Towards Predictable + Resilient Multi-Tenant Data Centers

Presenter: Ali Musa Iftikhar (Tufts University)

in joint collaboration with:

Fahad Dogar (Tufts), {Ihsan Qazi, Zartash Uzmi, Saad Ismail, Gohar Irfan} (LUMS),

Mohsin Ali (USC), Ruwaifa Anwar (SUNY SB)

Multi-Tenant Data Centers

- Flexible pay as you go model is attractive to tenants
- Meets variability in tenant demands

Multi-Tenant Data Centers

- Flexible pay as you go model is attractive to tenants
- Meets variability in tenant demands
- Yet, there are challenges to deal with

Why is Predictability Important?

- Data center is a shared resource
- Leads to high variability in the network
    - Potentially results in tenant's cost variability

Why is Predictability Important?

- Data center is a shared resource
- Leads to high variability in the network
    - Potentially results in tenant's cost variability

Thus we need to provide some sort of **Predictability**

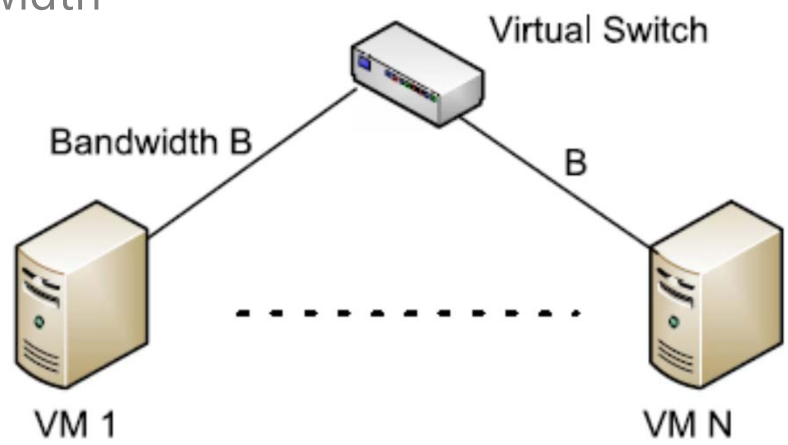
Virtual Abstractions for Predictable Performance

Virtual abstractions:

- Expose a virtual network to the tenants
- Tenants can then demand for guaranteed bandwidth

Examples of such abstractions include:

{Oktopus, FairCloud, CloudMirror} (Sigcomm '11 '12 '14), Hadrian (NSDI '13)



Virtual Abstractions for Predictable Performance

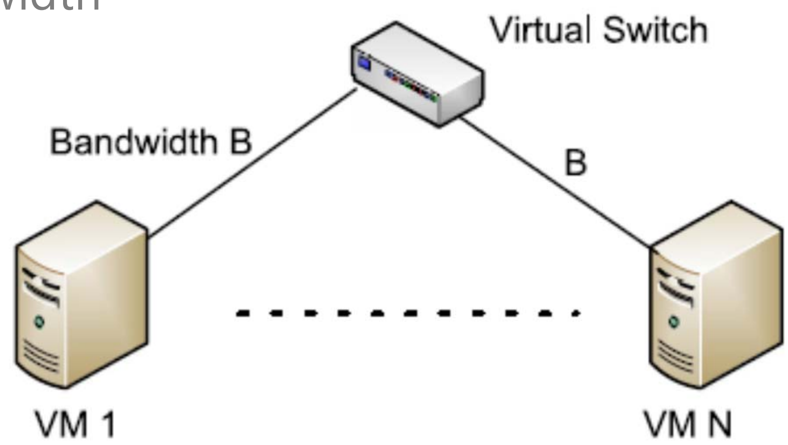
Virtual abstractions:

- Expose a virtual network to the tenants
- Tenants can then demand for guaranteed bandwidth

Examples of such abstractions include:

{Oktopus, FairCloud, CloudMirror} (Sigcomm '11 '12 '14), Hadrian (NSDI '13)

But, they tend to ignore a crucial factor!



A stark Reality – Failures!

Datacenter **Network Failures** are common:

- Studies have shown: *(Understanding network failures in data centers, Sigcomm '11)*
  - **30%** of the components show **less than** four 9s of availability
  - Time between successive failures could be as short as **5 minutes**
  - Time for recovery could even go beyond **1 week**
- These failures result in significant service downtimes hurting the tenants!

A stark Reality – Failures!

Datacenter **Network Failures** are common:

- Studies have shown: *(Understanding network failures in data centers, Sigcomm '11)*
  - **30%** of the components show **less than** four 9s of availability
  - Time between successive failures could be as short as **5 minutes**
  - Time for recovery could even go beyond **1 week**
- These failures result in significant service downtimes hurting the tenants!

