White Paper
Data Center
July 2022

# IT@Intel: Scaling Intel's Data Centers with Software-Defined Networking and Automation

## Intel IT has chosen an open, standardized approach to software-defined networking


### Intel IT Authors

**Sanjay Rungta**
Senior Principal Engineer

**Greg Botts**
Senior Network Engineer

**Chris Brinkmeyer**
Staff Network Engineer

**Matthew Gray**
Automation Engineer

**Brad Horner**
Senior Network Engineer


### Table of Contents


Executive Summary ..... 1
Background ..... 2
Selecting an SDN Approach and Architecture Components ..... 2
Improving Scalability by Adopting a Leaf-Spine Network Architecture ..... 3
Strategy for a Scalable, Robust SDN Architecture ..... 3
Solution Architecture ..... 5
Results ..... 9
Conclusion ..... 10
Related Content ..... 10


## Executive Summary


As Intel's business grows, demand for data center network capacity has increased by more than 25% annually. Additionally, business pressures require new capacity to be brought into production within 24 hours. As far back as 2014, we recognized the potential of software-defined networking (SDN) to help meet these challenges.

After evaluating SDN components and architectures, we selected an open, standards-based architecture instead of a supplier-centric solution. As our SDN architecture has matured, we have developed a standardized and scalable data center network architecture that takes advantage of automation. The open interface allows flexibility to integrate additional business-driven automation to meet our growth and timeline needs.

Our network architecture strategy relies on five pillars:

- **Scalability through standardization.** Maintain consistent switch hardware and OS, with strict naming conventions, topology, configurations and solutions to enable automation and rapid scalability at large data centers.
- **Programmability.** Allow our workforce to adapt to significant growth of network scale at improved velocity. It also enables full lifecycle provisioning of network infrastructure from Day 0 to end of life.
- **Security.** Ability to segment the network over common infrastructure to support different use cases and enhance data center security.
- **Resiliency.** Support continuous operations of network functionality, rapid recovery and the ability to maintain functionality in an impacted state.
- **Supportability.** Maintain the designed level of performance and availability of the network. Standards lead the way to improved troubleshooting.

Over the last three years, we migrated most of our data centers to a new SDN architecture that uses a leaf-spine underlay with overlay networks. Industry-standard components and protocols have enabled us to improve network delivery time with fewer human resources, thereby increasing overall efficiency. We have improved the stability and reliability of the network and consolidated multiple dedicated customer networks onto common infrastructure with enhanced security controls.

## IT@Intel Contributor

**Mario Villalta,** Industry Engagement Manager


## Acronyms

| | |
|---|---|
| **ACL** | access control list |
| **ASN** | autonomous system number |
| **BGP** | Border Gateway Protocol |
| **DHCP** | Dynamic Host Configuration Protocol |
| **DNS** | Domain Name System |
| **DOME** | Design, Office, Manufacturing and Enterprise |
| **eBGP** | External Border Gateway Protocol |
| **EVPN** | Ethernet Virtual Private Network |
| **HPC** | high-performance computing |
| **LACP** | Link Aggregation Control Protocol |
| **MLAG** | Multi-Chassis Link Aggregation Grouping |
| **POD** | point of delivery |
| **SDN** | software-defined networking |
| **STP** | Spanning Tree Protocol |
| **TOR** | top-of-rack |
| **VTEP** | Virtual Tunnel End Point |
| **VRF** | virtual routing and forwarding |
| **VNI** | VxLAN Network ID |
| **VxLAN** | Virtual Extensible Local Area Network |
| **WSGI** | Web Server Gateway Interface |
| **ZTP** | zero-touch provisioning |

## Background

Intel's data centers are the heart of a thriving, complex business. Intel IT operates 56 data center modules at 16 data center sites. These sites have a total capacity of 102 MW, housing more than 360,000 servers that underpin the computing needs of more than 116,000 employees. To support the business needs of Intel's critical business functions — Design, Office, Manufacturing and Enterprise (DOME) — while operating our data centers as efficiently as possible, Intel IT has engaged in data center network modernization since 2019. Intel's business is becoming increasingly data-driven, relying on machine learning, AI, big data analytics and automation. As data explodes, we are experiencing greater than 25% growth in demand for network capacity every year. In parallel, we desire to put the new capacity into production within 24 hours once received to optimize the value of the investment.

