Network Design

2026-04-07

Table of contents

1 Overview

Designing large-scale networks that meet today's dynamic business and IT requirements is a complex undertaking, especially when the existing network was built for technologies and demands that are no longer current. As organizations adopt new architectures (cloud, IoT, SD-WAN, etc.), the underlying network often cannot support them without deliberate redesign. A structured design methodology is therefore essential.

There are two common approaches:

- **Top-down approach**: starts with business requirements and application needs, then works downward through logical design to physical infrastructure. This keeps the design aligned with what the business actually needs and reduces the risk of costly rework.
- **Bottom-up approach**: starts by selecting technologies and products first, then tries to fit business requirements around them. This carries a high risk of failure because the resulting network may not satisfy the organization's goals or application demands.

The top-down approach is the industry-recommended method and the one emphasized throughout the CCDE curriculum.

To achieve a successful strategic design, the primary focus must be on business priorities, drivers, and outcomes rather than on technology for its own sake. This means understanding technical objectives alongside existing and future services and applications. In today's networks, business requirements drive IT and network initiatives, not the other way around.

The three elements every network designer must understand are:

1. **Network design fundamentals**: the foundation that defines the mindset, use cases, and business context driving the design
2. **Network design principles**: the criteria used to evaluate and guide design decisions (security, scalability, availability, cost, manageability)
3. **Network design techniques**: the structural methods applied to meet those principles (modularity, hierarchy, failure isolation, fate sharing)

These elements work together to form the architectural basis of any network design (Figure 1).

As we go through this chapter, we will also examine the most common network design pitfalls.

2 Network Design Fundamentals

The six foundational elements every network designer must understand:

1. Mindset
2. Requirements
3. Design use cases
4. The business
5. Constraints
6. "Why"

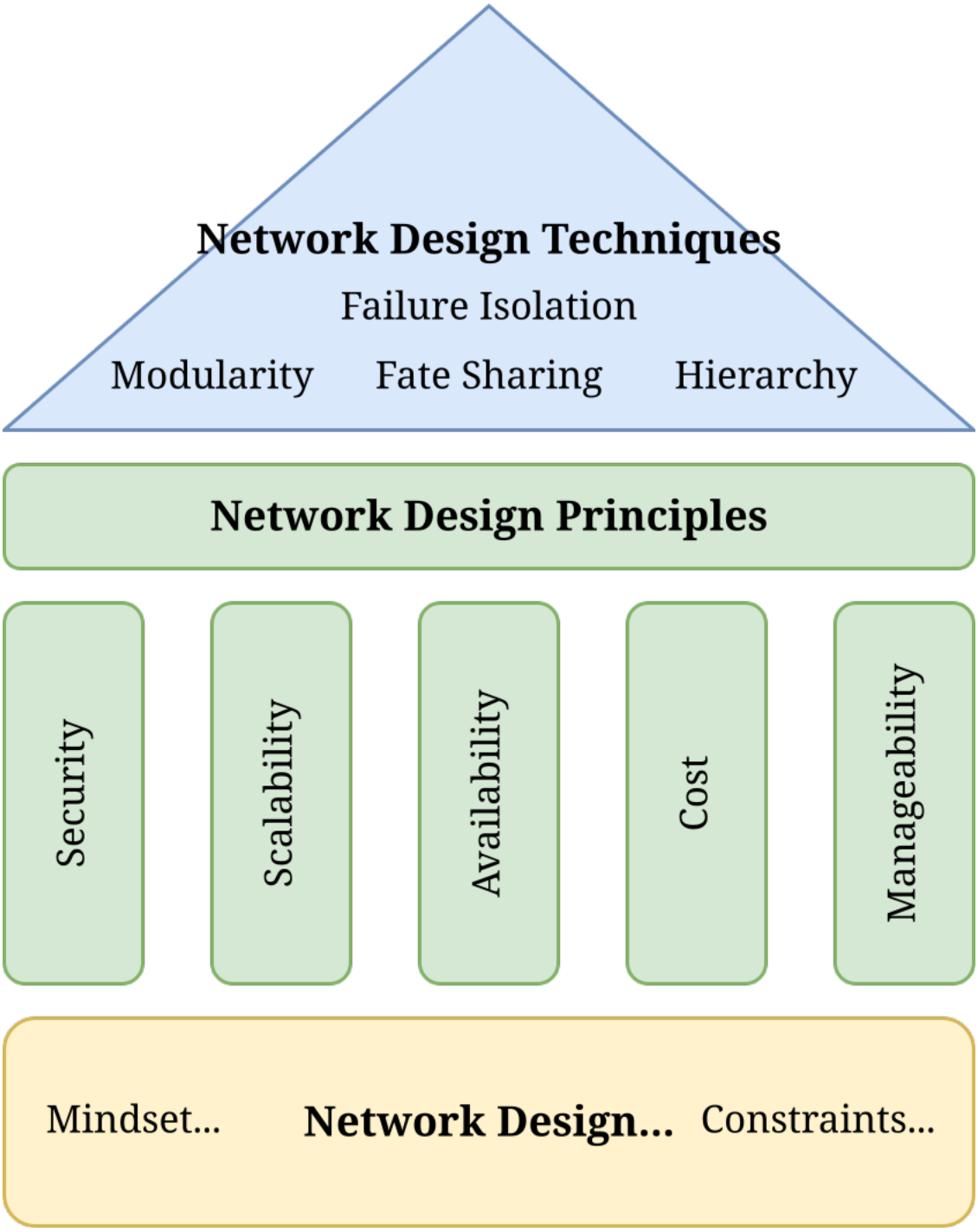

Figure 1: Network Design Framework

2.1 Mindset

Your mindset is the single most important factor in network design. Knowing the technology is necessary but relatively straightforward because you can study and learn what you don't know. What most engineers lack is a proper *design* mindset. An implementation mindset focused on configuring and deploying will not work for network design; you need to think in terms of trade-offs, business alignment, and holistic system behavior.

2.2 Requirements

Requirements form the foundation of any design. They fall into three categories:

2.2.1 Functional Requirements (behavioral requirements)

Functional requirements define what the system or technology must *do* from a technological standpoint. They are sometimes called behavioral requirements because they describe system behavior.

Example: "The PE routers must send VoIP traffic over 10G fiber links while data traffic is sent over the OC-48 link." This implies the PE routers need a mechanism like MPLS-TE to steer different traffic types over different paths.

> **Note**
>
> A design that does not address functional requirements is a poor design. In real-world scenarios, not all functional requirements are handed to the designer directly. Because it is the responsibility of the designer to discover and document these requirements, many must be discovered, documented, and signed off on before design work begins.

2.2.2 Nonfunctional Requirements (technical requirements)

Nonfunctional requirements (also called technical requirements) specify the technical aspects the infrastructure must provide: security, availability, and integration. They are the most dynamic requirement type because they change as technology evolves.

Examples:

- Heightened network availability (e.g., FHRP)
- Integration with network tools and services (e.g., NetFlow collectors, RADIUS servers)
- Infrastructure security techniques (e.g., control plane protection, infrastructure ACLs)

> **Note**
>
> Technical requirements help specify required features, protocols, software versions, and can influence hardware platform selection.

2.2.3 Application Requirements

Application requirements are the driving factors that dictate, and often constrain, a network design. Users' expectations (quality of experience) is typically the highest priority any design must satisfy.

End users fall into three categories:

- **Customers**: individuals or organizations whose satisfaction directly impacts business reputation and revenue
- **Internal users**: employees whose productivity translates to business performance efficiency
- **Business partners**: entities working together toward strategic goals, where efficient interaction enhances mutual success

A network that cannot deliver the desired quality of experience means failure to achieve business goals. The network becomes a cost center rather than a business enabler.

Application requirements can also drive functional requirements. For example, if an SLA requires VoIP traffic with less than 150 ms one-way delay and less than 1% packet loss, this drives the PE devices to use technologies like CBTS MPLS-TE with FRR protection.

Key questions when evaluating application requirements:

- How much network traffic does the application require?
- What is the criticality and service level requirement of this application?
- Does the application have separation requirements for regulatory or security compliance?
- What are the application's characteristics (traffic pattern, protocol, etc.)?
- How long does the application need after losing connectivity to reset its state or session?

2.2.4 Crafting the Design Requirements

Different requirement types collectively lead to the desired network design, which ultimately facilitates business goals. The flow goes from business outcomes down to technical requirements (features, protocols).

Design scope must be determined before gathering information:

- Is this a greenfield or brownfield (production) network?
- Does the design span a single module or multiple modules?

Design Scope	Example
Enterprise campus + remote sites	IP telephony rollout requiring redesign of VLANs, QoS across LAN, WAN, DC, and remote-access edge
Campus only	Adding multi-tenancy requiring VLANs, IPs, and path isolation across the campus
Enterprise edge availability	Adding a redundant link for remote access, requiring WAN module and remote site redesign

> **Note**
>
> Identifying what is *out of scope* is equally important. It prevents scope creep and clarifies what is available to the designer in their decisions.

2.3 Design Use Cases

Properly identifying the specific design use case is critical to building a successful design.

2.3.1 Greenfield

A clean-slate design with no existing infrastructure. This is the best situation for a designer, but you must ensure what you propose is actually needed by the business.