Thus we need to provide **Reliability** + **Predictability**

"Predictability + Resilience": Requirements

Goal	Requirement(s)
Predictability	**Bandwidth Reservation**

“Predictability + Resilience”: Requirements

Goal	Requirement(s)
Predictability	**Bandwidth Reservation**
Resilience	• **Firstly: Provide Backup Resources to enable recovery** • **Secondly: Ensure speedy recovery** *(Aspen Trees CoNEXT ’13, F10 NSDI ’13)*

"Predictability + Resilience": Requirements

Goal	Requirement(s)
Predictability	**Bandwidth Reservation**
Resilience	Focus of this talk • **Firstly: Provide Backup Resources to enable recovery** • **Secondly: Ensure speedy recovery** *(Aspen Trees CoNEXT '13, F10 NSDI '13)*

Providing Backup Resources for Resilience

One approach:

- **Reserve Backup Bandwidth** to tolerate failures along with tenant reservations

We simulate this approach on a typical fat-tree topology to test our hypothesis.

Reserving Backup Bandwidth on Fat-Tree: Simulation

Simulation details:

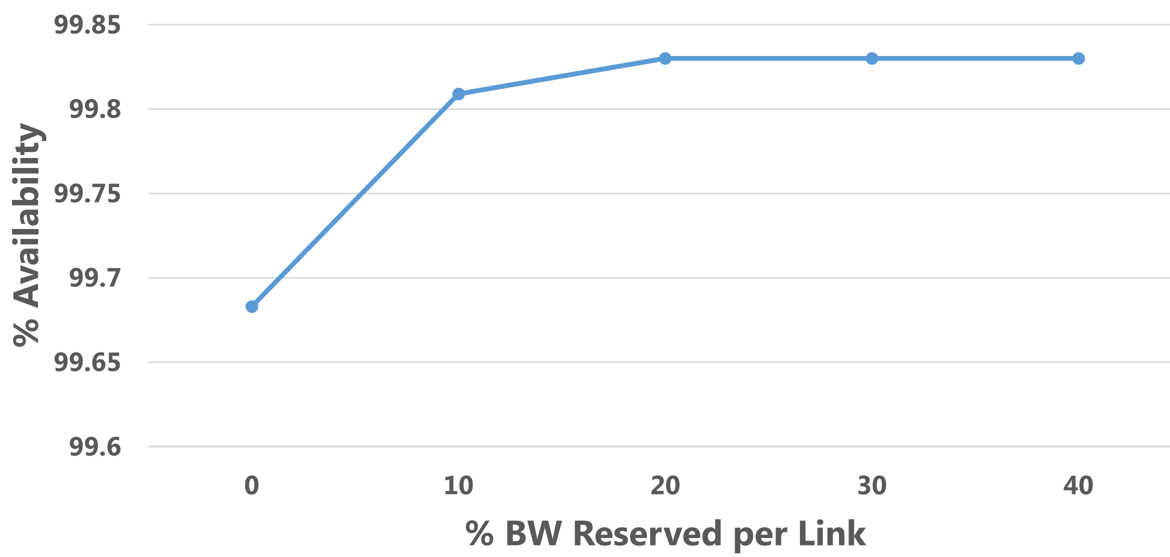
- 48-ary fat-tree: *A Scalable, Commodity Data Center Network Architecture (Sigcomm '08)*
- Induce failure model: *Understanding network failures in data centers (Sigcomm '11)*
- Virtual cluster abstraction: *Oktopus (Sigcomm '11)*
- Metric:

$$\textbf{Percentage Availability} = \frac{\text{Total uptime experienced by tenants}}{\text{Total duration}} \times 100\%$$

Reserving Backup Bandwidth on Fat-Tree: Simulation

Simulation details:

- 48-ary fat-tree: *A Scalable, Commodity Dc*
- Induce failure model: *Understanding net*
- Virtual cluster abstraction: *Oktopus (Sig*
- Metric:

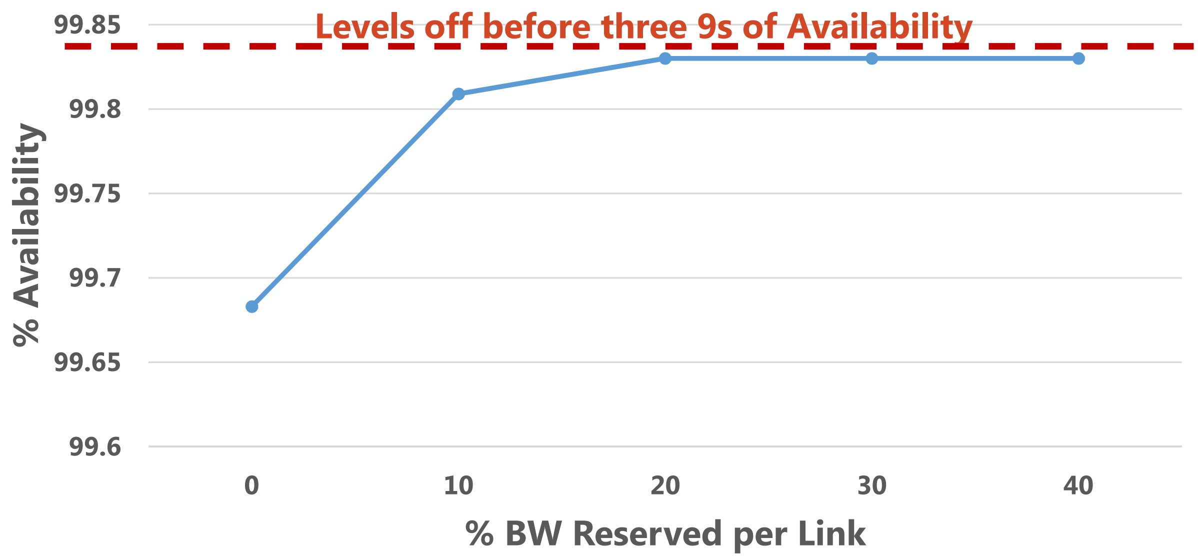


$$\textbf{Percentage Availability} = \frac{\text{Total uptime experienced by tenants}}{\text{Total duration}} \text{ x } 100\%$$