In 2014, we began to evaluate software-defined networking (SDN) solutions as a way to meet these data center challenges. Until that time, traditional networking approaches using fixed-purpose hardware met the needs of client/server computing. But with the proliferation of cloud-based services and server virtualization, along with continued business growth, we needed a way to keep up with a more dynamic computing environment, and SDN offered a lot of potential. Our SDN solution provides us with an interface that enables programmatic manageability. It also offers an integrated and automated control plane, which allows us to scale while maintaining a standardized environment. The new SDN architecture is now used in three of the four DOME environments; however, the Manufacturing environment uses a different approach due to its unique business drivers.

## Selecting an SDN Approach and Architecture Components

In 2018, as we started exploring how to adopt 100 Gbps technology, we scanned the industry and SDN solutions. As the SDN market evolved, we noted that solutions tended to fall into two categories:

- Closed-loop SDN using supplier-centric technologies.
- Open, standards-based SDN that supports next-generation data center architectures featuring underlay and overlay designs.

While each approach has its advantages, we determined that developing standardized, scalable building blocks for our data center network architecture would better support the automation necessary for on-demand provisioning, self-healing and scalability. The open architecture enables us to integrate additional, business-driven automation capabilities to meet our specific requirements. Plus, it helps avoid vendor lock-in and takes advantage of a growing, evolving ecosystem.

Once we settled on an overall SDN approach, we used a scorecard that included proof of concept testing, cost analysis and scalability assessment (see Figure 1). We used this scorecard to select a new switch supplier that we could use for both the Enterprise and Design data centers. In our Enterprise and Design environment tests, we developed key criteria, including cost efficiency; product capability, architecture and openness; manageability and automation; vendor support and integration with existing design components.

Lowest 1 2 3 4 5 Highest

| | Supplier #1 | Supplier #2 | Supplier #3 |
|---|---|---|---|
| **Cost Efficiency:** Includes cost per port, solution and devices to fulfill solution. | 5 | 4 | 5 |
| **Transition:** Effort to transition environments, including education, interoperability and complexity. | 4 | 3 | 5 |
| **Product Capability:** Ability of product to provide necessary connectivity to endpoints, capacity and throughput. | 5 | 5 | 4 |
| **Product Architecture/ Openness:** Buffer per port, dedicated management plane and roadmap pipeline. | 5 | 3 | 5 |
| **Manageability and Automation:** Operations ability to maintain solution, backup, change ticketing and overall analytics | 5 | 4 | 4 |
| **Vendor Support:** Ability to support and fix quickly, as well as ability to customize the solution. | 5 | 5 | 4 |
| **Security:** Segmentation and transparent securing of the workflow. | 4 | 5 | 4 |
| **Total:** | Supplier 1 33 | Supplier 2 29 | Supplier 3 31 |

**Figure 1.** A technology scorecard helps us quantify switch supplier evaluation results for our Enterprise and Design environments.

## Improving Scalability by Adopting a Leaf-Spine Network Architecture

Traditionally, Intel's data center network architecture was implemented with a three-tier hierarchical model. This industry-standard method of connectivity consisted of Core, Distribution and Access layer switches. Using L3 protocols for routing between the Core and Distribution layer switches and L2 protocols between the Distribution layer and the Access layer switches enabled simple, intuitive deployment of services that helped increase the supportability of our critical data centers. However, this architecture could not scale well enough to support Intel's growth needs within its massive Design centers that use high-performance computing (HPC); nor could it support the growing complexity within the Enterprise data center environments. In addition, our Design and Enterprise data center network traffic experienced a significant shift from primarily north-south traffic to mostly east-west traffic. This shift caused congestion on the Core and Distribution layers. To better support the new traffic patterns and Intel's growth, we are continuing to modernize our network architecture. We are replacing the three-tier hierarchical model with a leaf-spine architecture. (See "Migration Strategy" later for our approach to transparently transitioning the network from one model to the other.)

In a leaf-spine architecture, the leaf switch is connected to multiple spine switches, which indirectly provides higher bandwidth and improved redundancy. By adopting a scalable unit of leaf and spine (also called a point of delivery, or POD), it is easy to scale the data center network using fixed-configuration switches on an as-needed basis (Figure 2).