2.3.2 Brownfield

The most common use case. An environment with production traffic already running. Spend time up front discovering the current state, including both the technical details (protocols, diagrams) and the business and its lines of effort. Understand what the business is trying to accomplish before making design decisions. When you do make changes, prepare a migration plan and limit potential failures.

2.3.3 Add Technology or Application

Adding technology or an application to an existing network. Consider: will anything break? Understand the application's traffic pattern, convergence time, delay requirements, and fine-tune the network (QoS, etc.) to deliver the desired experience.

2.3.4 Replace Technology

Replacing an existing technology (WAN technology, routing protocol, security mechanism, core network technology, etc.) to meet new requirements. Consider the implications on the current design, such as enhanced scalability or potential conflicts with existing application requirements.

> **Note**
>
> When replacing or adding technology, ensure there are direct business justifications. Have a properly tested migration plan with validation tasks at each step and a backout plan in case something goes wrong.

2.3.5 Merge or Divest

One of the most challenging use cases. Merging or separating networks means integrating (or splitting) different design philosophies with potentially conflicting concepts.

- **Merger**: combining two businesses into one end-to-end architecture. Watch for overlapping technologies (e.g., overlapping RFC 1918 subnets). Have a short-term plan (e.g., NAT) and a long-term plan (e.g., renumbering during a planned upgrade). Consolidate where possible (e.g., four data centers down to two, unless requirements dictate otherwise).
- **Divestment**: splitting apart a business and its architecture. The hardest design use case. The network designer must ensure each resulting business can still function along its lines of effort with the independent architectures left over.

2.3.6 Scaling a Network

Covers scalability at multiple levels: physical topology, Layer 2, and Layer 3. Considerations:

- Is the growth planned or organic?
- Are there issues caused by the growth?
- Should the network be redesigned to account for growth?
- What is the most scalable design model?

Example: a single flat OSPF area 0 that no longer scales. You could leverage multiple areas, area types, and LSA filtering.

2.3.7 Design Failure

Nine times out of ten, design failure is the design use case you will be brought in to fix. There is a problem and you must resolve it, much like an ER doctor identifying and fixing issues as quickly as possible. Example: misaligned STP root bridge and FHRP default gateways causing suboptimal routing.

2.4 The Business

Network designers make design decisions for the sake of the businesses they support. Specifically, we design so that businesses can make money (or, for not-for-profit and public sector organizations, achieve their specific goals and reduce costs).

Designing without understanding the business purpose leads to decisions made "just because" or "it's how we've always done it." This is a path to failure.

> **Note**
>
> Document all design decisions and the reasons behind them in a design binder. This allows all team members, whether past, present, or future, to understand why a feature or design option was implemented. Remembering why a decision was made 6 months ago is far easier than 2 years later.

2.5 Constraints

Constraints are the hard rules and limitations that box in a network designer. They fall into three categories:

- **Business constraints**: A business constraint can be as simple as a monetary budget limitation. Other examples include staffing constraints (lack of skilled personnel), contractual constraints (scope or duration)
- **Application constraints**: Application constraints are limitations to us as designers because an application was developed in a specific way. (e.g., requiring Layer 2 connectivity, hard-coded IP addresses, multicast support for cluster technologies)
- **Technology constraints**: locked into specific vendor hardware, proprietary solutions, or inability to use proprietary technology

No two design situations have the same constraints. Don't assume constraints. Qualify each one with evidence.

The most common constraints that a network designer must consider:

Constraint	Description
Cost	Most common limiting factor. Only consider as a constraint if explicitly mentioned. For the CCDE purpose, don't assume technology A costs more than B based on personal experience. MPLS circuits are not automatically more expensive than Internet circuits in the context of the CCDE exam, as an example.
Time	Aggressive timelines for resolution, especially in design failure scenarios.
Location	Can introduce indirect limitations. For example, a remote site with no fiber and only wireless connectivity will have reduced link speed, affecting sensitive applications.
Infrastructure	Legacy devices that won't be replaced can limit the design if new features or protocols aren't supported.
Staff expertise	Proposing cutting-edge technology is problematic if staff can't operate it. Options: train existing staff (risk of longer resolution times) or hire experienced staff (higher operational cost).

2.6 "Why"

Customers often declare something a "requirement" but cannot explain why. Use the concept of the five levels of "Why" by keep asking until you reach the root cause. This is similar to Toyota's "Five Whys" system for investigating production problems.

For example:

1. "We need SD-WAN." — Why?
2. "Our WAN is too expensive." — Why?
3. "We're paying for MPLS circuits we don't fully use." — Why?
4. "Traffic patterns changed when we moved apps to the cloud." — Why?
5. "The business shifted to SaaS two years ago but the network was never redesigned."

Now the real problem is clear: the network wasn't redesigned after a business shift. The designer can address the root cause rather than just deploying SD-WAN because it's trendy.

Many evolving technologies (5G, Zero Trust, cloud, SASE, DevOps) are seen as silver bullets. Network designers must determine whether these solutions are truly needed by the business.

Example questions to surface the real "why" without literally asking "why" repeatedly:

- What are you solving with this technology?
- What are your employees'/customers' workflows today?
- How will this technology change these workflows?
- How will your business (governance, policy, processes, culture) evolve because of this technology?

You want the customer to articulate specific problems. They believe a solution will solve issues, save money, increase security, or simplify operations. Right or wrong, there is always a reason, and it's the network designer's job to find it. Once you understand the true "why," that is where you start to design, architect, and engineer.

Review Questions

1. Which of the following options are network design fundamentals? (Choose two.)

a. Security
b. Redundancy
c. Constraints
d. Scalability
e. Mindset

Answer

c and e. Constraints and mindset are the only two network design fundamentals listed. Security and scalability are both network design principles. The six network design fundamentals are mindset, design use cases, the business, constraints, requirements, and "why?"

---

2. From a network design perspective, what are the three categories of constraints?

a. Application, security, and business
b. Functional, technical, and application
c. Compliance, technical, functional
d. Business, application, and technology

Answer

d. The three categories of constraints are business, application, and technology. The only answer that has these three items is option d. The other options are incorrect.

---

3. Which network design use case focuses on when a company or business is split into two or more entities?

a. Scaling
b. Divest
c. Design failure
d. Merger

Answer

b. Divest is where you as a network designer are splitting a business or company and the corresponding architecture into two or more functional independent architectures. Scaling is the network design use case that focuses on the scalability of the technology or holistic architecture. In a design failure use case, there is a problem to resolve. A merger is when two or more different networks are integrated to create one single end-to-end architecture.

3 Network Design Principles

The five network design principles:

1. Security
2. Scalability
3. Availability
4. Cost
5. Manageability

3.1 Unstated Requirements

It has become more prevalent where customers do not articulate their specific requirements. They assume requirements, and you as the network designer have to figure out what requirements are important and determine the level of each requirement. For example, does this network require no single points of failure or no dual points of failure?

Every network design principle (security, scalability, availability, cost, manageability) has become an unstated requirement. Customers assume these are a given and won't explicitly state them, but the designer must still identify and quantify the required level of each.

3.2 Pervasive Security

Historically, security hasn't been identified as a network design principle. It's been added because of the impact it has on the overall business:

- What happens to your business if your network is compromised?
- What happens to your business if the integrity of your data is compromised?

Possible impacts: business reputation suffers, customers lose trust, revenue loss, compliance failures (fines or shutdown), and in extreme cases, business failure.

Three security models every network designer should know:

- **Perimeter security (turtle shell)**: Legacy model with a firewall at the perimeter (the turtle shell). No security devices inside the network. Full east–west (lateral movement) traffic between users and resources. Inside threats become prevalent.
- **Session- and transaction-based security**: Evolved from perimeter security. Users and devices are locked down, including resources like printers, applications, and cameras. East–west traffic is secured dynamically based on what the device is, who is using it, where they are using it, and what they need access to (not what they want). This leads to 100% authentication and 100% authorization of each session and transaction.
- **Zero Trust Architecture**: Adds real-time capture and analytics (AI/ML) for real-time decision making. Every device, user, application, server, service, and resource (even data itself) is assigned a trust score that changes based on what the analytics engine observes. For example, a user connecting from a coffee shop over VPN might get a lower trust score (less access) than one at a company location. Dynamic characteristics like time of day, type of data, or volume of data can increase or decrease a trust score on the fly.

Wherever possible, include security capabilities to ensure confidentiality, integrity, and availability requirements are met. A business cannot fulfill its goals, outcomes, or mission if the business or its data is compromised. This is also where compliance requirements like HIPAA, NIST, and PCI DSS come into play. Non-compliance can result in disconnection or shutdown.

3.3 Shifting of Availability

Availability encompasses redundancy, resiliency, reliability, and more:

- **Redundancy**: Multiple resources performing the same function/role so that if one fails, the other takes over with limited to no impact on production traffic
- **Resilience**: The ability of the network to automatically fail over when an outage occurs
- **Reliability**: How much of the network data gets from source to destination in the right amount of time to be leveraged correctly

The need for availability is just assumed today. What level of availability is needed is the true question. Once identified, the designer must assess the complexity and cost (both monetary and non-monetary) of that level of availability. As you increase the level of availability, complexity and cost increase with it.

What level of redundancy, resiliency, and reliability is too much? When the increased complexity, cost, and the associated return for availability are not worth it. For example, redundant links (two) are simplest, four links can be preferred in some architectures, but five or more tends to add complexity with diminishing returns.