Reserving Backup Bandwidth on Fat-Tree: Simulation

Simulation details:

- 48-ary fat-tree: *A Scalable, Commodity Dc*
- Induce failure model: *Understanding net*
- Virtual cluster abstraction: *Oktopus (Sig*
- Metric:

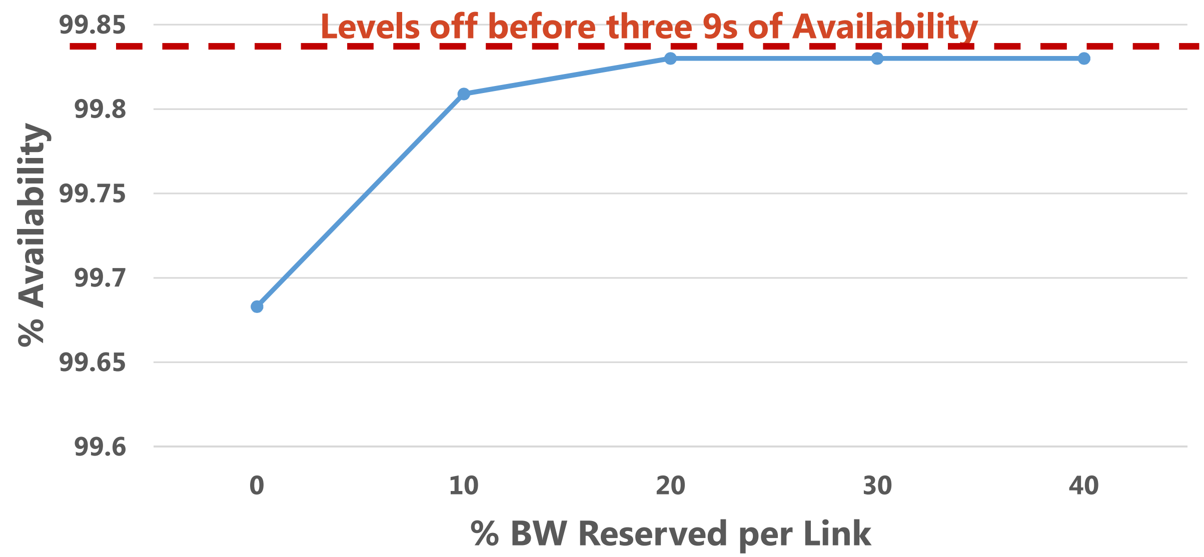


$$\textbf{Percentage Availability} = \frac{\text{Total uptime experienced by tenants}}{\text{Total duration}} \times 100\%$$

Reserving Backup Bandwidth on Fat-Tree: Simulation

Simulation details:

- 48-ary fat-tree: *A Scalable, Commodity Da*
- Induce failure model: *Understanding net*
- Virtual cluster abstraction: *Oktopus (Sig*
- Metric:



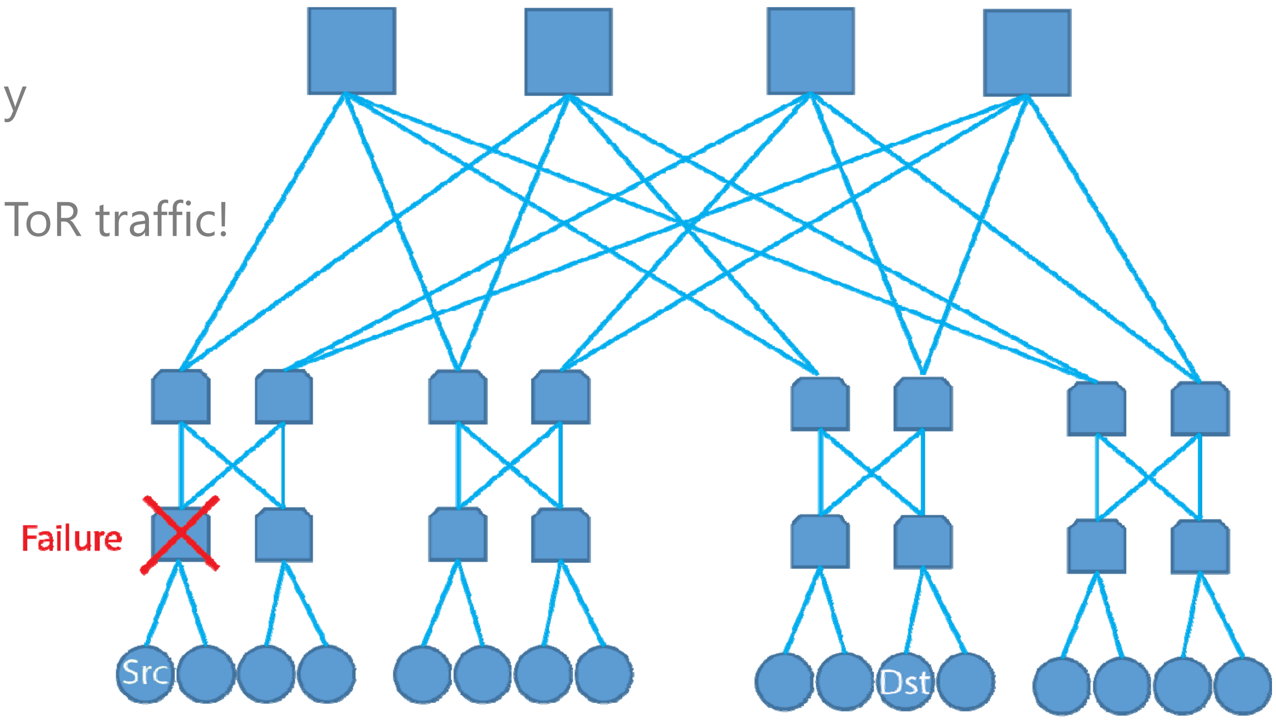
$$\textbf{Percentage Availability} = \frac{\text{Total uptime experienced by tenants}}{\text{Total duration}} \times 100\%$$

