



From Fat Tree to Jellyfish

Rewiring India's Networks for the AI Era

Lalit S Chowdhary

CTIO, Lightstorm

#apnic62

MUMBAI, INDIA
4 - 10 September 2026



From Fat Tree to Jellyfish

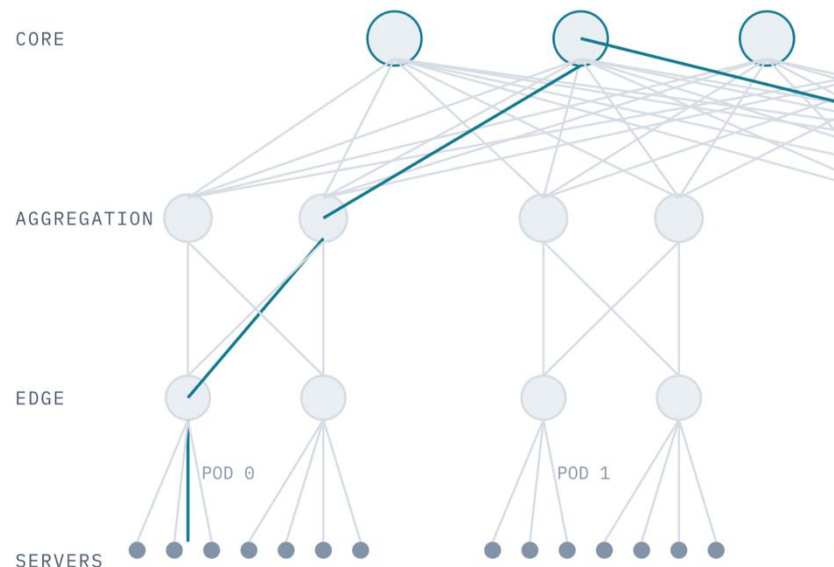
- Introduction to Fat Tree Data Center Network
- How Fat Tree Fits in Data Center Interconnect as Spine-Leaf Topology
- Pros-Cons of Fat Tree
- Introduction to Jelly Fish Topology
- Lightstorm's Innovation of converting Spine-Leaf Topology to Jelly Fish by introducing CDC-F ROADMs
- Future Proof Network for AI Demand

What is a Fat Tree Network

Fat Tree topology was invented by Professor Charles Leiserson, MIT in 1985, as a specialized network design with wider bandwidth links near the root to prevent bottlenecks. Modern data centers and high-performance clusters use it for equal upload and download speeds across all connected servers. There are three fundamental questions we need to ask:

1. How much bandwidth stays within the Data Center (Scale Up and Scale Out)? How much bandwidth is required to connect between two Data Centers if this architecture was to Scale Across?
2. How much latency can Scale Across Networks handle?
3. How much over subscription is required to meet Five Nine availability.

All three come down to how many equal-cost paths exist, and how long a bit takes to cross them?



A $k=4$ fat tree. Every edge switch reaches every core switch through some aggregation switch, so many equal-cost paths connect any two servers — ECMP spreads flows across them.

Bandwidth Per Rack

Assuming n-servers each having 25-100G NIC for hyper converged private cloud:

1. Scale Up network bandwidth is sum of all the links 1:1
2. Scale Out network bandwidth is 3x with over-subscription
3. Scale Across network bandwidth is 4:1 contention ratio

8 Servers in rack can generate 2.4Tbps Scale. The scale across bandwidth is inversely proportional to latency. Higher the latency; the longer it takes to commit every transaction.

1. One millisecond round trip delay can tolerate 1:4 contention ratio (200Gbps)
2. 25 millisecond RTD the contention ratio drops to 1:20 (40 Gbps)