The large shift with availability is that the focus of network design is no longer network availability but rather application and service availability. The network is a service, getting data from point A to point B at the right time. As an analogy, the network is the plumbing and data is the water.

As network designers, we must identify the required level of availability for applications and services (in most cases, requirements are unstated). We must partner with application owners to understand requirements and interdependencies, and make appropriate design decisions to ensure the required level of availability is achieved.

3.4 Limiting Complexity — Manageability

When comparing different design options that provide the same capability, choosing the simpler option is the way to go. Keep it super simple (KISS).

A key question: "Can the network design I'm proposing be managed by the team at hand?" If your design has multiple CCIE-level elements but the customer has no CCIE-skilled professionals, how can they manage or troubleshoot it?

If no other choice exists but to leverage a more complex design and the local team lacks the skill sets, raise the issue with the business. Assume the role of trusted advisor and explain in business terms (not technical terms) why they need higher-level skilled professionals.

Obfuscating the complexity of a solution still yields a complex solution. Leveraging technology to hide complexity does not make it simple; it might make it manageable, but it's still complex (and often more so). Example: using a GRE tunnel to form a routing adjacency over a complex OSPF multi-area design hides the underlay but doesn't reduce the complexity.

3.5 Making a Business Flexible with Scalability

Scalability has always been a network design principle and will most likely always be one. Scalability is about more than making the network scalable. Scalability is also about making the business flexible. You can provide flexibility to the business through your network design, allowing the business to adapt on the fly. By doing this, the network becomes a business enabler and is no longer a cost center.

Examples:

- **Architecture level**: designing a data center with purpose-built pods within a spine-leaf architecture provides extreme scalability and business flexibility
- **Low-level design**: leveraging routing boundaries, OSPF areas/area types, and LSA filtering techniques to increase routing scalability

3.6 Cost Constraints and What to Do

There is always a cost constraint. Cost isn't always monetary; it can also be a resource cost: personnel, time, and technical costs like memory, CPU, storage, bandwidth, power, and cooling.

Never be so disconnected from your customer that you don't realize the cost budget is drastically lower than the proposed design's cost.

If the cost budget is drastically lower than the proposed design, show the customer the pros and cons for each design option, including all associated costs. Let them make an educated decision.

Review Questions

4. Which of the following are security models? (Choose three.)

a. Perimeter security
b. Cybersecurity
c. Zero Trust Architecture
d. Session- and transaction-based security

Answer

a, c, and d. Perimeter security, Zero Trust Architecture, and session- and transaction-based security are the security models the industry has been shifting between over the last 20-plus years. Cybersecurity is not a security model but a component within these three models.

---

5. The ability of the network to automatically fail over when an outage occurs is the definition of what availability component?

a. Reliability
b. Resilience
c. Redundancy
d. Routing failover

Answer

b. Resilience is the ability of the network to automatically fail over when an outage occurs. Reliability is best defined as a measure of how much of the network data gets from source to destination in the right amount of time. Redundancy is the concept of having multiple resources performing the same function/role so that if one fails, the other takes over. Routing failover is the ability of the routing protocol and configuration to fail traffic between different routing paths.

---

6. How much of the network data gets from source to destination locations in the right amount of time to properly be leveraged correctly is an example definition of what availability component?

a. Reliability
b. Resilience
c. Redundancy

d. Routing failover

Answer

a. Reliability is best defined as how much of the network data gets from source to destination locations in the right amount of time to properly be leveraged correctly. Resilience is the ability of the network to automatically fail over when an outage occurs. Redundancy is the concept of having multiple resources performing the same function/role so that if one fails, the other takes over. Routing failover is the ability of the routing protocol and configuration to fail traffic between different routing paths.

4 Network Design Techniques

To help address the concept of "bad network designs do happen," this section covers the most common network design techniques that all network designers should know, both to avoid creating bad designs and to remediate bad designs others have created.

We'll use a higher education campus as a case study throughout this section. Figure 2 shows the current architecture topology for this higher education campus.

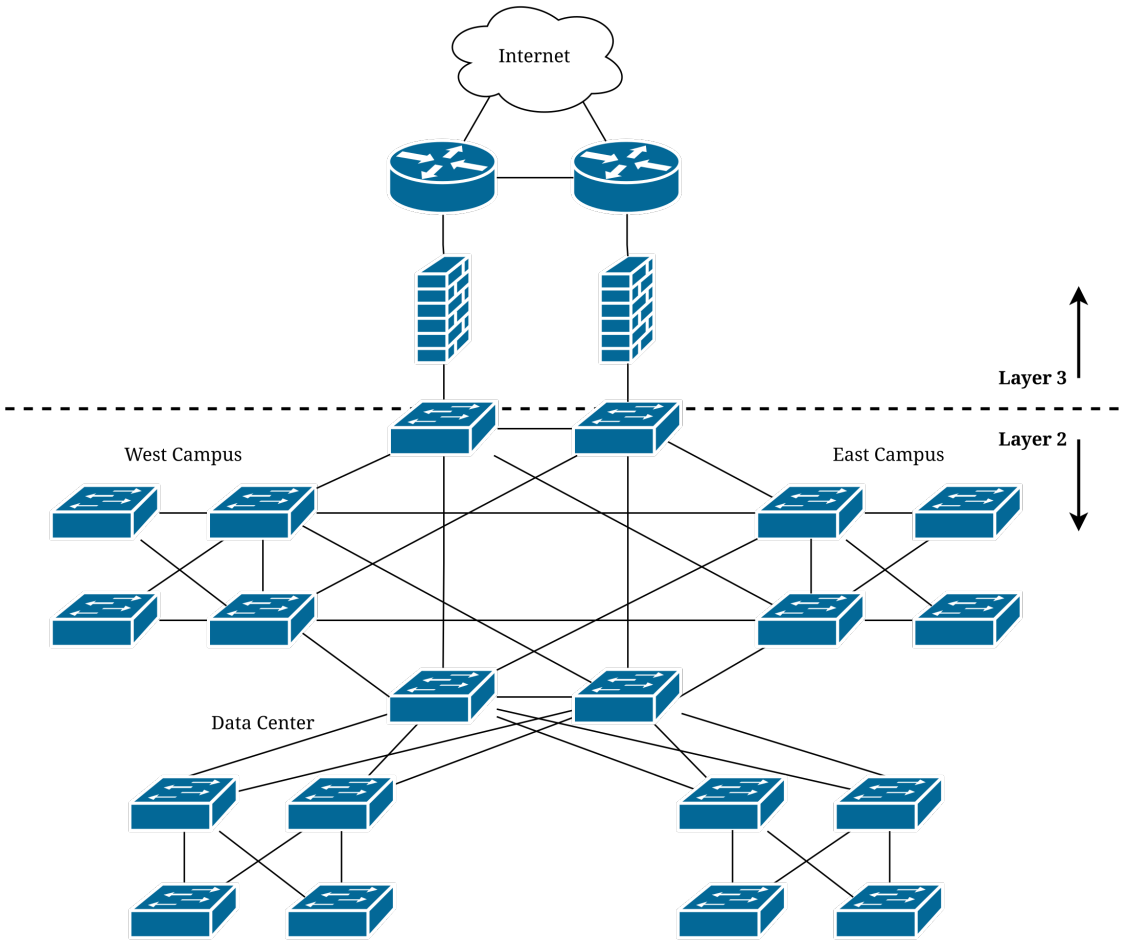

Figure 2: Higher Education Campus Architecture Overview

4.1 Failure Isolation

A failure domain is an area in which an outage can propagate. Failure isolation involves creating logical boundaries to limit the propagation of failures. This is also referred to as failure radius, impact domain, and failure boundary.

In the campus architecture (see Figure 2), there is a large Layer 2 environment spanning multiple locations. If a student plugged a hub device into an access switch twice (two ports, two cables) in the east campus, a broadcast storm would be created as shown in Figure 3. In this design, that broadcast storm would propagate to the main routers, the west campus, and the data center. One device and one student could bring down the entire campus in minutes. This is a design failure.