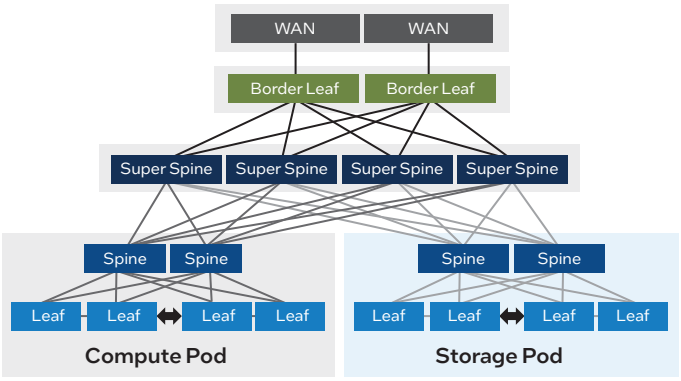


**Figure 2.** A leaf-spine network architecture better supports Intel's data centers, compared to a traditional three-tiered hierarchical network architecture.

### Network Fabric Design Details

To make the adoption and scaling of a leaf-spine architecture most efficient, we require every aspect of the architecture to have repeatable building blocks (such as point of deliveries), deterministic communication flow and solution flexibility to meet a growing number of use cases required by Intel's business units. This building-blocks approach enables large-scale deployment at an increased deployment velocity. Our network architecture is also built with strict standards and guidelines that encompass the full stack of our network.

To the fullest extent possible, we automate a switch's lifecycle from onboarding to end of life. This lifecycle automation enables transparent deployment and maintenance:

- Day 0 with zero-touch provisioning (ZTP) and onboarding.
- Day 1 configuration for fabric deployment.
- Day 2 configuration for a specific use case.
- End of life removal or decommissioning of the switch from the network.

For the underlay network, the leaf-spine design and Border Gateway Protocol (BGP) routing are critical aspects. To provide L2 mobility across the fabric and highly secure, transparent enclaves, overlay networks are built using Virtual Extensible Local Area Network (VxLAN) and BGP's Ethernet Virtual Private Network (EVPN) capabilities. To optimize the network for non-blocking network communications, we have eliminated the Spanning Tree Protocol (STP) from the network. Also, all network switches are non-blocking-capable devices; this means that the switch can carry ingress/egress network traffic at wire speed (the maximum bandwidth of the interface).

## Strategy for a Scalable, Robust SDN Architecture

Historically, our network strategy has been optimized predominantly for cost, although we also considered network performance. To better support Intel's growing business, we have redefined our network strategy to pursue technological advances to modernize and transform the network to ensure not only cost effectiveness but also best-in-class service quality. The following sections provide some details around the five pillars that underpin our data center network strategy.

### Scalability through Standardization

When we set our initial goals for SDN, we realized the solution we developed needed to be automatable and scalable both locally and globally. This necessitated a well-defined set of conventions that covered both local configuration parameters and those that would potentially have a global relevance. This was an early, critical acknowledgment. To that end, everything was designed with standardization in mind. We also constructed the documentation so it could be interpreted by developers. We embedded all configuration specifications in our architecture guide to encourage and enable automation. The documentation includes variables, input parameters and configuration outputs.

Some of the critical conventions that we defined include the following:

- **Device naming.** We implemented device naming so that the name indicates a device's location information and function. From the naming, we can derive configurations that are location-specific (such as local Domain Name System [DNS] and directory services) and function-specific (such as spine versus leaf configurations).
- **VLAN definitions and parameters.** We identified VLAN use cases and configuration parameters. Each VLAN is assigned to a security zone and carries certain attributes within the zone. Much of the automation configuration is based on this information. Over time, we have found VLAN definition to be the most dynamic network aspect, as we continually add new use cases. Our VLAN construct has been invaluable in maintaining structure within the fabric as we manage new deployments.
- **BGP autonomous system number (ASN) allocation.** We allocated ranges based on location and within each data center function (Design or Enterprise). Similar to other conventions, this allows for predictable, automatable deployments.
- **Connectivity assignments.** We pre-allocated which ports would be assigned based on device and functionality. Depending on placement within the fabric, device types were assigned along with the connectivity conventions to neighboring elements.
- **Device types and OS.** We used a limited set of certified devices in the solution to simplify spare parts inventory and device support. New devices are only added as critically needed. Sometimes this forces us to use devices that aren't a perfect fit, but the need for consistency outweighs the use of one-off device types. We minimized the device type list to help reduce the OS count and specifications that we have to test against. When we certify a new OS, we push the upgrade across the install base, which helps ensure that all features are available and perform as expected. Our approach to device types and OS use allows us to design without having to account for deployment inconsistencies.
- **Base-build configurations and security specifications.** We identified and propagated common configurations that incorporate security across all devices to help ensure stability and a common configuration to build upon.
- **Security zones.** Each identified security zone receives the appropriate and relevant conventions.
- **VxLAN Network ID (VNI) allocations.** We globalized VNI mappings with ranges pre-assigned to each data center, so when we implemented Data Center Interconnect (DCI), there were no VxLAN VNI conflicts. We could use legacy VLAN information within each data center without worrying about VLAN conflicts in other locations.

