LUMEN®

AI BACKBONE PLAYBOOK

Build the network like AI depends on it. Because it does.

Powered by ciena

Executive summary

The hidden constraint in AI infrastructure: the network that determines performance, resiliency, and speed to launch.

The network may be a small line item in an AI build. But it decides whether AI performance shows up—or falls apart under load, failure, or growth.

Whether you're building purpose-built public AI infrastructure, standing up enterprise AI, or enabling others, one pattern keeps repeating: teams move fast on compute, power, and space—then discover the backbone is the limiter. Not because the network is "bad," but because AI workloads punish weak links.

AI changes the traffic profile in two important ways:

- Training moves massive datasets and checkpoints across sites. When that movement is slow or inconsistent, model cycles lengthen, slowing time to value.
- Inference is increasingly the real stress test. As usage grows, inference becomes a network resource hog in its own right, with tight experience expectations and very little tolerance for variability. Many AI applications operate within 10–40 milliseconds round-trip latency budgets; TTFT (time to first token) is one common way teams track experience impact. If the path can't hold that line consistently, the experience degrades, no matter how strong the model is.

This is why the backbone has to be a Day 1 design decision, not a cleanup task. The hidden cost isn't the circuit. It's the late-stage rework: redesigning routes, rethinking resiliency, and scrambling for capacity when demand jumps. That rework creates avoidable delays, change-order cost, and lost deployment momentum.

This paper breaks down the tradeoffs, and what they mean for AI performance, resiliency, and speed to capacity. You'll get:

- A decision framework for choosing waves vs. dark fiber—and the hybrid options in between—based on speed, control, scale, and operating model.
- A practical resiliency playbook that explains single points of failure and the real difference between unprotected, protected, diverse, and dual-diverse designs—including where active-active matters.
- Guidance tailored to AI builders/ Neoclouds, data center operators, and large enterprises—one set of principles, then the implications for each.

Bottom line: If you want AI performance you can count on, design the backbone for speed, resiliency, and change—because AI traffic won't grow in a straight line.

1

The overlooked constraint in AI builds

AI infrastructure conversations tend to start with what's scarce: GPUs, power, cooling, real estate. While logical, focusing on these topics can also hide a simple truth:

AI performance is end-to-end. If data can't move predictably, models can't train efficiently and inference can't deliver a consistent experience.

Business leaders usually feel backbone constraints before they can name them.

- **Launch delays:** Training and validation cycles stretch because moving data across sites takes longer than expected—lengthening model cycles and pushing out time to value.
- **Inconsistent user experience:** Inference feels fast one moment and sluggish the next—hard to debug and harder to trust.
- **Operational churn:** Teams spend time chasing "mystery performance" issues that turn out to be congestion, routing, or resiliency behavior.
- **Rework under pressure:** Capacity arrives late, diversity is missing, and resiliency becomes a retrofit, driving avoidable delays, change-order cost, and lost deployment momentum.

In short: The backbone isn't a background dependency. It's part of the product experience.

2

A small line item. A big limiter.

Here's the contrarian point: **When network spend is a small percentage of the build, you can afford to design it right—and you can't afford to design it late.**

Late-stage backbone constraints create costs that don't show up neatly on a bill of materials:

- **Schedule risk:** Upgrades and route changes don't move at the speed of software releases.
- **Performance risk:** Adding bandwidth doesn't automatically fix congestion behavior, path variability, or failure-mode slowdowns
- **Fragility risk:** Rushed changes introduce gaps—partial diversity, untested failover, "backup" paths that can't carry real load.
- **Customer impact:** Inference performance becomes inconsistent, which feels like product instability even when the model is fine.

.

Buying criteria shouldn't start with "What's cheapest?" Instead, it should start with **Which backbone choices reduce rework, preserve performance, and scale even in times of uncertainty.**

3

The AI Backbone Checklist: What “good” looks like.

