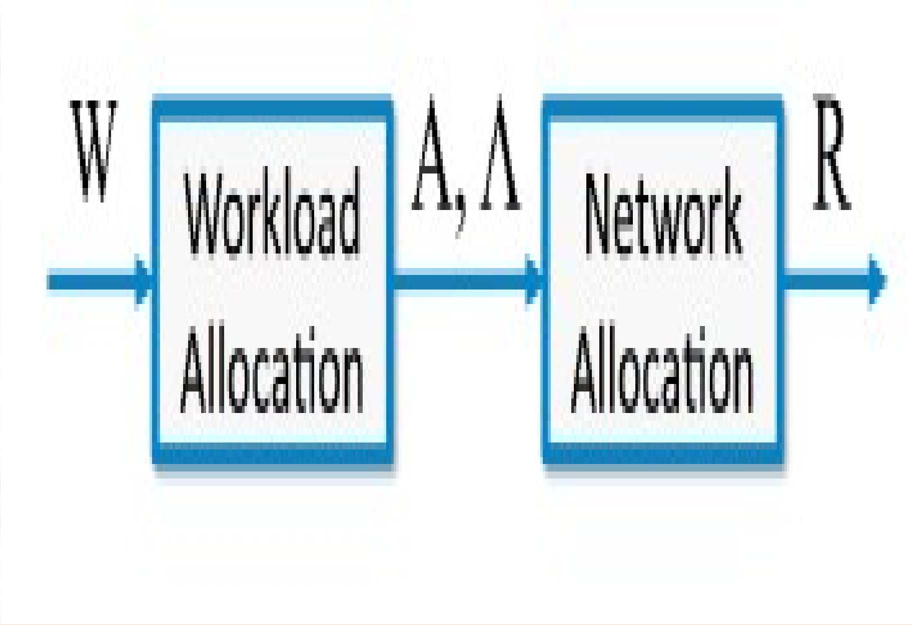
Managing a Data Center

- Performance and cost are at odds with each other
- **Best performance:** By spreading workload to the maximum possible
- **Most energy efficient solution:** Concen-trate all load on minimum possible servers

Quick Question

If performance is not a consideration, what will be the most energy efficient solution for data centers?

Workflow Allocation in Data Center



Done in two steps:

1. Work allocation to servers, to meet some performance criteria
2. Traffic is routed by Network. Current approach is to min imize congestion and maximize fault-tolerance

ElasticTree: A Network Power Optimizer

ElasticTree

Its a dynamic network power optimizer. Uses the following two ways to calculate traffic routing

- Near optimal solution: Uses integer and linear programs
- Heuristic: Fast and scalable, but suboptimal

Near-optimal Solution

- System is modeled as Multi-Commodity network Flow (MCF)
- Objective is to minimize total N/W power
- Usual MCF constraints like
- Link Capacity
- Flow conservation
- Demand satisfaction
- Additional constraints
- Traffic only on powered on switches and links
- No such thing as half-on Ethernet link
- Model does not scale beyond networks of 1000 hosts!

Heuristic Solution

- Exploits regularity of fat trees
- Assumes flows are perfectly divisible
- Using traffic matrix, compute the max traffic between an edge switch and aggregation layer
- Total traffic divided by link capacity gives the min number of aggregation switches needed

Heuristic Solution(Contd)

$$N_i^{agg} = \lceil \max_{s \in E_i}((\sum_{t \in A_i} F(s \to t))/r) \rceil$$

- **N_i^{agg}** is number of switches required in pod i
- **E_i** is set of edge switches in pod i
- **F(s → t)** is rate of flow between 's' and 't'
- **A_i** is set of nodes for which F(s → t) must traverse aggregation layer of pod 'i'
- '**r**' is the link rate

Heuristic Solution(Contd)

$$N^{core} = \lceil \max_{s \in C}((\sum_{t \in B_i} F(s \to t))/r) \rceil$$

- **N^{core}** is number of switches required in core
- **C** is the set of core switches
- **B_i** is set of nodes for which flow F(s → t) must traverse aggregation layer of pod 'i'

Heuristic Solution(Contd)