Together, these conventions enable a highly automatable and scalable deployment as well as a significant reduction in mean time to deploy (MTD) and mean time to repair (MTR).

## Programmability

Our previous network solution had limited automation capabilities. Onboarding network devices required physical touch; could be accessed only by older methods like SSH and command-line interfaces (CLIs); and had to be configured and managed individually, mostly with human intervention.

With our new SDN solution, a central controller onboards the devices and manages them from one central location. This enables us to enforce standardization, change control and have a single source of truth for our network environment.

Once we could efficiently manage our fleet of devices, we programmatically generated all the relevant configurations for them. Having the Standardization components already defined algorithmically provided us with configuration templates and the variables that would be used per-site/device. It also provided us with the algorithms for computing the values of those variables. We created Python code to compute the values and pass them into Jinja2 templates, rendering a device configuration that is complete with its specific values. See the Orchestration and Automation Framework section later for additional details.

The combination of standardization and programmability gives us consistency across the network environment, drastically reducing human error and downtime while allowing us to quickly deploy new network capabilities.

## Security

A critical aspect of the new design was to enable multi-tenant support with appropriate security at all layers over the common underlay data center IP fabric. We enabled multiple security capabilities — such as large-scale access control list (ACL), virtual routing and forwarding (VRF), traffic redirect, etc. — in the toolbox so that right tool can be used at the underlay or overlay layer to control the traffic flow. Integration with external security capabilities like a firewall was also essential. It was also critical for the security solution to scale beyond 10 Gbps performance with next-generation firewalls. Finally, we used sFlow processing in the design to keep the visibility in the environment.

## Resiliency

Our goal is to enable continuous operations of network functionality (even in the face of network failures) and rapid recovery. Here are some of the ways in which we are increasing network resiliency:

- Expanding the routing domain to create an equal-cost/multiple-path design. We are using External BGP (eBGP) to significantly scale data centers at locations with multiple availability zones (see Figure 3).

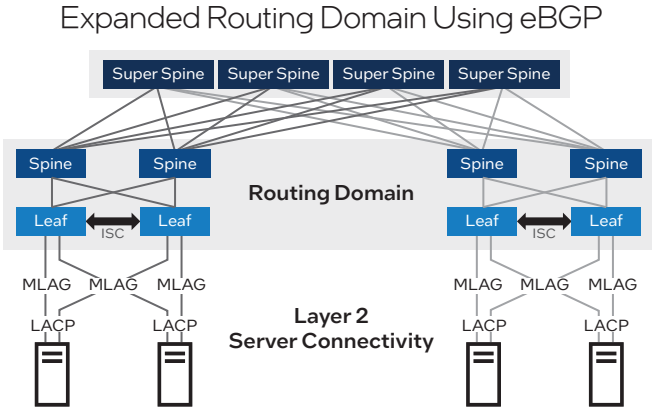


**Figure 3.** We are using eBGP to expand the routing domain, while concurrently reducing L2 connectivity to improve network resiliency.