Web / general

10-25 Gbps · 3-4:1 oversub

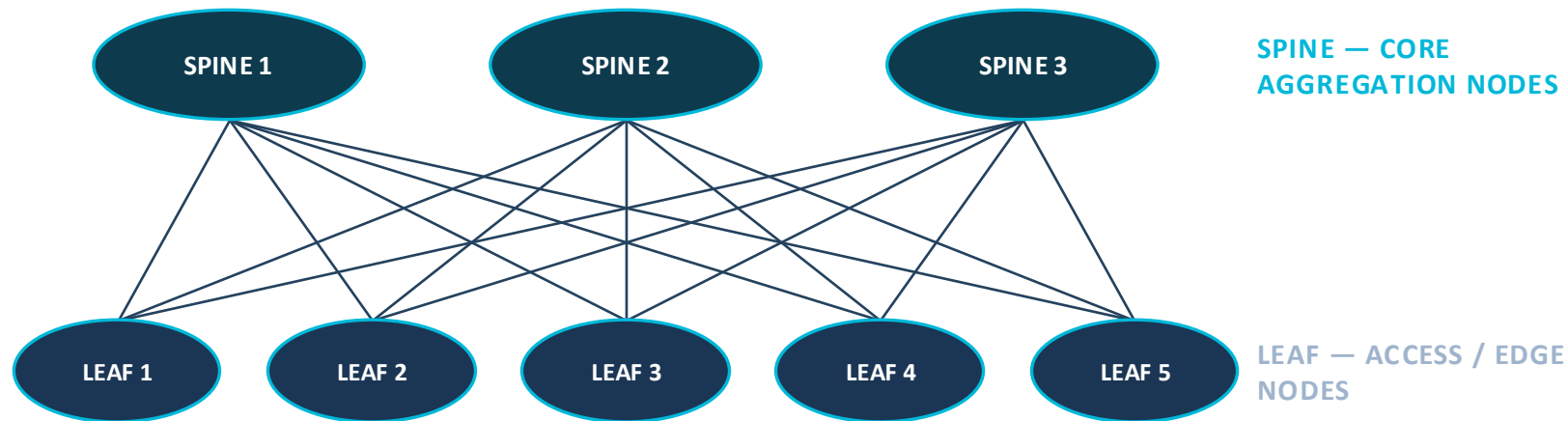
Distributed
storage

25 Gbps · ~1:1

Virtualized /
cloud

25-100 Gbps · 2:1

How to build a Shared Spine-Leaf Metro Data Center Interconnect Network



- Every leaf connects to every spine upto a 24 Tbps Link — full mesh fabric capacity is 72 Tbps available as 10Gbps or 100Gbps or 400Gbps wavelength services
- Any leaf-to-leaf path is a constant two hops, regardless of scale. Each Wavelength can be routed independently across any two nodes. Wavelength services are provisioned using Robotic Arm.
- Grow by adding leaf nodes (edge reach) independently without disturbing the existing traffic.

How Spine Leaf Architecture provides superior network uptime over traditional mesh network

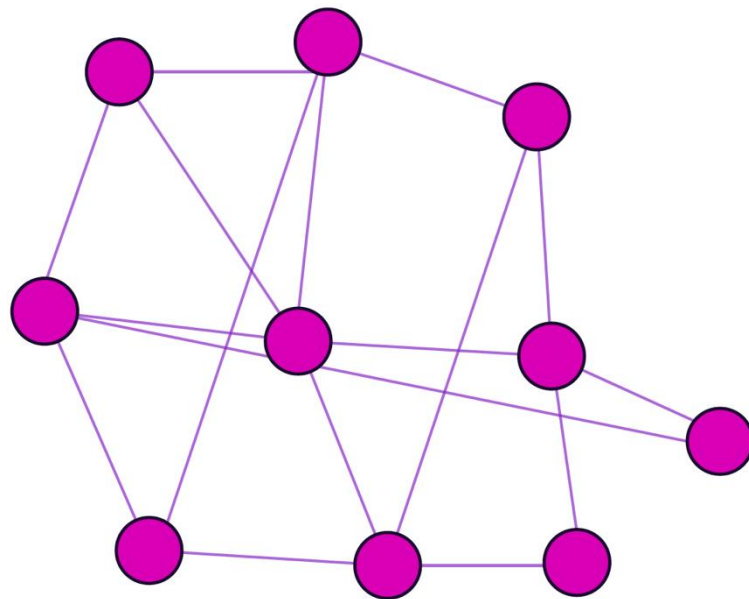
- There two important lessons to learn when scaling the Intra Data Center Network to an Inter Data Center Network
 - Use the Fat Tree Topology Over Subscription can be used to improve network availability in the spine leaf metro topology
 - 1:1 Subscription provides 98% Availability (assuming 4 fiber cuts per link or 14 hours of outage per month)
 - 1:2 Over Subscription provides 99.96% Availability (17 minutes outage per month)
 - 1:3 Over Subscription provides 99.999% Availability (26 seconds outage per month)
 - Distribute the traffic across different nodes (switches) to rule out single point of node switch failures
- Spine Leaf architecture not only provides for superior uptime, but it also always ensures that round trip delay remains within bounds even during multiple fiber cut scenarios.