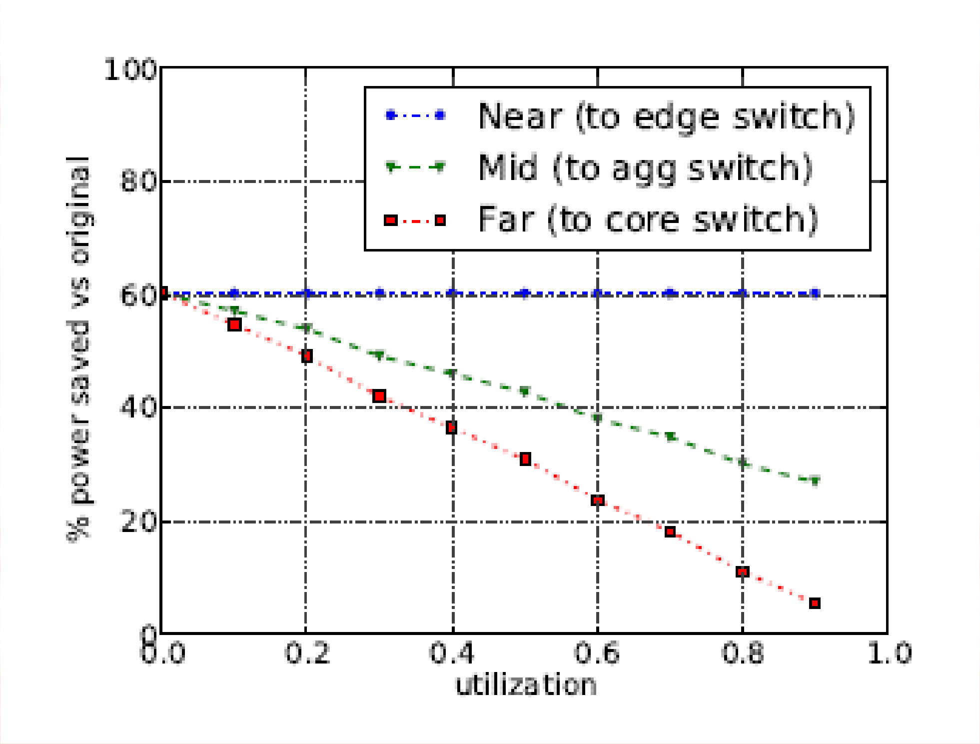
- Heuristics assume 100% link utilization
- K-redundancy by adding k switches to each pod and N^{core}
- Similarly max link utilization can be set to 'r'

Evaluation

Traffic Extremes

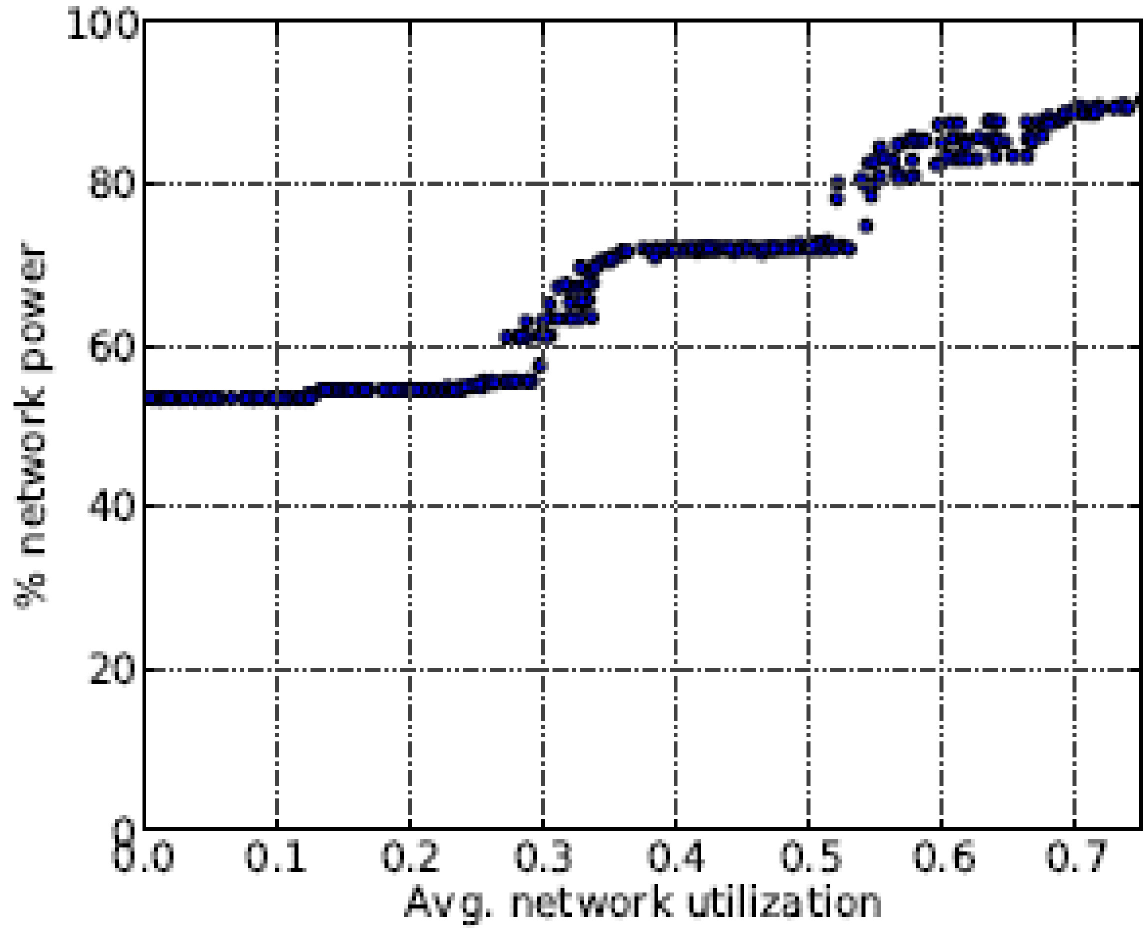
- **Near traffic:** Here servers communicate with other servers only through their edge switch (best-case)
- **Far traffic:** Servers communicate with serv-ers in other pods only (worst-case)
- For “far traffic” savings depend heavily on network utilization