So what did we overlook?

Single Point of Failure – ToRs

Inherent to the fat-tree topology

- No alternate path to reroute ToR traffic!



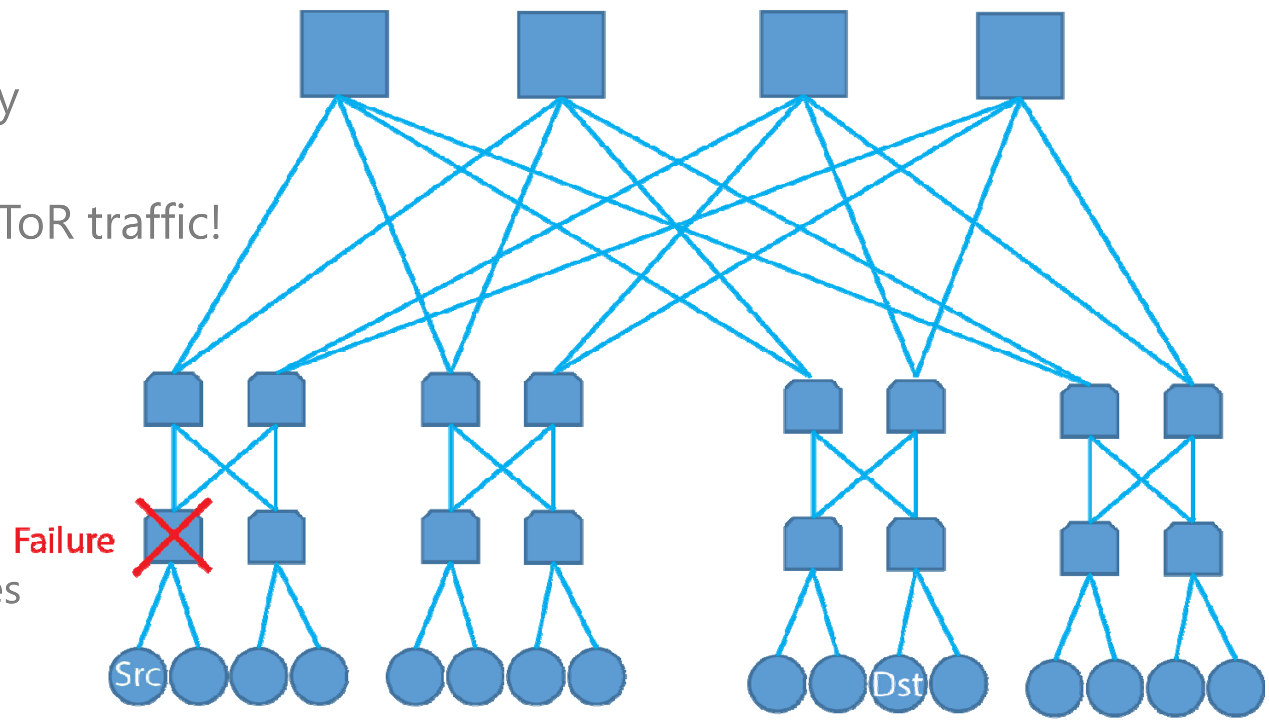
Single Point of Failure – ToRs

Inherent to the fat-tree topology

- No alternate path to reroute ToR traffic!