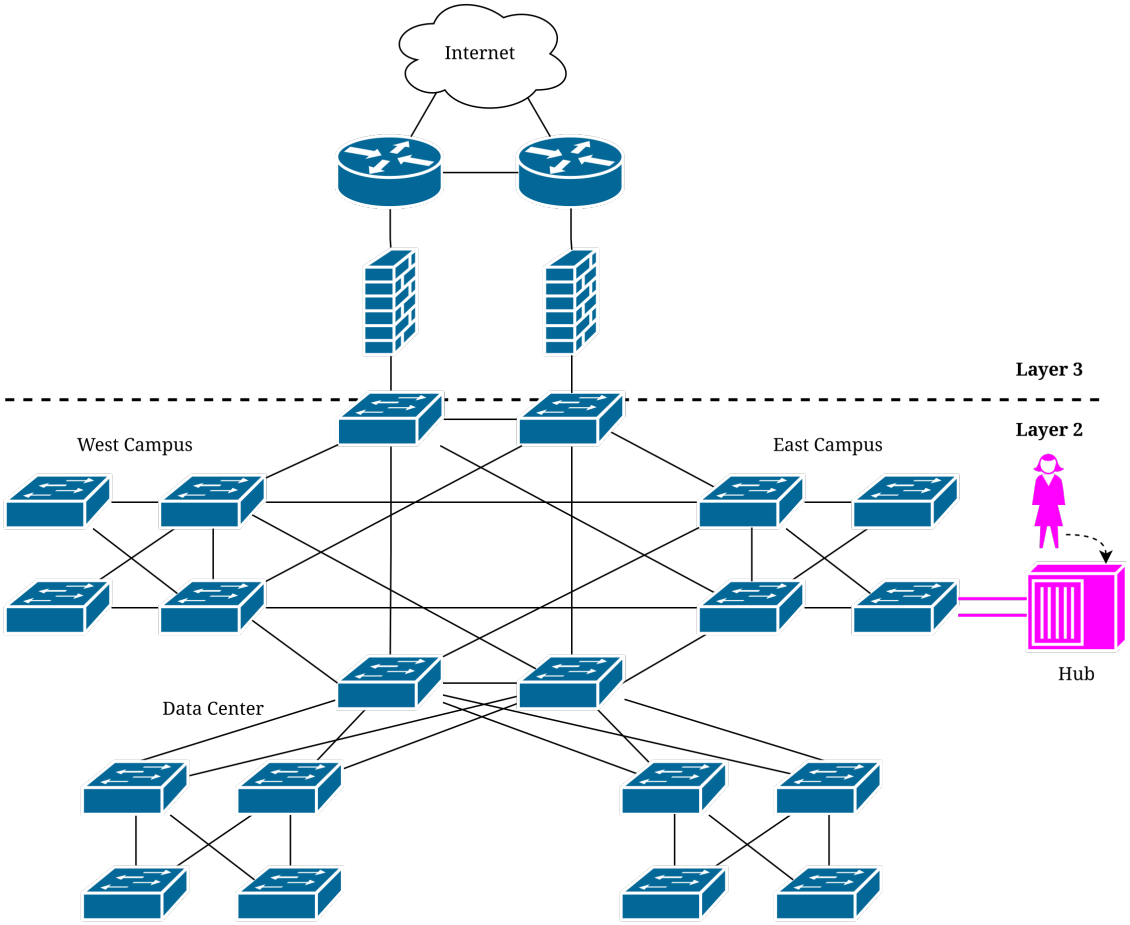

Figure 3: Higher Education Campus Architecture Broadcast Storm Propagation

The solution: push the Layer 2/Layer 3 boundary from the main routers to each location's core devices, as shown in Figure 4. The local site core devices all become routers and Layer 2 links no longer run between locations.

After implementing failure isolation, a Layer 2 broadcast storm would be isolated to the specific location and not propagate to the entire architecture (see Figure 5).

Failure isolation can be leveraged in almost every networking protocol. This applies within Layer 3 as well. A flat OSPF area 0 design (Figure 6) with 100 new satellite campus locations would cause route reconvergence churn impactful to all devices. Every time a new router is added or removed, these routes would be added and removed in every router within the OSPF area 0 design.

Instead, design a multi-area OSPF with multiple area types (Figure 7):