Use this checklist to make tradeoffs visible across technical and business stakeholders.

Performance

- Do we know the latency tolerance for the most important inference experiences—and what “bad” looks like to users?
- Do we know the throughput requirement for training data movement between sites (peak and sustained)?

Scale

- Can we scale capacity without redesigning the backbone?
- Do we have a clear upgrade path (for example, 100G > 400G > 800G) that matches adoption curves?

Resiliency

(failure behavior)

- What happens when there’s a fiber cut, equipment failure, power event, or planned maintenance?
- Do we have true route diversity, or “two paths” that share the same geography?

Speed to capacity

- If demand doubles in 90 days, what changes—and how fast can it change?
- How quickly can we add capacity on critical routes without breaking resiliency?

Operating model

- Do we want full control (and full responsibility), or a managed operating model?
- Who owns monitoring, troubleshooting, upgrades, and validating failover behavior under load?

If these answers aren’t clear, the backbone design isn’t finished—even if compute is.

4 Backbone options and tradeoffs

Backbone choices aren't just technical preferences. They shape how quickly you can scale, how failures behave, and how much operational burden you carry. In practice, most AI infrastructure ends up using a **combination of waves and dark fiber**, with **managed hybrid options** in between (such as MOFN) to balance control and complexity by route and geography.

Waves

Best fit when speed matters. Waves are used for rapid deployment, flexibility, and scaling as demand shifts, especially on higher-capacity inter-city routes where time-to-capacity is a competitive advantage.

Tradeoffs: You gain speed and a clear upgrade path, but you may have less direct control over how the underlying transport is engineered than in a self-managed fiber model.

Dark fiber
(privately operated fiber)

Best fit when traffic is predictably heavy and you want full control. Dark fiber can be cost-efficient in metro connectivity and compelling at scale when engineering control and long-run economics are priorities.

Tradeoffs: Dark fiber typically comes with longer deployment timelines and longer contract terms, and it shifts more operational responsibility to your team (or your chosen partner).

Hybrid options in between
(MOFN, for example)

Best fit when you want fiber-based capability with a managed operating model. MOFN can be a practical middle path, especially in metro scenarios—when you want performance and control characteristics without taking on full day-to-day operational burden.

Decision framework

Start with the constraint and apply it per route (long-haul vs metro, inference-critical vs bulk movement):

If the constraint is **time-to-capacity** → Waves tend to fit early-stage builds and fast expansions.

If the constraint is **engineering control at sustained high volume** → Dark fiber can be the right foundation, especially where metro fiber economics are favorable.

If the constraint is **operational complexity** → Hybrid managed options in between can provide a workable balance.

5

Resiliency is a familiar word.
The real differentiator is what breaks, what degrades, and what stays stable.

Resiliency planning often stops at uptime. AI needs a step further to failure behavior. Many problems show up as degradation—latency spikes, timeouts, throughput collapse, before they show up as a clean outage.

Single points of failure: the risk you can't ignore. Any route, component, **piece of equipment**, or dependency where failure causes a service-impacting event, because there's no effective alternative—creates risk to AI performance.

Common places single points of failure hide:

- A single physical route (shared conduit or shared geography)
- A single metro gateway/meet-me room dependency
- A single regen/amplification dependency on long-haul paths
- A single device/chassis carrying critical traffic
- A failover design that exists on paper but hasn't been proven under load

The goal isn't "zero risk." It's removing **avoidable** single points of failure on the paths that matter most.

Four Levels of Network Resiliency and Risk

Unprotected

One path. If it fails, traffic stops—or degrades sharply. This concentrates risk in the route and the equipment on that path, so it's best reserved for non-critical workloads, early pilots, or situations where disruption is acceptable. Example planning model: **~99.45%** availability in common wave architectures.