Power Savings vs Locality

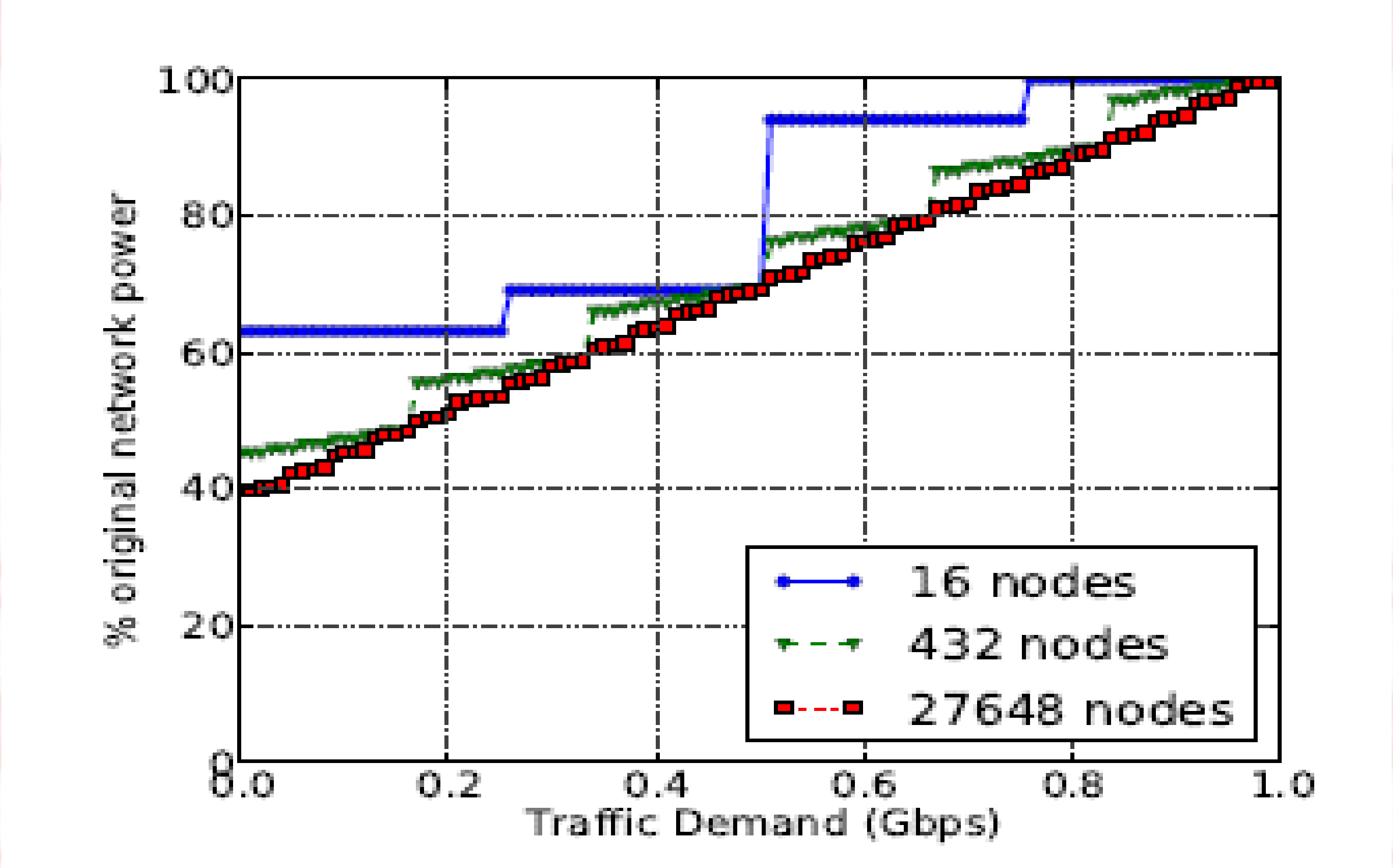


- Increased savings for more local communications
- Savings to be made in all cases!