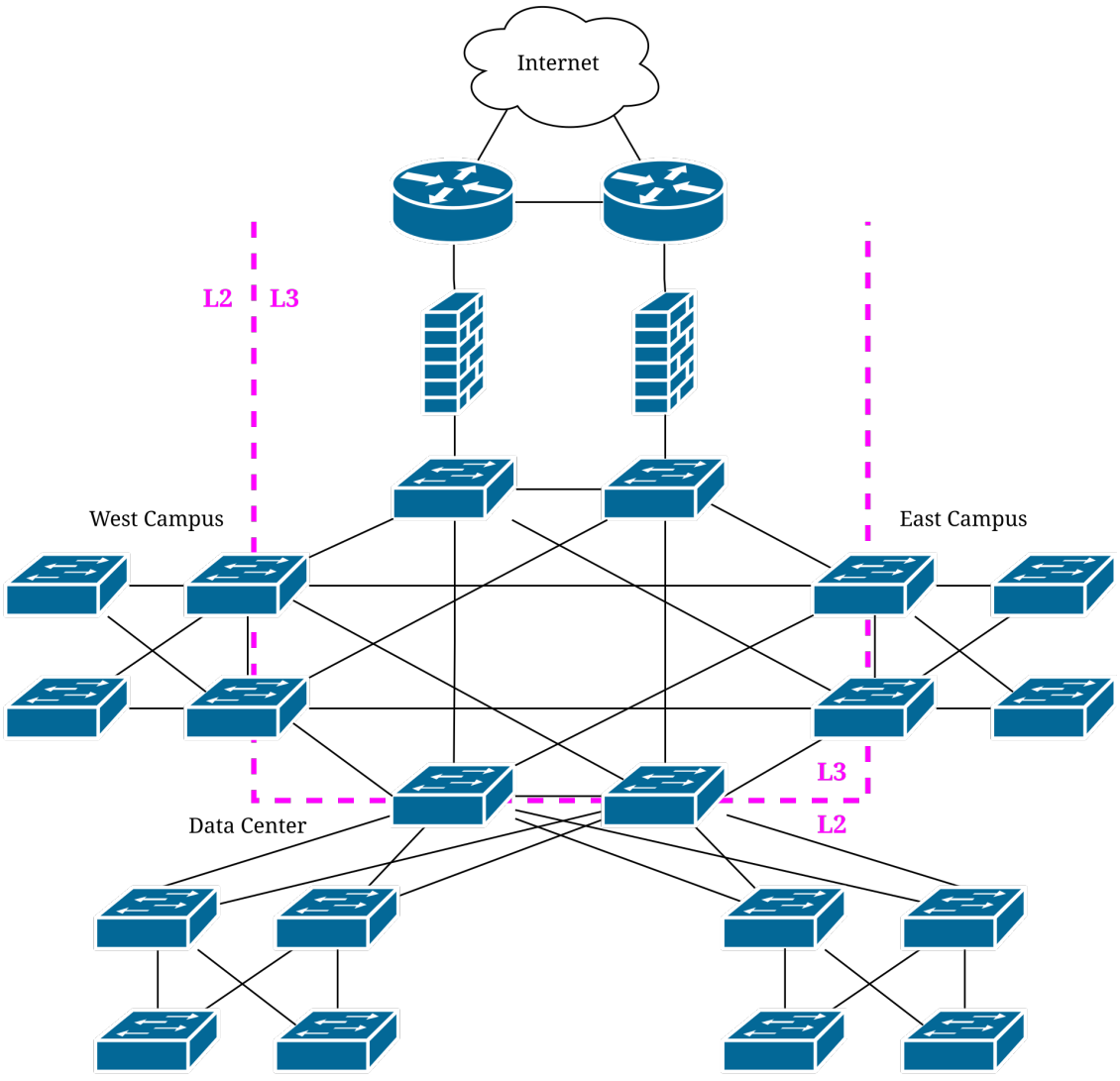

Figure 4: Higher Education Campus Architecture New Layer 2 and Layer 3 Boundary

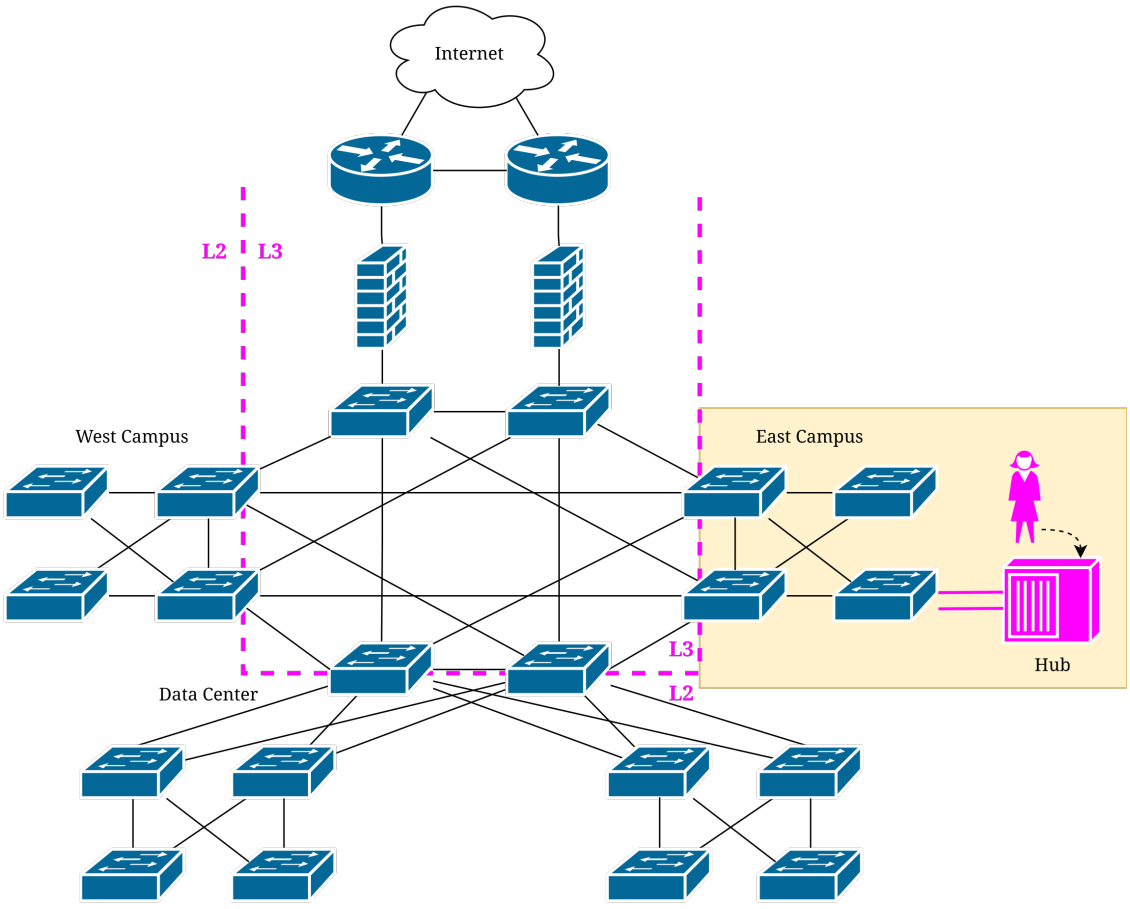

Figure 5: Higher Education Campus Architecture New Failure Domain

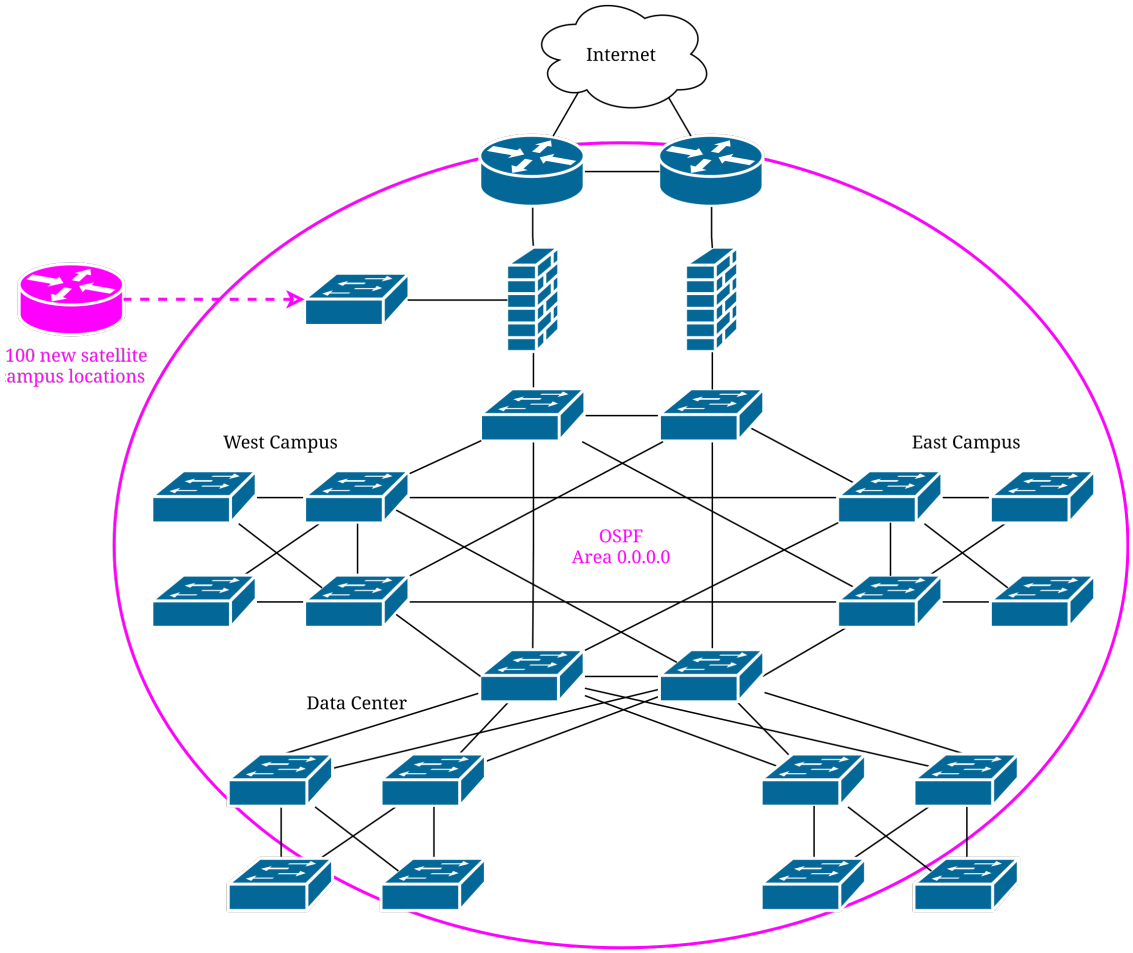

Figure 6: Higher Education Campus Architecture Flat OSPF Area 0 Design

- The main routers in the Internet section would be the area 0 demarcation
- Each site in its own totally stubby area (only a default route propagated into the area)
- Internet site in its own NSSA (area 20) for redistributing the default route

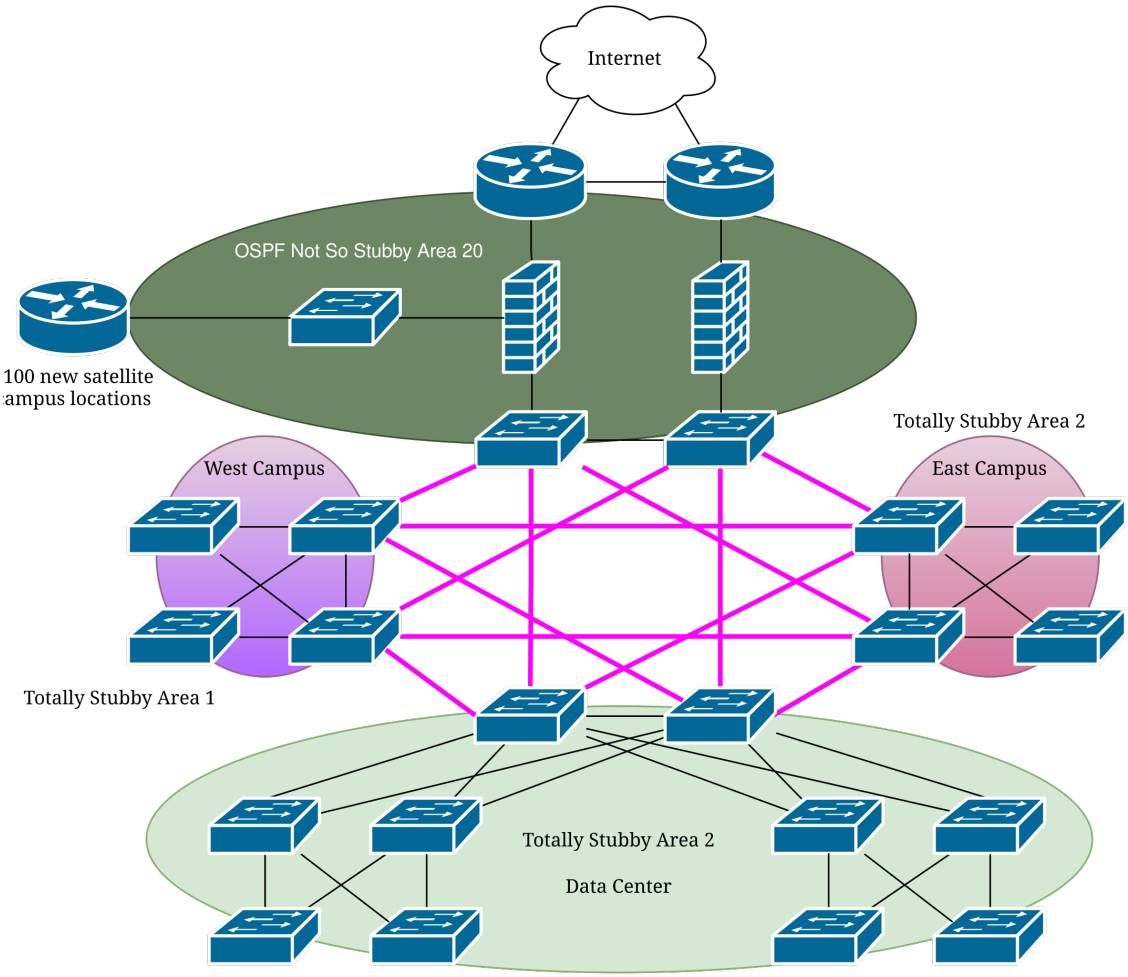

Figure 7: Higher Education Campus Architecture Multi-Area OSPF Design

4.2 Shared Failure State (Fate Sharing)

Shared failure state (fate sharing) is when a device, system, or portion of the network is filling multiple critical roles, services, and/or protocols.

Example: BGP route reflectors in the Internet pod serving as both IPv4 and IPv6 route reflectors for all campus locations (Figure 8). If an IPv4 vulnerability brought these routers down, the architecture would also lose its IPv6 BGP route reflectors, even though there was no IPv6 vulnerability.

To mitigate this shared fate scenario, break up the roles (IPv4 and IPv6 BGP route reflectors) onto dedicated stacks (Figure 9). This mitigates the shared fate scenario but increases cost and complexity. The designer must weigh the costs and benefits.