Protected

Two paths, but they follow the same geography. This can reduce certain equipment-related risks, but a geographic event (shared conduit issue, construction cut, localized impact) can still take both paths down. Protected can be a meaningful step up from unprotected, but it's not the same as true diversity. Example planning model: **~99.99%** availability in common protected wave architectures.

Diverse

Two paths that are geographically separate. This materially reduces correlated failure risk and is often the right baseline for inference flows that can't tolerate disruption or large performance swings. Diversity is as much about **where the paths go** as it is about having two paths.

Dual-diverse

Multiple gateways in a city so redundancy doesn't require a "detour city." This reduces chokepoint risk while protecting latency-sensitive traffic. Dual-diverse matters most when you need resiliency without paying for it in user experience.

Practical use: Set your resiliency tier based on what the experience can tolerate. If the inference experience can't degrade without business impact, diversity becomes a performance requirement, not just an availability upgrade.

6

Active-active protects performance

Active-active runs meaningful traffic across both paths, rather than keeping one path idle until a failure. This is less about "can we fail over?" and more about **whether performance can hold steady during incidents, maintenance, and spikes.**

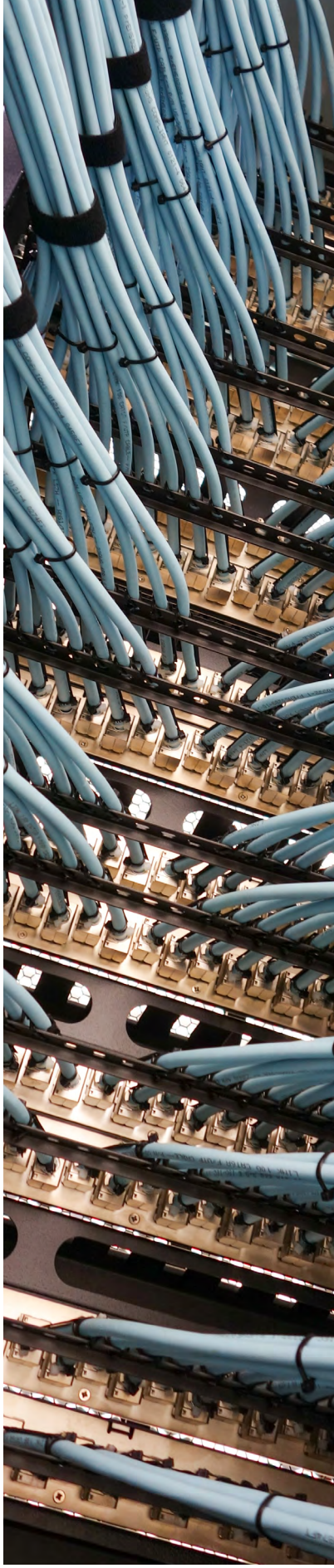
Active-active can:

- Stabilize latency by spreading load
- Reduce congestion spikes that happen when traffic suddenly shifts during failover
- Make maintenance less disruptive because you're not betting your experience on a seldom-used path
- Force operational discipline: both paths must be real, tested, and provisioned

Where active-active tends to be worth the complexity:

- Inference paths tied to user experience or revenue
- Environments where performance variability is effectively a failure
- Routes where capacity headroom during incidents is non-negotiable

The practical takeaway: If inference consistency matters, design for performance during failure, not merely recovery after failure occurs.



7

AI traffic doesn't grow in a straight line

Most networks weren't designed for AI-era demand. Patterns shift with model cycles, adoption, where compute can be deployed, and where users are served. That's why backbone plans built on steady, linear forecasts tend to break at the worst time, right when demand spikes, performance expectations tighten, and change becomes risky. AI traffic is often:

- **Bursty:** retrains, fine-tunes, and batch work create spikes
- **Step-function growth:** a new customer, a new feature, a new model, a new region
- **Unpredictable in mix:** training dominates early; inference accelerates as products mature and usage scales

Two implications follow:

1 Flexibility is a design requirement

You need a plan for adding capacity quickly without destabilizing performance.