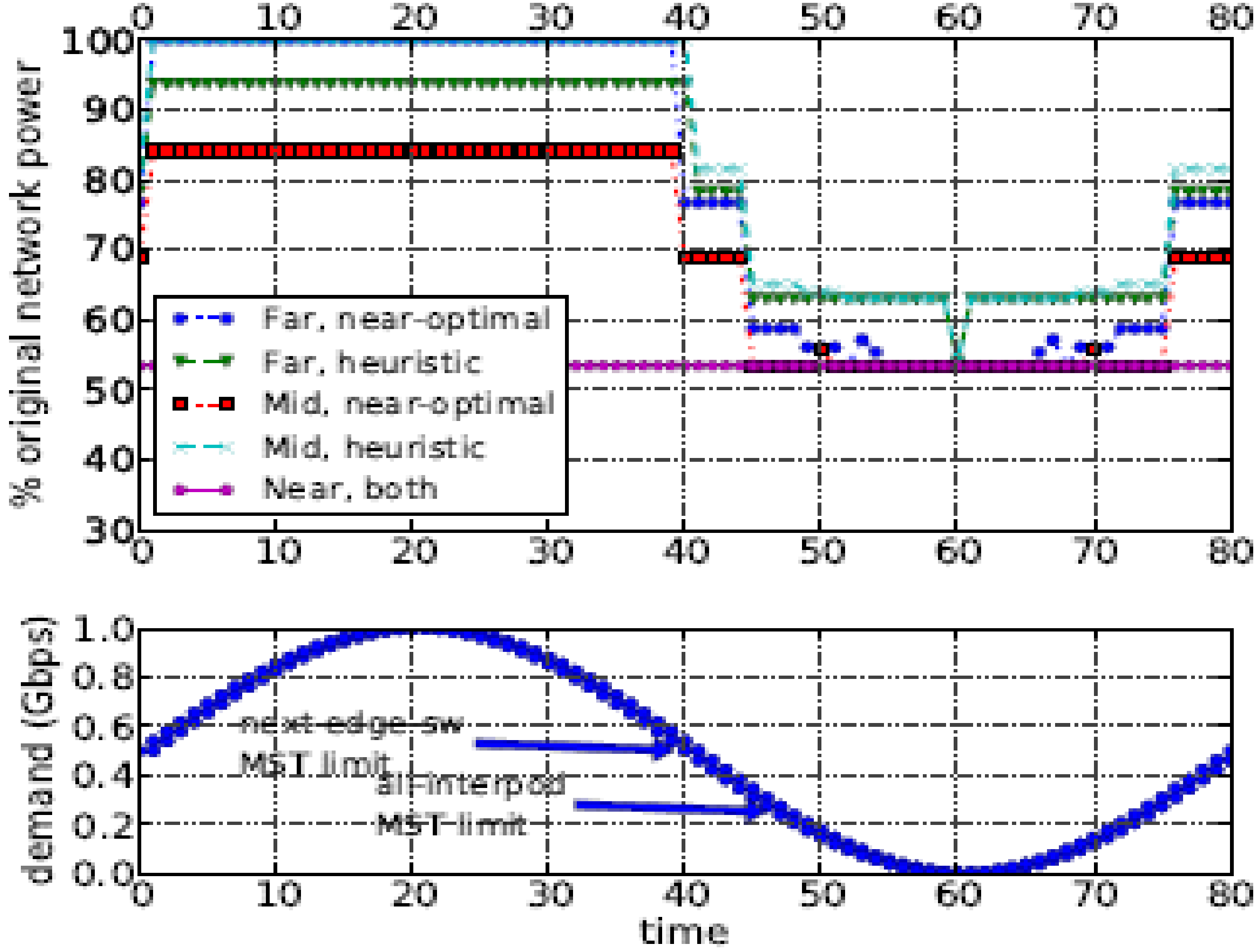
Power savings with Random traffic



Energy savings vs N/W size and demand



Time-varying utilization