A shared fate scenario isn't always bad. It can be leveraged as a design element to ensure that when a specific function fails, the rest of the affected architecture fails over as well. In the example above, if the Internet edge fails, the entire campus architecture fails over to the backup Internet edge. If the Internet edge is up, the entire campus architecture is up. This is a shared fate scenario that can be leveraged as a design element.

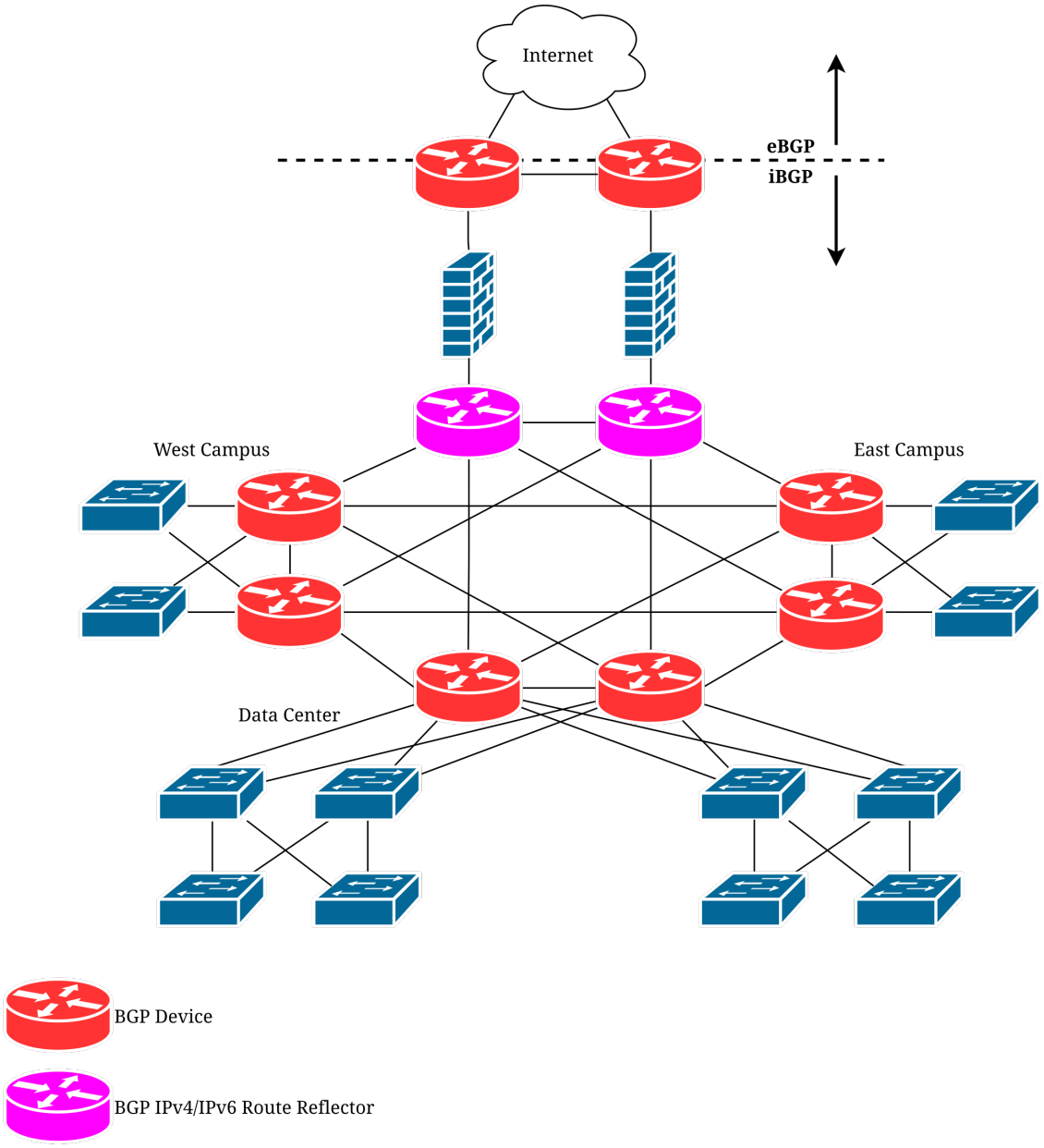

Figure 8: Higher Education Campus Architecture BGP Fate Sharing

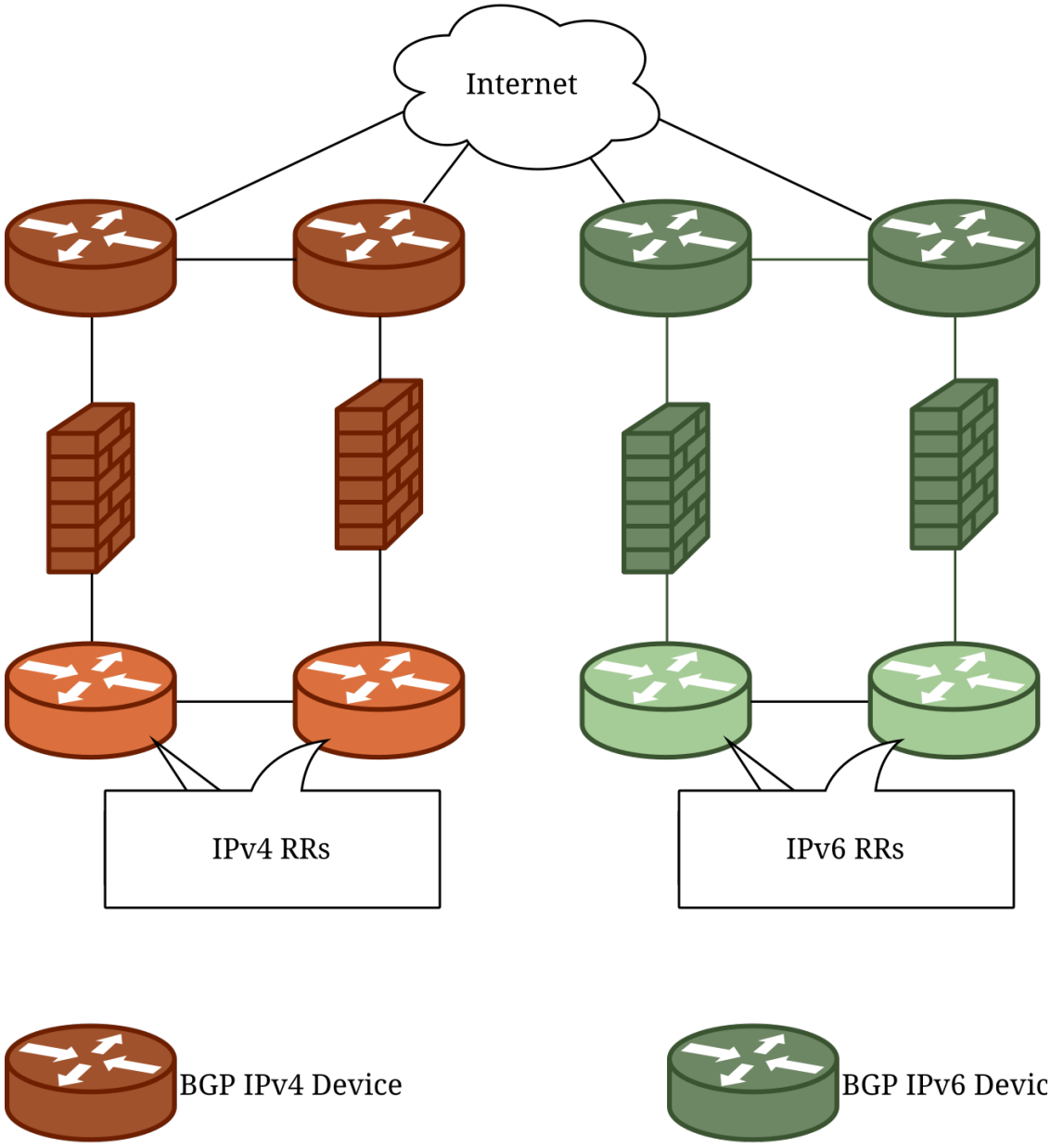

Figure 9: Higher Education Campus Architecture BGP Dedicated Internet Stacks

4.3 Modularity (Building Blocks)

Modularity is the concept of breaking design elements into functional blocks or pods, and isolating technologies and corresponding capabilities within that block. Think of purpose-built blocks.

The campus architecture already has purpose-built pods (Figure 10): Internet pod, west campus pod, east campus pod, and data center pod.