Potential solutions:

- VM migration
  - Has its own set of challenges
- Modify topology



Fat-Resilient-Trees: High Level Idea

Key idea: Multi-home the end hosts

Fat-Resilient-Trees: High Level Idea

Key idea: Multi-home the end hosts

Goals we target:

- Must have the same cost as its fat-tree counterpart

  Which requires having the same number and size of switches

Fat-Resilient-Trees: High Level Idea

Key idea: Multi-home the end hosts

Goals we target:

- Must have the same cost as its fat-tree counterpart

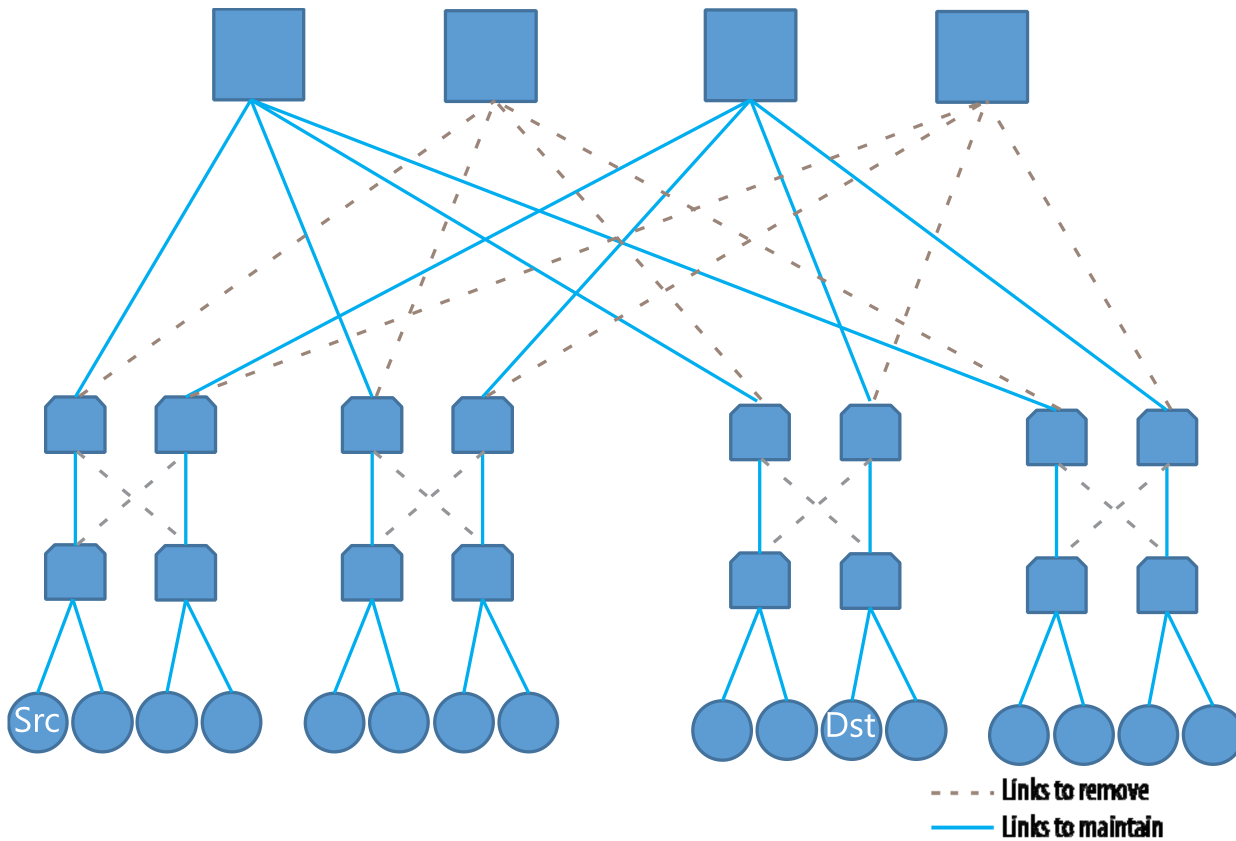
    Which requires having the same number and size of switches

So we simply **Rearrange** the existing redundancy

- Introducing redundancy at ToR level by stripping it form overly redundant levels.