Pros and Cons of Spine Leaf Topology

- Full bisection non-blocking bandwidth — by design
- Predictable, equal-cost multi-pathing
- Easy to extend Intra-DC traffic to Inter-DC Traffic
- Cost of Feeds and Speeds make it cost prohibitive for enterprises: 10G ⇔ 40G ⇔ 100G ⇔ 400G
- Sharding (distribute data) and Fanout (distribute compute) across DC exasperates this problem

Introducing Jelly Fish topology

- The Jellyfish datacenter topology was invented by a team of computer science researchers from the University of Illinois at Urbana-Champaign (UIUC) and HP Labs.
- Many Data Centers have moved to Jelly Fish topology which provides benefits over Fat Tree:
 - Random Topology, no tiers, no hierarchy, just connectivity
 - Benefits: 25% More Bandwidth, 25% More Servers and Shorter Mean Path
- Lightstorm Converted all its point-to-point Metro Spine Leaf Topology into Jelly Fish topology simply by introducing CDC-ROADM based Wavelength Switching.



*Random regular graph — switches wired at random.
No tiers. No hierarchy. Just connectivity.*

The optical enabler: CDC-F ROADM

Random topologies need a programmable photonic layer

- **Provisioning:** New wavelengths in minutes, not months, pooled transponders pre-connected at every node.
- **1:N optical protection:** Restore around any failure without dedicating 3:1 Oversubscription of capacity
- **Mesh Restoration:** Path Computation Engine reroutes around dual failures using the optical reachability table.
- **Defragmentation:** Re-optimize the spectrum periodically — recover stranded capacity automatically.



Colorless

Any wavelength on any add/drop port — no manual repatching.



Directionless

Any direction at any node — re-route in software, not in trucks.



Contentionless

No wavelength conflicts on the drop side — full flexibility.



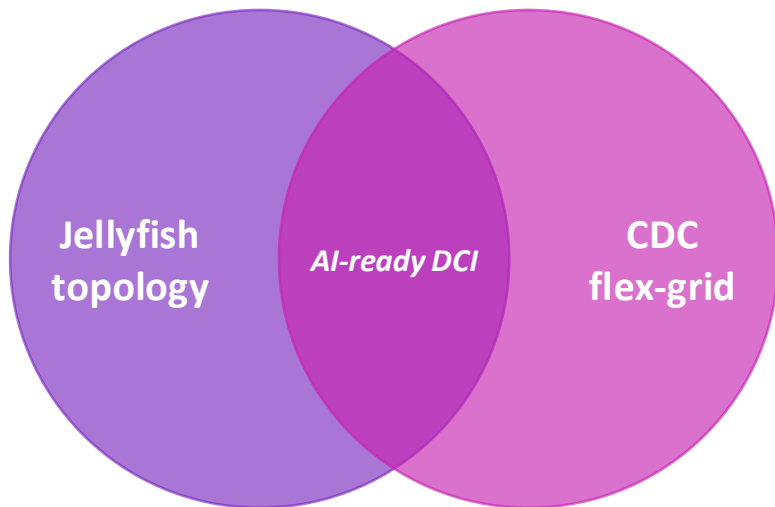
Flexible grid

Variable spectrum widths — 50, 75, 100, 150 GHz on demand.