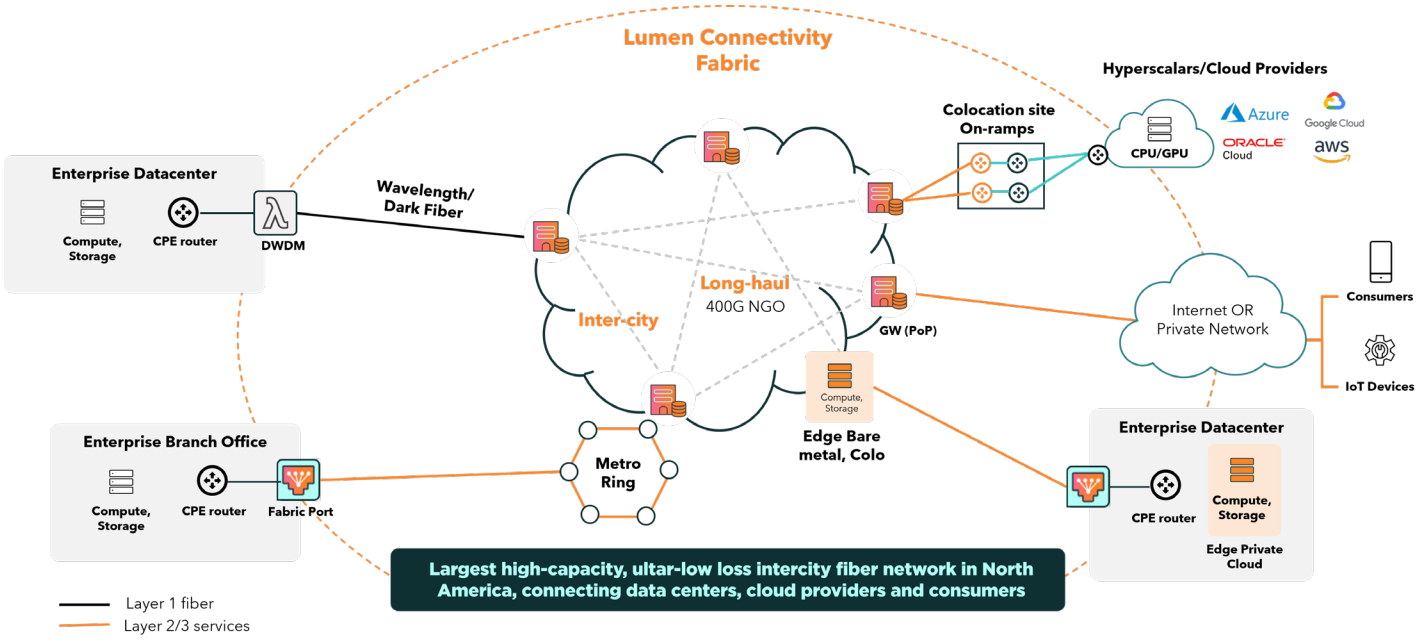
2 Inference becomes the long-term load

Training is heavy. Inference is persistent, and scales with adoption. As AI matures, inference can become the backbone demand driver because it is continuous, user-visible, and sensitive to variability.

8

Network infrastructure architecture for AI performance

AI performance depends on more than compute. Models and applications rely on a connected path that spans enterprise sites, colocation and edge environments, cloud on-ramps, and end users. The diagram shows an end-to-end view—from enterprise data centers and branch locations, through edge/colo and cloud providers, out to consumers and IoT devices—and the backbone connectivity that ties those environments together.



Key takeaways

AI traffic is distributed by default. Training data, model artifacts, and inference requests move across multiple locations—enterprise sites, AI data centers, edge/colo, and cloud on-ramps, so the backbone becomes part of the AI system.