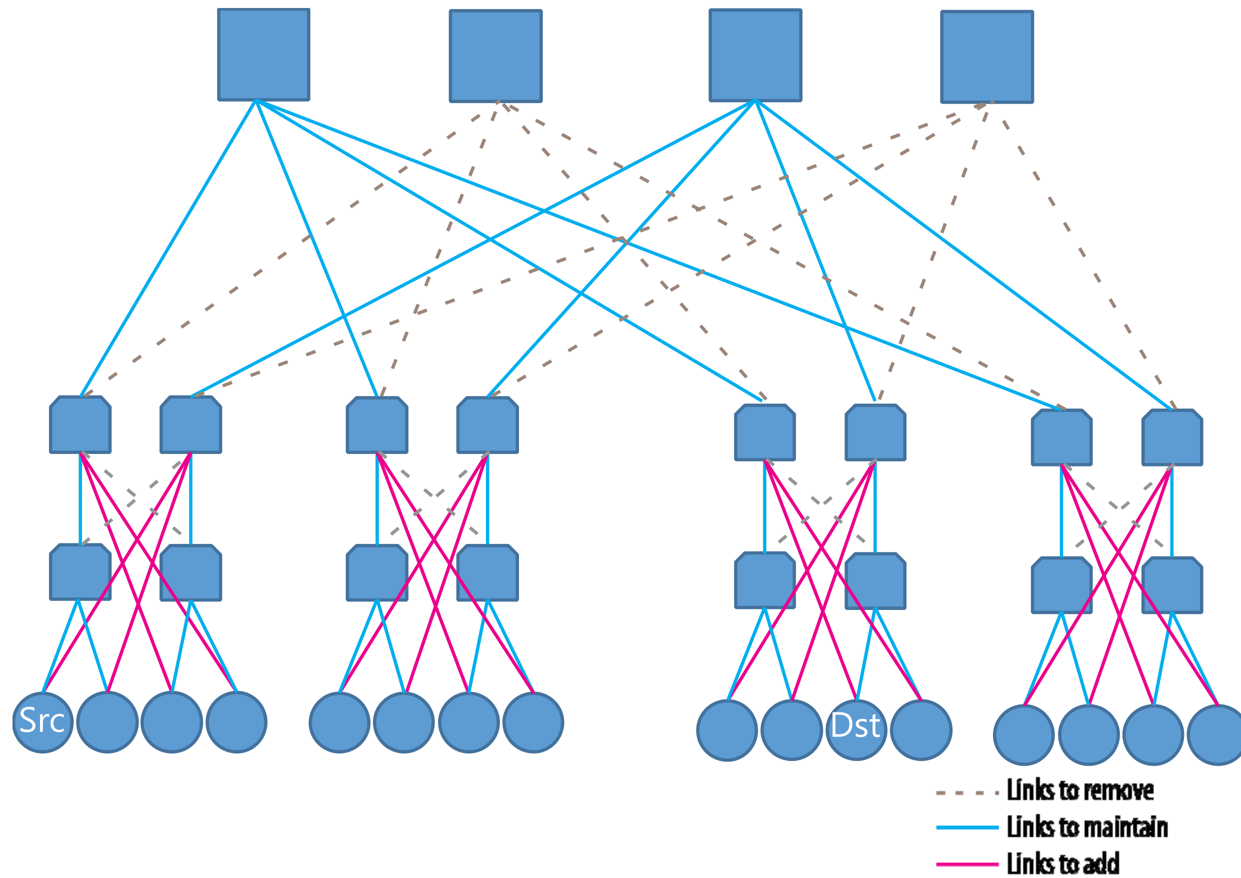
Fat-Resilient-Trees: High Level Idea

- Uniformly remove the overly redundant links



Fat-Resilient-Trees: High Level Idea

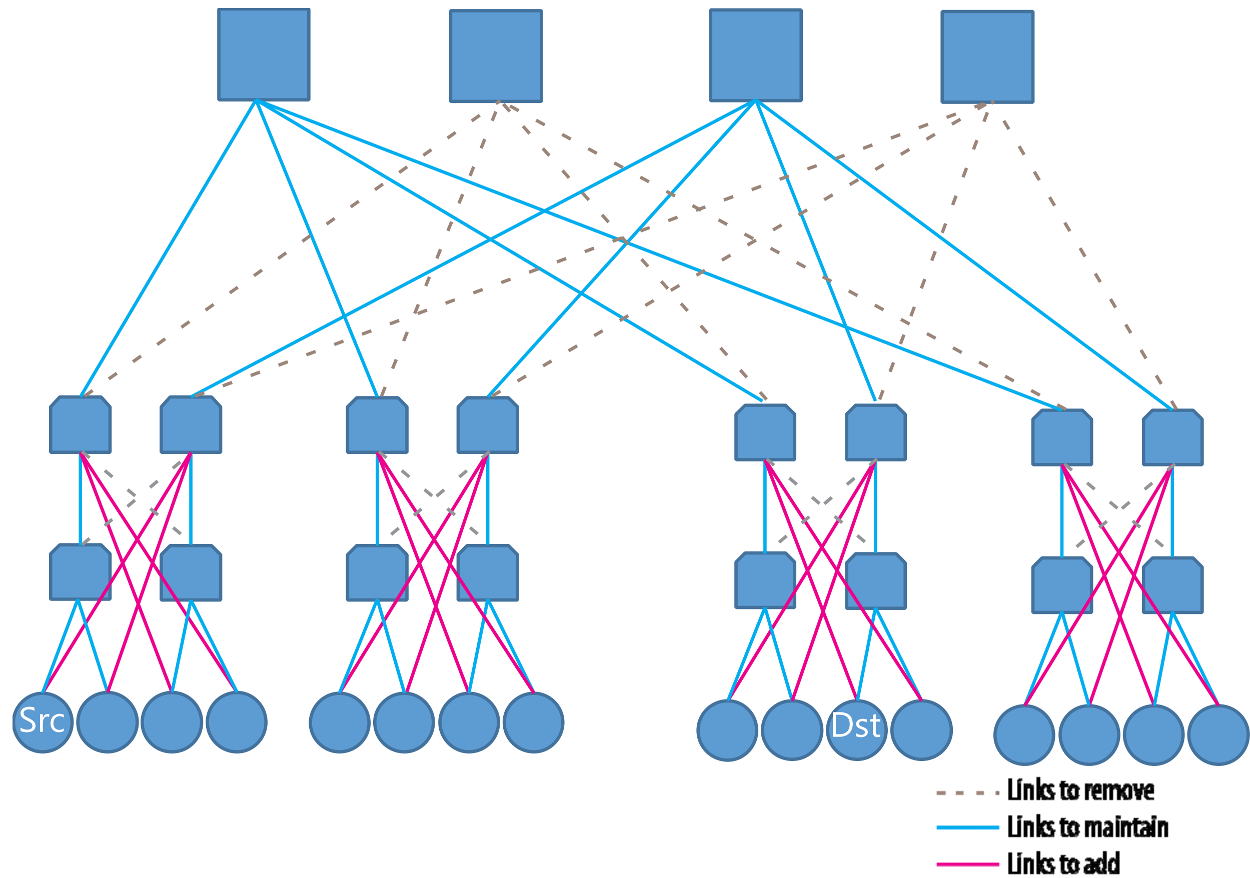
- Uniformly remove the overly redundant links
- Reconnect them in a way which ensures that every end-host is connected to every other end-host