- Reducing L2 outage domains within racks. This includes eliminating the STP to improve data throughput and using a /31 subnet mask to conserve IP address space for point-to-point links. The latter technique eliminates a port channel between the leaf and spine, which in turn eliminates the possibility of uneven load balancing (hash polarizations).
- Deploying dual home servers to increase server uptime and enabling the network team to perform maintenance without affecting customers.
- Using the standards-based Link Aggregation Control Protocol (LACP) within IEEE 802.3ad to allow the logical bundling of links, while negotiating with far-end devices to enable graceful removal of links that are not transmitting the LACP. This approach reduces cabling issues and link faults.
- Employing Multi-Chassis Link Aggregation Grouping (MLAG) to deliver system-level redundancy to servers. MLAG logically teams two switches to appear as one logical switch from the server's perspective.
- Collocating critical services such as DNS and Network Time Protocol with servers. We deployed a DNS solution in our HPC data centers to help ensure that WAN outages would not impact local data center functionality. Without communication to a DNS, all servers and services fail within the data center.
- Implementing a zero-congestion strategy. Network traffic congestion is difficult to correct quickly. Our network designs include downlink-to-uplink bandwidth ratios to avoid congestion on links.

### Supportability

The other four pillars — standardization, programmability, security and resiliency — combine to provide us with the ability to maintain the designed level of performance and availability of the network. Our use of standardization leads to reproducible configurations and designs and reduces or eliminates non-compliancy and difficult-to-support one-off designs. This in turn leads to repeatable and standards-compliant predictive troubleshooting. The result is a modern, highly automated and resilient SDN that powers Intel's digital transformation through seamless secure connectivity.

## Solution Architecture

The following sections detail some of the high-level features of our SDN architecture.

### Orchestration and Automation Framework

Comprehensive SDN at our scale is not possible without an automated management plane. We developed an automation framework that integrates with the SDN controller to drive the overall orchestration in both the Design (that is, HPC) and Enterprise data center environments. But it is important to note that although we have made great strides in network automation, this is a journey like all of IT transformation. We have pivoted as we learn, as our environment grows and as we continually optimize. We believe that adopting a spirit of continuous integration and delivery is crucial to ongoing progress.

In addition, we consistently attempt to use existing in-house platform and hosting solutions. Examples include server builds, database-as-a-service[1], Cloud Foundry application service, Ansible, in-house Git repository system, DHCP and DNS. We used these standard network services to aid in automation.

As detailed in "Selecting an SDN Approach and Architecture Components" earlier, we decided on a new switch and router platform, and began ordering them in high quantities. To quickly deploy the new equipment (more than 2,000 new switches across Enterprise and Design data centers), we knew we needed to effectively provision and manage them. The supplier offers a turn-key management/orchestration platform, with several choices ranging from supplier-provided appliances, to a VM image hosted on-site, to an as-a-service cloud instance (only recently available). We chose the VM option. This entailed buying our own servers, adding our hosting-supported OS build and installing an open-source hypervisor (KVM) to host our 17 regional orchestration clusters.

When we first deployed the orchestration platform, we were able to support our initial deployments by using short Python scripts that used Jinja2 templates and yaml seed files to enable automatic provisioning, streaming telemetry and standard configuration management. The orchestration solution provided ZTP, where a network technician can edit a DHCP scope and power on a new switch, allowing it to provision itself enough to onboard into the orchestration system. From there, our scripts, templates and yaml files could push the proper configurations and image onto the switch with just a few clicks.

However, we quickly realized we needed a source of truth for our network attributes — something API-accessible that could provide our scripts and templates with the attribute data they needed for device configuration, such as VLAN, ASNs, authentication servers and management IP. Initially, we used disparate yaml files for this purpose, but they quickly became unmanageable. We also found ourselves limited by the orchestration development environment, because we could not reference other scripts and did not have access to an integrated development environment. We were limited to simply editing siloed scripts in a browser.

To solve these issues, we used our in-house, enterprise-grade database-as-a-service (DBaaS) to provide our source-of-truth database. We moved our code that generated configurations, along with the templates they consumed, into our in-house Git repository. We built a Web Server Gateway Interface (WSGI) in our in-house Cloud Foundry environment (so we did not have app server operating systems to manage) to provide a remotely accessible backend to our orchestration controller. Next, we changed the scripts on the orchestration platform to be lightweight, rarely changing "caller" scripts that gather local device data and pass it up via API call to our WSGI. The configurations are then rendered off-box and returned to the caller script. Then, the orchestration platform deploys those configurations to the devices (see Figure 4). This solution solved our flexibility issues while still utilizing the orchestration platform.