The backbone must support multiple "layers" of need. The architecture includes options at Layer 1 fiber and Layer 2/3 services—because different routes and workloads require different operating models and levels of control.

Edge and colo are where performance expectations get real. AI experiences often depend on proximity and fast response; placing compute closer to users helps, but only if the end-to-end path stays consistent.

On-ramps are strategic, not incidental. Colocation sites and cloud on-ramps reduce friction for connecting enterprise and AI infrastructure into hyperscalers and other cloud providers.

Resiliency is an architecture decision. Diversity and active-active patterns need to be designed at the backbone and gateway layers, not bolted on after performance issues appear.

The architecture highlights the role of **gateways/PoPs** and **on-ramps** as the handoff points where performance and resiliency decisions show up first.

This end-to-end view is what makes backbone decisions consequential. Once you map where training traffic originates, where inference must be served, and which handoffs are unavoidable (gateways, on-ramps, inter-city routes), you can choose the right mix of connectivity models by route and apply resiliency where it changes outcomes—protecting user experience, keeping capacity usable during incidents, and reducing redesign risk as demand shifts.

9

Same backbone principles. Different constraints.

AI builders / Neoclouds

Primary pressure: speed and change. Sites, demand, and traffic patterns evolve quickly. Beyond raw performance, builders need connectivity economics that support AI applications as usage scales.

What matters most:

- Time-to-capacity on critical routes
- A scaling path that doesn't require redesign
- Diversity and active-active where inference experience is the product
- The ability to mix models by route: speed here, control there, hybrid where it fits

Data center operators

Primary pressure: Customer expectations are changing, and AI is a major driver. Tenants increasingly expect facilities to support high-capacity interconnect, clear resiliency tiers, and fast expansion, because AI workloads demand it.

What matters most:

- Standardized connectivity tiers customers can select based on workload criticality
- Clear definitions of diversity (so "resilient" means something measurable)
- A capacity roadmap that scales across the footprint as AI demand ramps

The opportunity is straightforward: Operators that make AI-ready connectivity easier to buy, and easier to trust—reduce friction for everyone downstream.

Large enterprises

Primary pressure: modernization plus governance. Many enterprises are layering AI workloads onto networks built for a different era, while maintaining reliability and security expectations.

What matters most:

- Upgrade paths that increase capacity without constant disruption
- Clear failure behavior: what breaks, what degrades, what stays stable
- Diversity where user experience and continuity demand it
- Security and privacy decisions made intentionally—not by default

For many enterprises, backbone modernization isn't separate from AI. It becomes the prerequisite for AI to perform consistently.

Clarifying the resiliency bar

Set resiliency and diversity requirements based on inference experience. If inference can't degrade without business impact, prioritize geographic diversity and consider active-active. Training can often tolerate more variability because it's less user-visible, even when it's expensive.

10

A practical path forward

Most backbone decisions don't fail because teams lack intelligence. They fail because tradeoffs aren't made explicit early, before timelines tighten and change becomes risky. A simple action plan helps align stakeholders around performance requirements, operating model, and failure behavior while choices are still flexible.

Start by defining two flows—training and inference—and be specific about what "good" looks like for each. Then set three targets: latency tolerance (especially for inference), near-term capacity, and a 12–24 month scaling path. Once those are clear, choose an operating model (waves, dark fiber, or a hybrid in between) route by route, and apply resiliency tiers where they actually change outcomes.

Then stress test the design with two questions: **What happens if a route fails?** and **What happens if demand doubles in 90 days—and then doubles again?** If the answers require a redesign, you've found the real work, while it's still cheap to fix.

Bottom line

AI will keep moving fast. The backbone can keep up quietly—or become the reason performance stalls. Design for speed, resiliency, and change, and you reduce the risk of late-stage surprises that show up as launch delays, inconsistent experience, and operational churn.

Pressure-test your AI backbone before it becomes a launch constraint. Review your routes, resiliency model, and scaling assumptions with an expert.