Fat-Resilient-Trees: High Level Idea

Works because of Locality in Traffic:

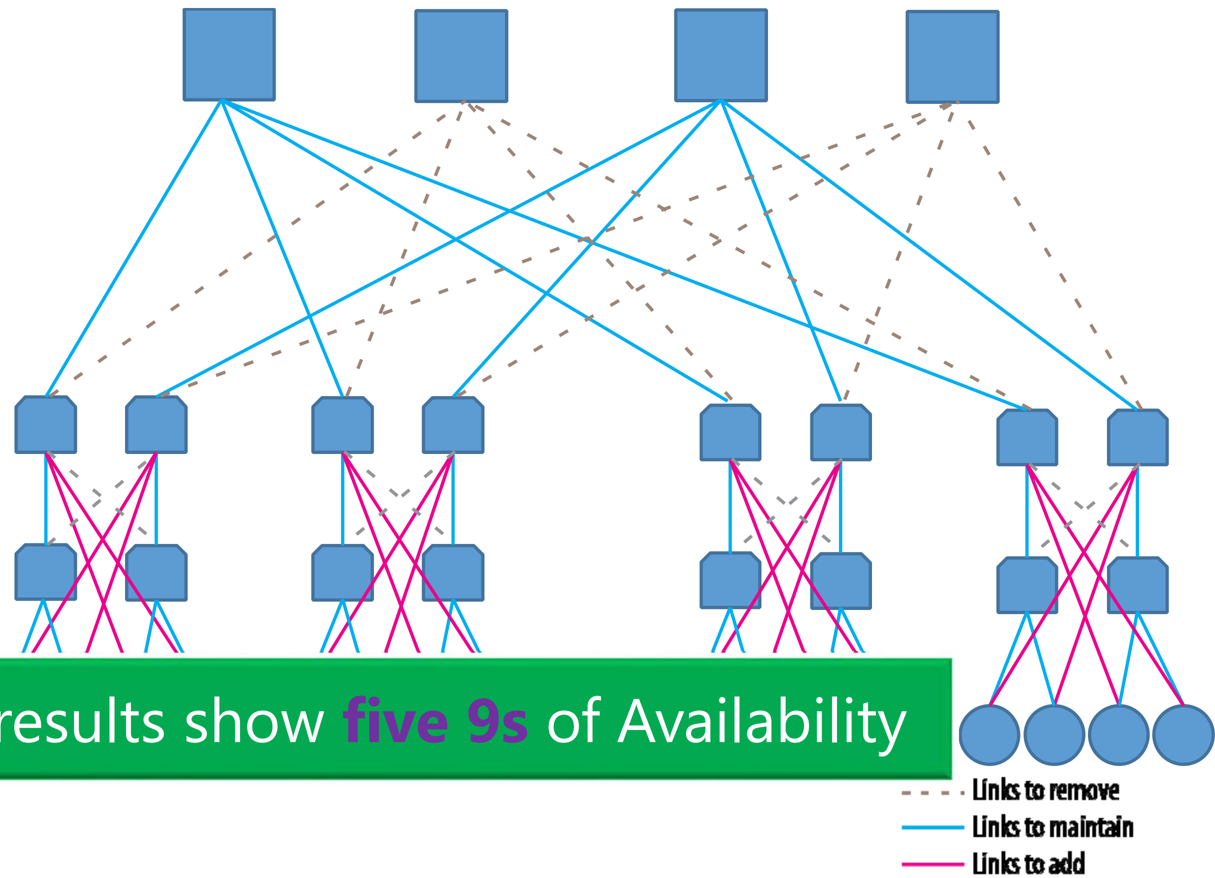
- Collocation motivates that full bisection BW is perhaps at every level an overkill



Fat-Resilient-Trees: High Level Idea

Works because of Locality in Traffic:

- Collocation motivates that full bisection BW is perhaps at every level an overkill



Preliminary simulation results show five 9s of Availability

Ongoing Work

- Understand and evaluate the implications Fat-Resilient-Trees
- Extensively compare against existing topologies
- Build a fast recovery mechanism

Questions & Feedback?

Thank you for your time ☺

References

[1] H. Ballani, P. Costa, T. Karagiannis, and A. Rowstron, "Towards predictable datacenter networks," in *ACM SIGCOMM 2011*.

[2] H. Ballani, K. Jang, T. Karagiannis, C. Kim, D. Gunawardena, and G. O'Shea, "Chatty tenants and the cloud network sharing problem," in *NSDI 2013*.