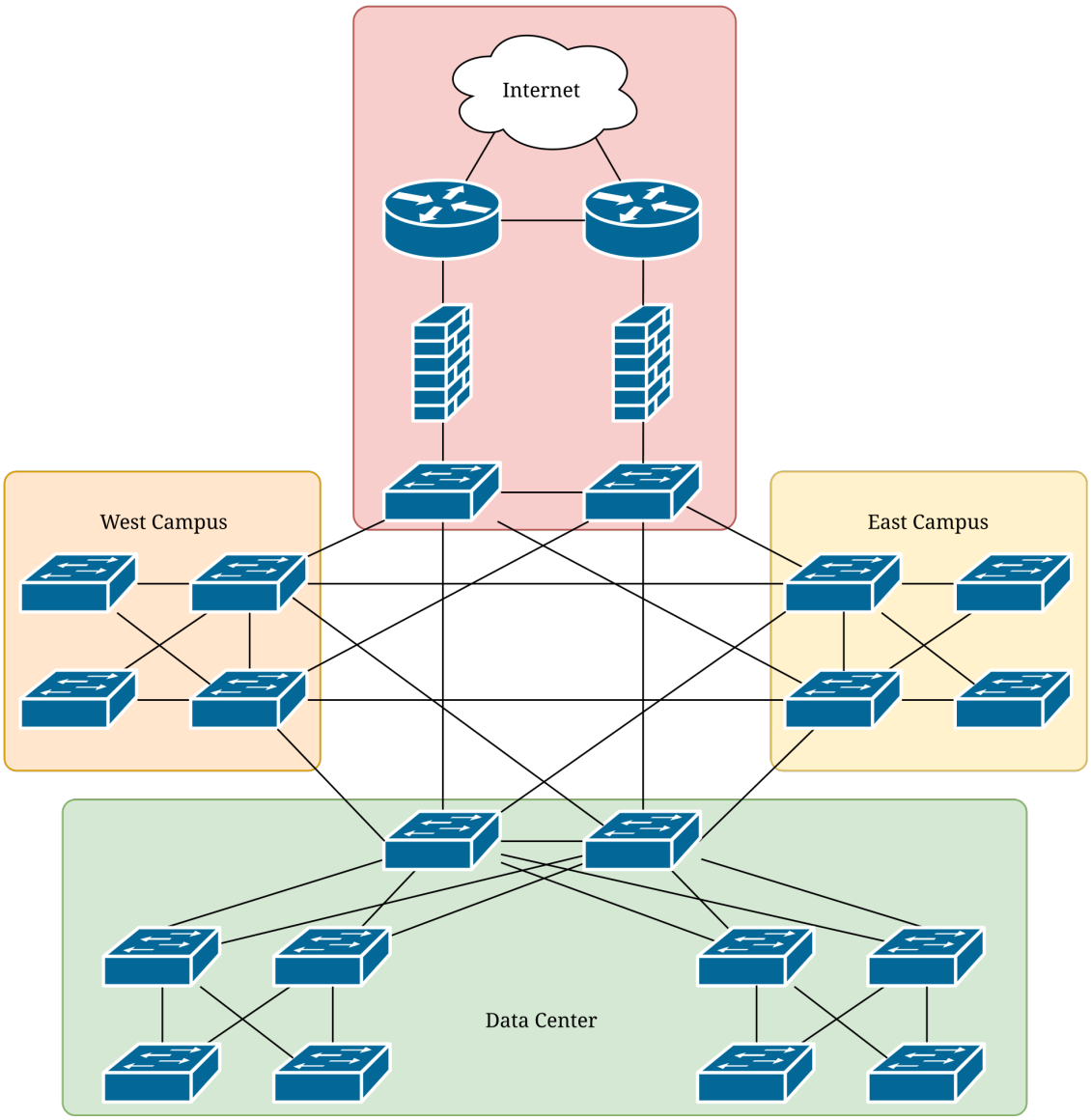

Figure 10: Higher Education Campus Architecture Purpose-Built Pods

Suppose that because of a merger with another higher education institute, these two architectures have to be merged. If this new campus requires adding north and south campus locations, a full mesh of all locations would create a large, convoluted network with enormous fiber runs (Figure 11).

Figure 11: Higher Education Campus Architecture Modularity Merged Full Mesh

A better design would be a dedicated core pod that all sites connect into, including the Internet edge pod

(Figure 12). From a scalability perspective, once the core pod hits its scale limitation, simply add a second core pod interconnected with the original.

Figure 12: Higher Education Campus Architecture Modularity Merged Core Pod

Within each pod: its own Layer 2 isolated boundary, Layer 3 mechanisms to limit failure impact, and redundant devices and links for availability.

With a modularity approach, we can set performance factors and replicate each pod architecture as the network and business grow. If a design needs a new core pod, we add it. If a design needs a new Internet edge block, we add it. If a design needs a new access block, we add it. This is repeatable, expandable, and manageable.

4.4 Hierarchy of Design

Hierarchy of design is the idea of creating dedicated levels for different purposes within the architecture. The traditional hierarchy model is access, distribution, aggregation, and core.

Five west campus locations connecting directly to the core pod is an extremely flat architecture that lacks hierarchy (Figure 13).

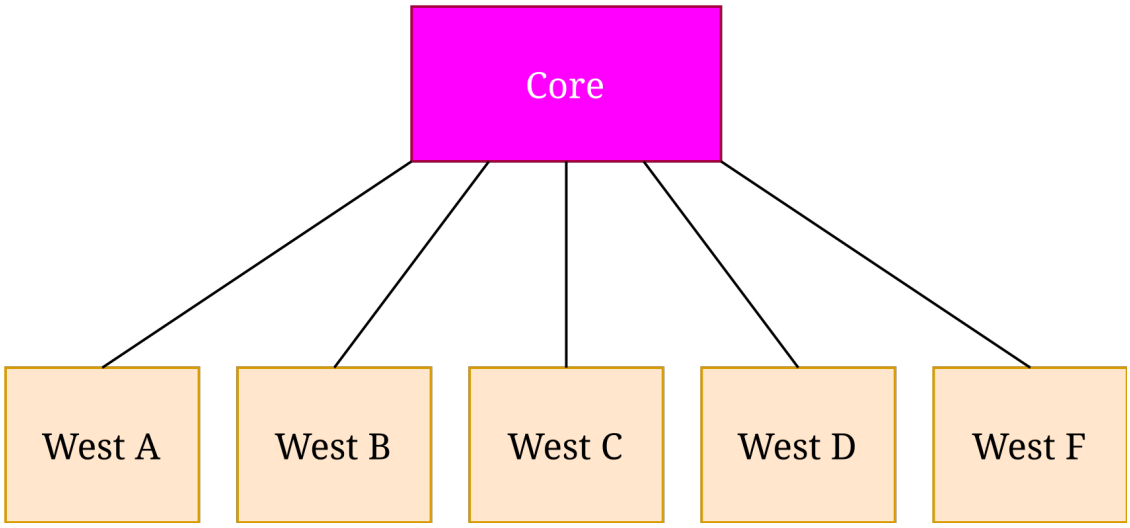

Figure 13: Higher Education Campus Architecture Flat Hierarchy

By breaking this up into distribution and access layers, we create a robust and scalable architecture. You do not need to add layers just because. Add more layers as the requirements, business, and network need them. When the time comes for a dedicated distribution, aggregation, and access layer (Figure 14).

4.5 Putting It All Together

We now have a dedicated core block interconnecting the entire campus architecture, with Internet and data center blocks, and campus blocks with different scale sizes based on their own needs and requirements. Start simple, move up in complexity based on business need.

These concepts, while simple and easy to understand, are extremely impactful from a network design perspective. As shown in Figure 15, we are left with a robust network architecture based on sound network design fundamentals, techniques, and principles.

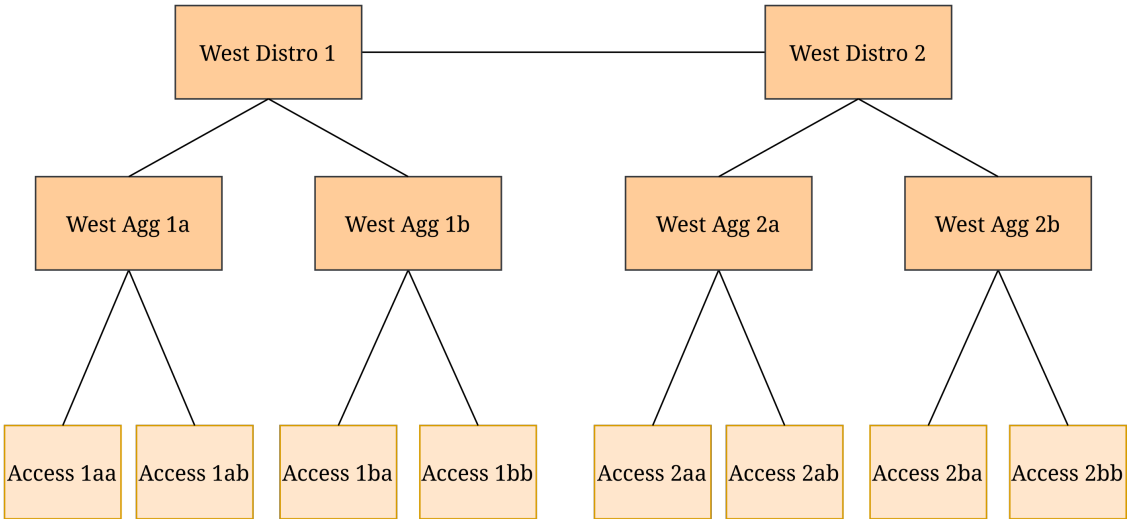

Figure 14: Higher Education Campus Architecture Distribution Hierarchy

Review Questions

7. Which network design technique allows for purpose-built building blocks to be leveraged as the business needs arise?

a. Failure isolation
b. Shared failure state
c. Modularity
d. Hierarchy

Answer

c. Modularity allows for purpose-built building blocks to be leveraged. Failure isolation is a technique that creates boundaries within the network design to help contain problems from propagating. Shared failure state is where a device is performing multiple critical functions, and if it were to incur an outage from one of those functions, it would affect the other critical functions. Hierarchy is the process of creating layers within the architecture for a specific purpose.