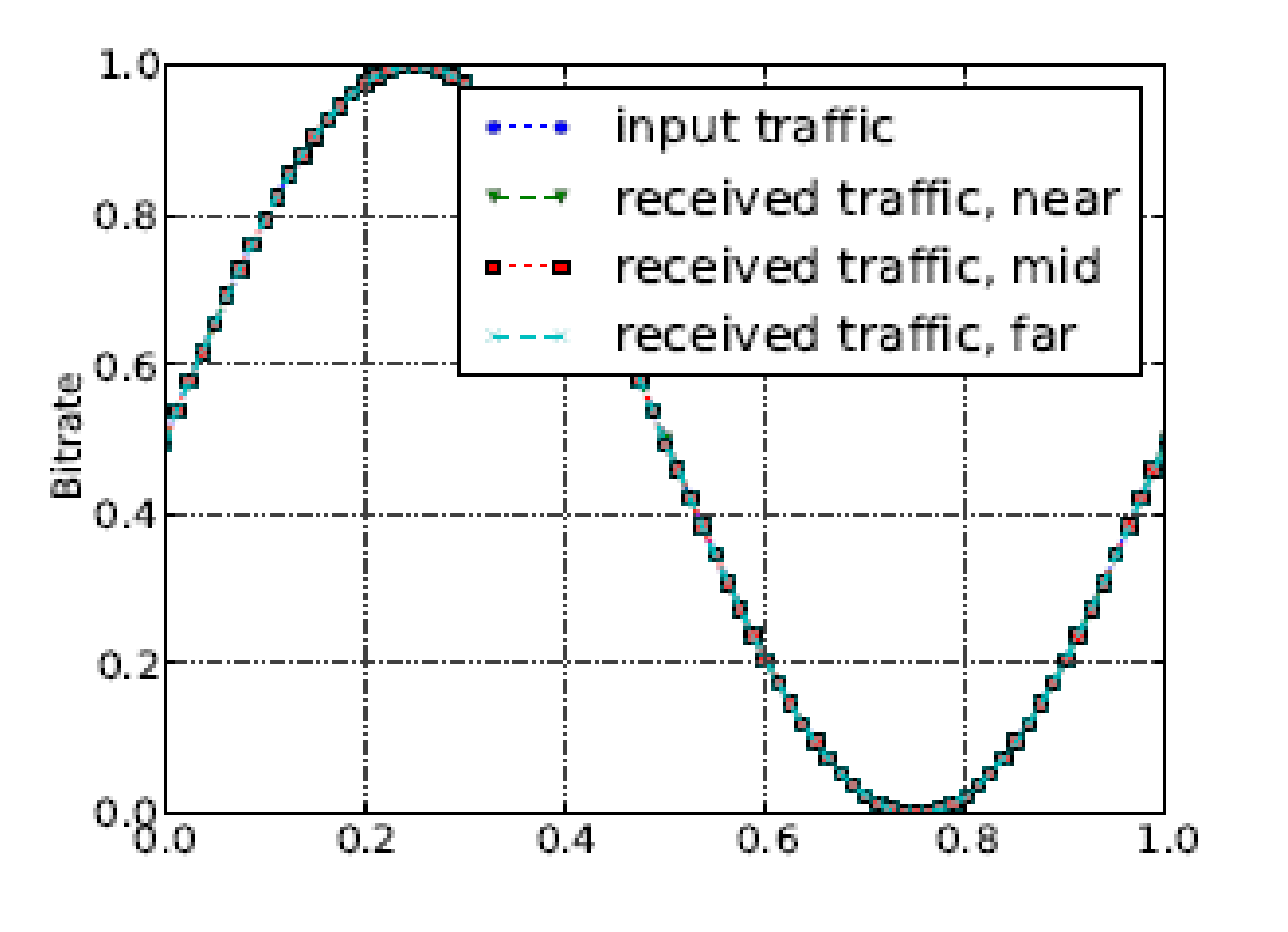


System Validation

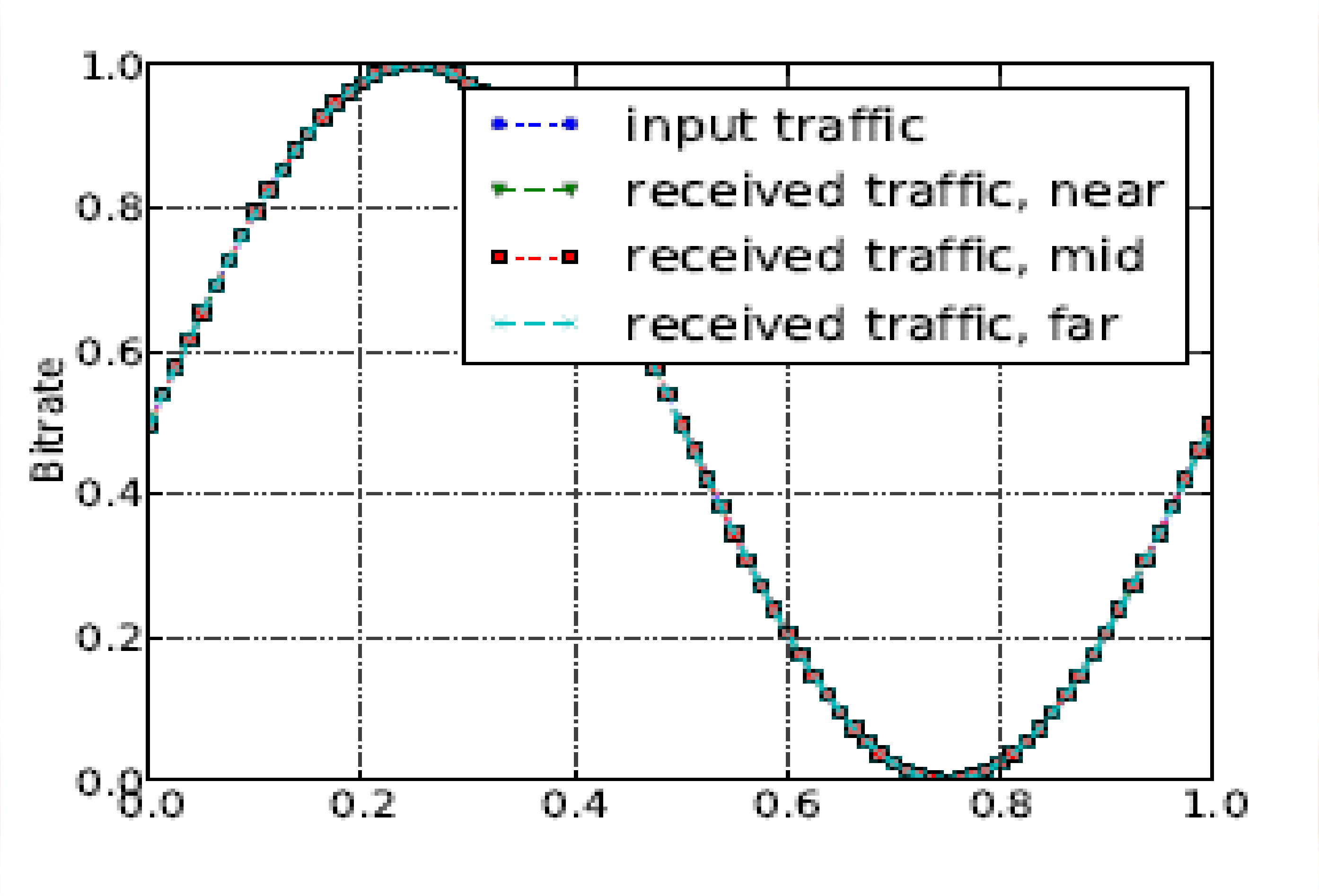
Bandwidth validation

- Both, near optimal and heuristic solution very closely match original traffic
- Packets dropped only when traffic on a link is extremely close to line rate
- Ensuring spare capacity can prevent packet drops