[3] L. Popa, G. Kumar, M. Chowdhury, A. Krishnamurthy, S. Ratnasamy, and I. Stoica, "Faircloud: Sharing the network in cloud computing," in *ACM SIGCOMM 2012*.

[4] P. Gill, N. Jain, and N. Nagappan, "Understanding network failures in data centers: Measurement, analysis, and implications," in *ACM SIGCOMM 2011*.

[5] C. Guo, G. Lu, H. J. Wang, S. Yang, C. Kong, P. Sun, W. Wu, and Y. Zhang, "Secondnet: A data center network virtualization architecture with bandwidth guarantees," in *ACM Co-NEXT 2011*.

[6] L. Popa, P. Yalagandula, S. Banerjee, J. C. Mogul, Y. Turner, and J. R. Santos, "Elasticswitch: Practical work-conserving bandwidth guarantees for cloud computing," in *ACM SIGCOMM 2013*.

[7] V. Jeyakumar, M. Alizadeh, D. Mazieres, B. Prabhakar, C. Kim, and A. Greenberg, "Eyeq: Practical network performance isolation at the edge," in *NSDI 2013*.

[8] H. Rodrigues, J. R. Santos, Y. Turner, P. Soares, and D. Guedes, "Gatekeeper: Supporting bandwidth guarantees for multi-tenant datacenter networks," in *WIOV 2011*.

[9] A. Shieh, S. Kandula, A. Greenberg, and C. Kim, "Sharing the datacenter network," in *NSDI 2011*.

[10] H. Ballani, P. Costa, T. Karagiannis, and A. Rowstron, "The price is right: Towards location-independent costs in datacenters," in *ACM HotNets-X 2011.*

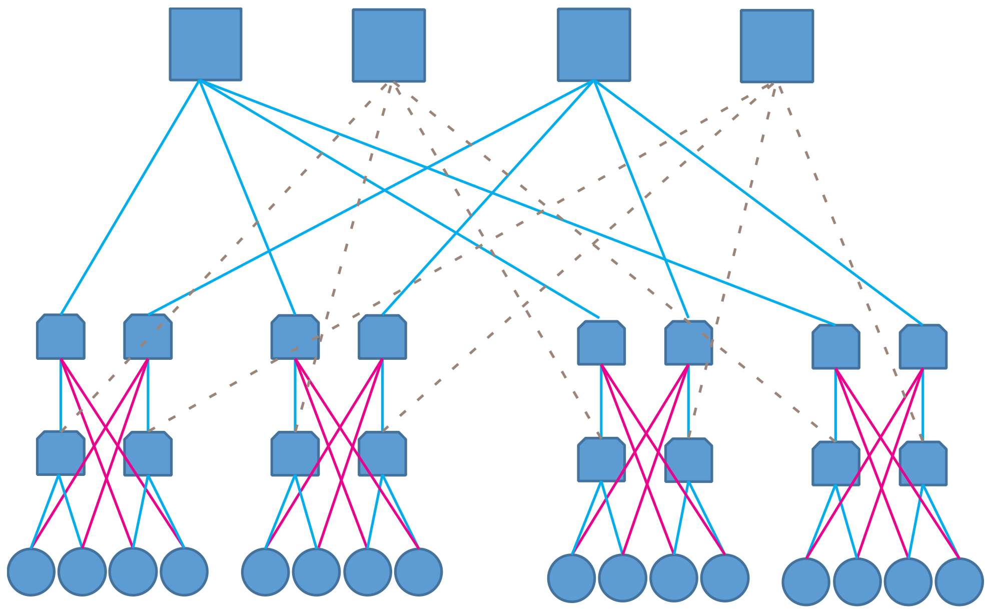
[11] D. Li, C. Guo, H. Wu, K. Tan, Y. Zhang, S. Lu, and J. Wu, "Scalable and cost-effective interconnection of data-center servers using dual server ports," *IEEE/ACM Trans. Networking*, vol. 19, no. 1, pp. 102–114, 2011.

[12] M. Al-Fares, A. Loukissas, and A. Vahdat, "A scalable, commodity data center network architecture," in *ACM SIGCOMM 2008*.

[13] V. Liu, D. Halperin, A. Krishnamurthy, and T. Anderson, "F10: A fault-tolerant engineered network," in *NSDI 2013*.

[14] W. Sullivan, A. Vahdat, and K. Marzullo, "Aspen trees: Balancing data center fault tolerance, scalability and cost," in *CoNext 2013*.

[15] P. Bod´ık, I. Menache, M. Chowdhury, P. Mani, D. A. Maltz, and I. Stoica, "Surviving failures in bandwidth-constrained datacenters," in *ACM SIGCOMM 2012.*



Backup Slides

VM Migration

	avail	efficiency
oktopus + nothing	99.683	1
oktopus + t2t backup	99.809	0.8225308642
oktopus+ t2t + 2 backups	99.83	0.7685185185
oktopus + e2e + 1 backup	99.9998	0.4907407407
oktopus + e2e + 2 backups	99.99999	0.3364197531
oktopus + sharing + 1 pod	99.9998	0.9768518519
oktopus + sharing + 5 pods	99.99999	0.8796296296
oktopus + new topology + backups	99.9997	0.8641975309