---

8. Which of the following network design techniques could help mitigate a Layer 2 broadcast storm from propagating from site to site?

a. Hierarchy
b. Shared failure state
c. Modularity
d. Failure isolation

Answer

d. Failure isolation is a technique that creates boundaries within the network design to help contain problems from propagating. Hierarchy is the process of creating layers within the architecture for a specific purpose. Shared failure state is where a device is performing multiple critical functions and, if it were to incur an outage from one of those functions, would affect the other critical functions. Modularity allows for purpose-built building blocks to be leveraged.

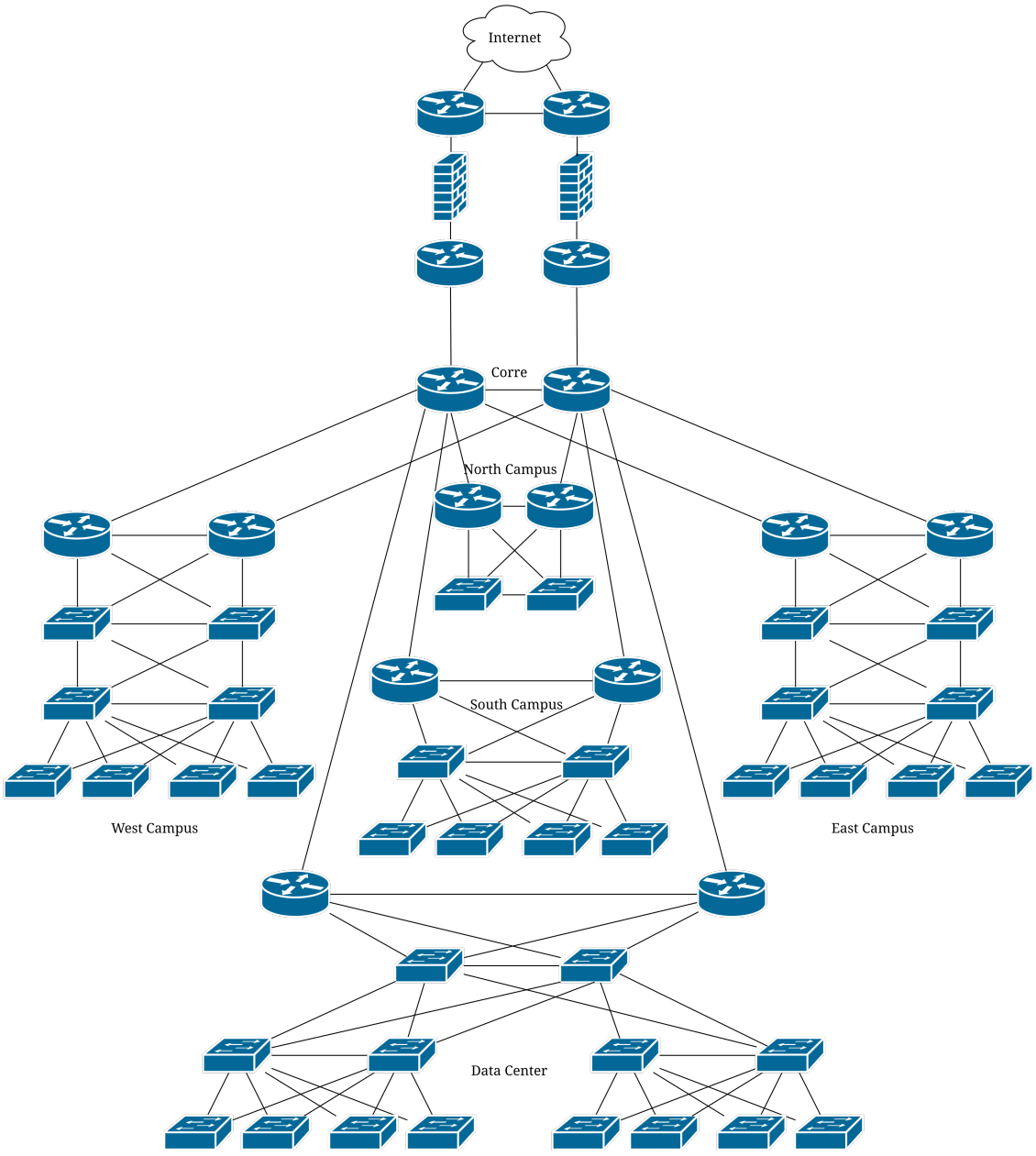

Figure 15: Higher Education Campus Architecture After Putting It All Together

9. Which of the following options leverages access, aggregation, distribution, and core layers to help structure a network design properly?

a. Modularity
b. Failure isolation
c. Hierarchy
d. Shared failure state

Answer

c. Hierarchy is the process of creating layers within the architecture for a specific purpose. The most common layers are core, distribution, aggregation, and access. Modularity allows for purpose-built building blocks to be leveraged. Failure isolation is a technique that creates boundaries within the network design to help contain problems from propagating. Shared failure state is where a device is performing multiple critical functions and, if it were to incur an outage from one of those functions, would affect the other critical functions.

5 Network Design Pitfalls

Network design pitfalls are the most common mistakes that can literally break any network design you create or any design decision you make.

5.1 Making Assumptions

In any design situation, you never want to assume anything. You can have assumptions, but don't make any design decisions based on an assumption until you validate that it is correct.

Validate your assumptions with questions to your customer, the business stakeholders, and your team. Leverage and lead workshops to drive customer dialog, but let them talk. Listen attentively, process what is being discussed, and take notes.

Don't assume complexity is bad. Don't assume the customer has a limited budget. Don't assume the customer isn't open to new technologies. If you do assume, validate before making any design decisions. Map everything back to the business and the requirements. You might discover that the customer's needs require a complex solution, and that is perfectly acceptable.

5.2 Overdesigning (Gold Plating)

Overdesigning (gold plating) is adding bells and whistles to a network design that are not needed, often under the assumption it will impress the customer, enhance job security, or future-proof the network.

Example: you are tasked with eliminating single points of failure, but you decide on your own to also eliminate dual points of failure. The solution has exponential complexity and resource cost, and is more likely to aggravate your customer than please them.

More subtle examples:

- Receiving the full IPv4 Internet table on Internet edge routers when there is no business need or requirement to do so
- Designing MPLS-TE tunnels with sub-second failover, node protection, and link protection when IGP timers would meet the requirements, adding a ton of complexity and daily management overhead
- Putting a technology into a production network just to see how it works, with no requirement for it

The easiest way to mitigate overdesigning is to hyperfocus on the requirements. Everything you do should have a direct business requirement that it maps to. This won't be a one-to-one mapping; it will be a one-to-many mapping: the business requirement mapped to many design decisions.

5.3 Best Practices

We cannot fall into the trap of "best practices." Instead, take best practices into consideration and modify them for each design decision based on the business requirements. Just because something is best practice doesn't mean it's going to work that way in your design.

Simple example: why do we enable an OSPF interface as point-to-point? Is there a business requirement for it? Or are we breaking a business requirement because of this "best practice" decision?

Complex example: implementing sub-second failover for an IGP versus less than 5 seconds failover. Are we implementing sub-second failover because it's "best practice" or because it correlates to a business need?

Sometimes you have to design a network that is not preferred from a best practice standpoint because the requirements demand it. Example: spanning Layer 2 via STP between two data centers creates a large failure domain, and most designers wouldn't want to do it. But application requirements (Layer 2 connectivity, hard-coded IPs) can force this design option.

If there are no relevant business requirements for a specific design situation and you are not violating another business requirement, then best practice is probably the way to go, but you need to understand the full picture before making these decisions.

5.4 Preconceived Notions

Preconceived notions are similar to assumptions but are defined by outside information from your experiences.

Just because a network designer likes EIGRP does not mean it's the correct IGP for every design situation. Just because MPLS L3VPN circuits are more expensive than MPLS L2VPN circuits in your experience does not mean you can make decisions based on that information in a design situation. The design should always be tied back to the customer's business requirements.

Review Questions

10. Overdesigning is one of the most common network design pitfalls; what's another phrase that means the same thing?

a. Gold plating
b. Preconceived notions
c. Best practices
d. Overthinking

Answer

a. Gold plating is the process of adding design elements that are excessive and do not meet any underlying requirement. Preconceived notions is when you as a network designer bring in outside information and assign that information as attributes to make design decisions, even though they do not apply to the situation in question. Best practices are the general list of configurations, features, and functions that should be deployed when there are no requirements governing, limiting, and restricting said item. Overthinking is a distractor option.

6 Summary

The network design elements covered in this chapter:

- **Network design fundamentals** (the foundation of a house): mindset, requirements, design use cases, the business, constraints, and "why"
- **Network design principles** (the framing of the house): security, scalability, availability, cost, manageability, and the give-and-take between them
- **Network design techniques** (the roof of the house): failure isolation, fate sharing, modularity, and hierarchy, demonstrated through a real-world higher education campus case study
- **Network design pitfalls**: assumptions, overdesigning, strict adherence to best practices, and preconceived notions

It really boils down to doing what is right for the specific situation that you are presented with. There is no "one fits all" solution.