Bandwidth validation, k=4



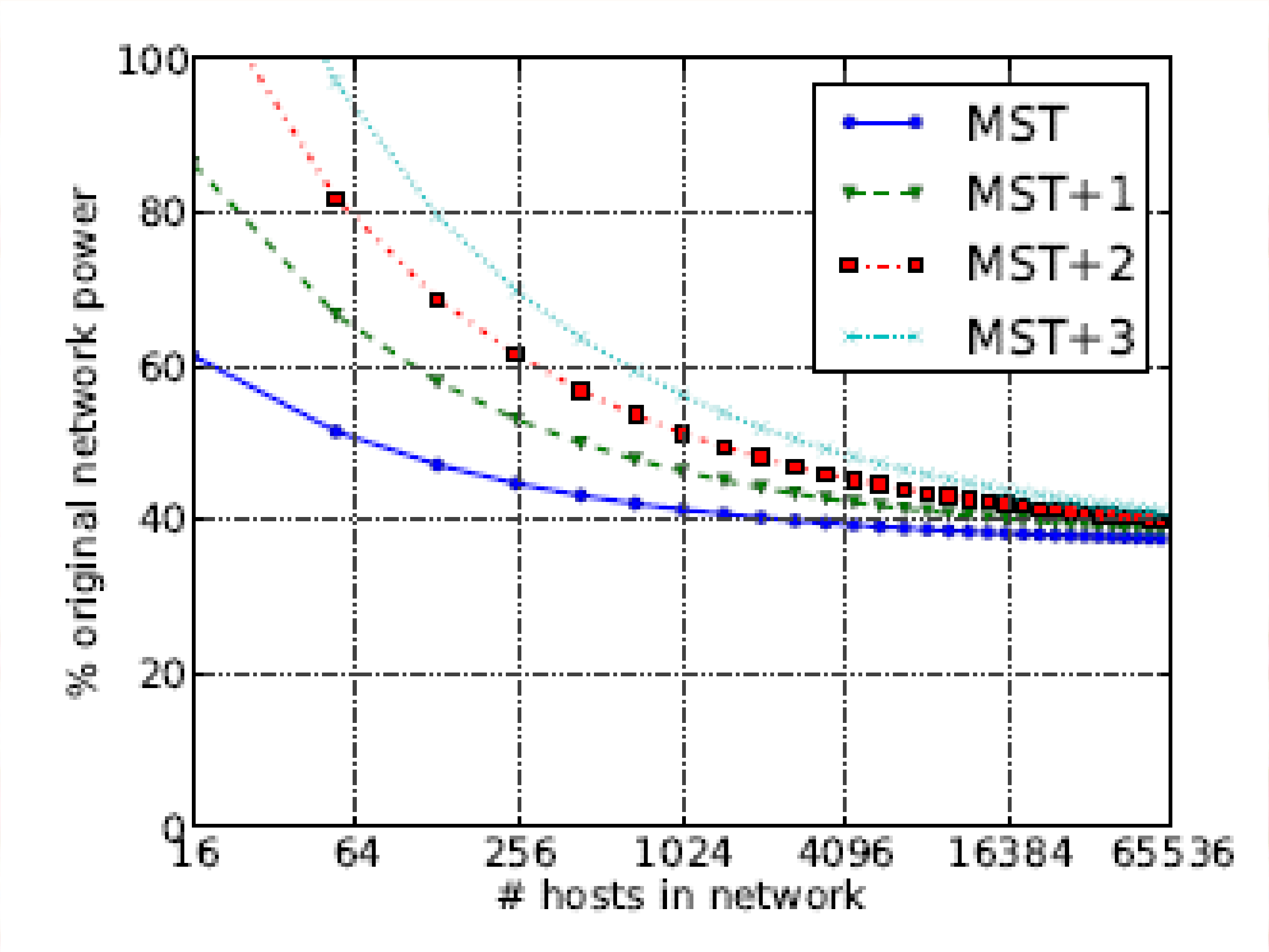
Bandwidth validation, k=6



Fault Tolerance

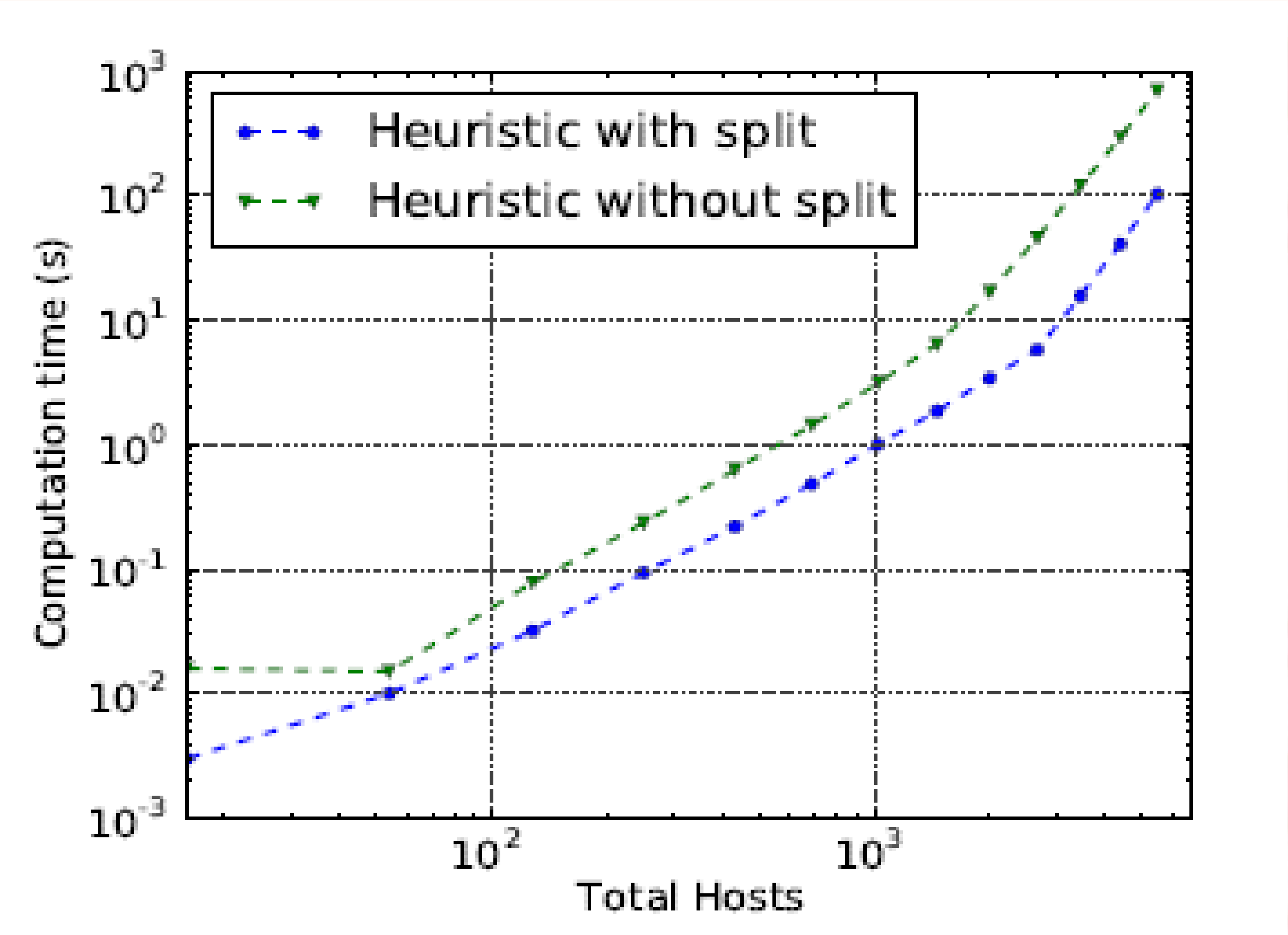
- MST certainly minimizes power but throws away all fault tolerance
- MST+i requires 'i' additional switches per pod and in the core
- With increase in N/W size, incremental cost of fault tolerance becomes insignificant

Power cost of redundancy



Scalability

Computation Time



Conclusion

- About 60% of network energy can be saved
- If workload can be moved quickly and easily, then the data center can be re-optimized fre-quently

Thank you