Jellyfish topology × CDC photonic layer

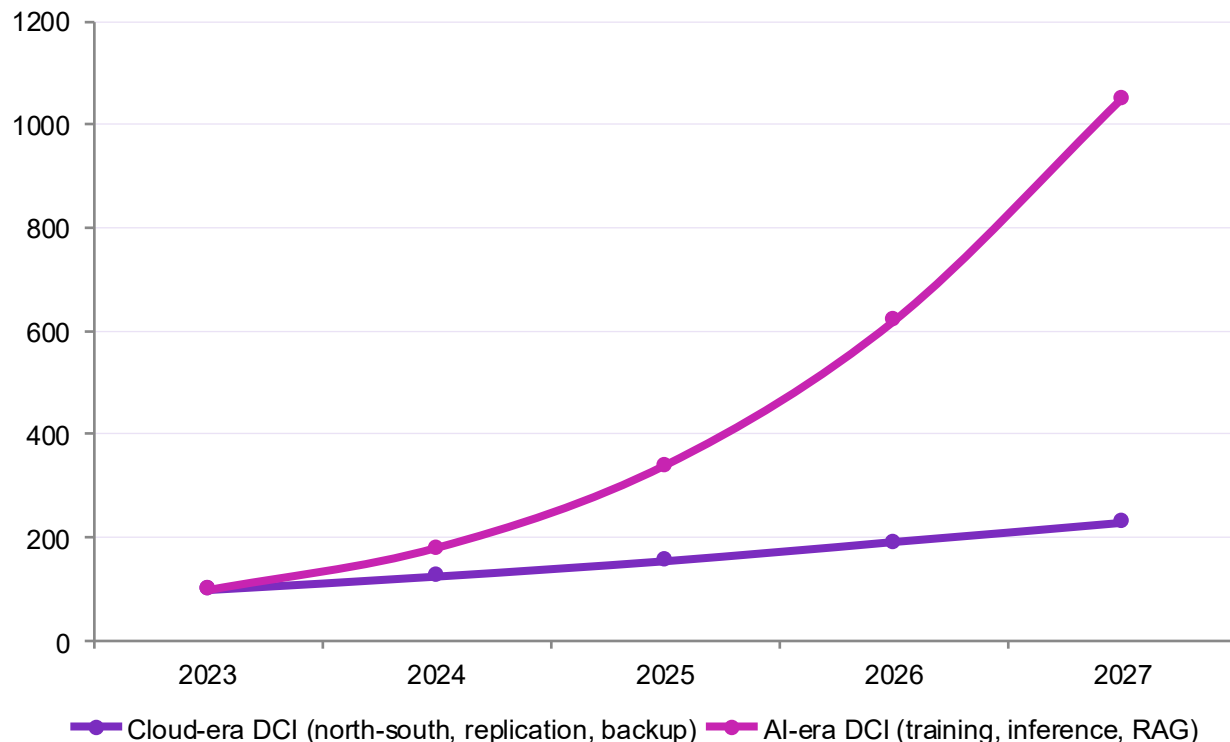
One enables the other. Neither works alone at AI scale.

- Random graph wiring is only feasible when any λ can land on any port
- Incremental expansion needs on-demand wavelength assignment
- Path-diverse restoration needs mesh-recovery + flex-grid
- Defragmentation recovers stranded capacity an unstructured topology will create



The capacity curve has bent

Indicative DCI demand profile across Indian metros, indexed to 2023 = 100



BY 2027

~10x

AI-driven DCI capacity
vs. 2023 baseline

Cloud DCI grows ~2.3x
in the same window.

What we're doing at Lightstorm

Polarin is evolving from a cloud DCI to an AI-grade fabric — in phases.

PHASE 1 · NOW

1

Lift the optical floor

Complete CDC + flex-grid coverage across all 10 cloud regions. Pre-connected transponder pools at every major node. Path Computation Engine in production for mesh restoration.

PHASE 2 · 2026

2

Jellyfish-style DCI overlays

Random-graph wavelength meshes between AI campuses — Mumbai inner-metro, NCR, Chennai. Telemetry-driven path selection. Begin shifting AI training tenants off rigid Clos overlays.

PHASE 3 · 2027+

3

Operate at AI scale

Closed-loop optimisation. Spectrum defragmentation as routine ops. Tenant-aware path engineering for collective communication patterns. Incremental capacity adds as a daily operation, not a quarterly project.

What this unlocks

Operationally, commercially, and for India's AI ambitions.



25%+

more usable capacity

same optics, same switches, same footprint



50%

protection overhead reclaimed

1:N restoration replaces 1+1 reservation



Days

to deliver new capacity

down from quarters — pre-connected pools



<1/2

average hop count vs. Clos

shorter paths = lower latency for AllReduce



1×1

expansion unit

add one switch or one λ at a time — not whole pods



100%

of Polaris is programmable

the optical floor is already AI-ready

Three Takeaways

If you remember nothing else...

01

AI workloads are in the DCI now

Training and inference fabrics stretch across buildings, campuses, metros. The DCI is no longer a backup pipe — it's the cluster fabric. Plan for ~10× growth.

02

Topology orthodoxy needs revisiting

Fat-tree's rigidity is a tax on the AI era. Random graphs — Jellyfish, expanders — deliver +25% capacity, shorter paths, and incremental expansion the workload demands.

03

The photonic layer is the unlock

CDC + flex-grid ROADMs are what make random topologies operationally real. India's AI fabric will be programmable optical first, packet topology second.

Thank You

Let's wire India for what's next.

APNIC 62

MUMBAI, INDIA

4 – 10 September 2026

#apnic62