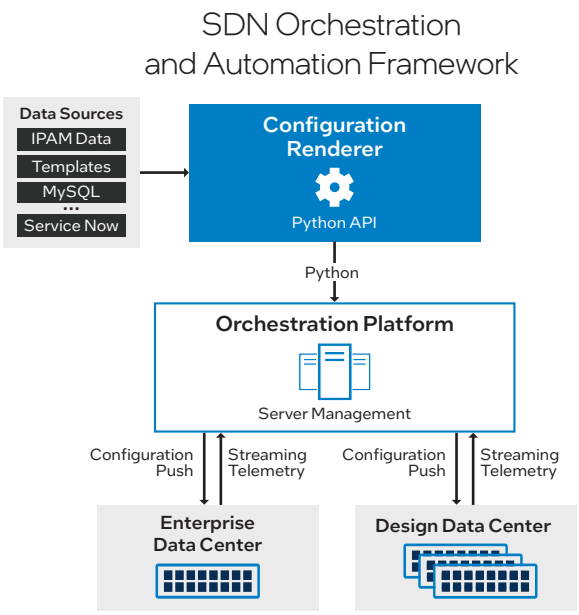


**Figure 4.** Our SDN orchestration and automation framework uses a supplier-provided management plane portal, along with in-house capabilities for network configuration data, templates, scripts and more.

While we were able to benefit from in-house hosting platforms without a fleet of app/database servers to manage, we found ourselves with a large fleet of hypervisors that were hosting our orchestration clusters. Even though we had standard builds for specific environments, we ended up with a total of 90 operating systems using three different Linux builds. We took advantage of our in-house managed Ansible platform to distribute files, perform upgrades, add monitoring agents and other tasks.

### Remaining Automation Challenges

Historically, our network teams have been focusing on pure network technology skill sets. But with SDN and automation, our teams need a mix of network technology and automation skills. As our journey continues, we need to better understand how to staff and organize teams, not just in terms of number of staff, but also by considering their skill set.

Other technology-related challenges include:

- How to develop front-end solutions to enable customers or technicians to self-service.
- Integrating more robust pre/post validation beyond what is available in the orchestration platform.

We are exploring the possibility of removing all scripts from the orchestration platform and making the entire configuration-rendering process off-box. In this scenario, the configurations would then be deployed to the orchestration platform to push to devices (using the platform's excellent built-in configuration management functionality).

## Underlay Technology

The primary goal of the underlay network is to provide a routed path for the overlay networks, so that VxLAN Virtual Tunnel End Points (VTEPs) can communicate with each other. Our overlay network is built on top of a highly redundant underlay network, using L3 point-to-point connections to build our fabric (see Figure 5).

The underlay network is documented in a VRF global table, so that the information is available to all overlay networks. We use the same dynamic routing protocol that we use for overlay networks (although other options do exist), because doing so offers the following benefits:

- Ease of management, because we are using a single protocol.
- Lower complexity due to reuse of the same autonomous system and configuration blocks.
- Ability to scale well in large topologies.
- Support for the BGP open standard. We use Interior BGP at the leaf layer (redundant L2/L3 pairs) and eBGP between spine layers (no route reflectors needed).

Our approach to the underlay network differs slightly between the Enterprise and Design data center environments.

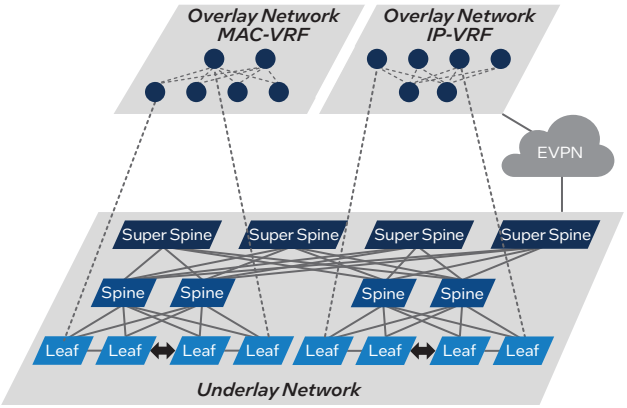


**Figure 5.** Our overlay network is based on a highly redundant underlay network.

### Enterprise Data Centers

The Enterprise data center environment had use cases for overlay networks from the beginning, so we built the underlay network with that in mind. However, initially we did not choose to expose the underlay network's global VRF table outside of each data center, which prevented us from being able to easily extend an overlay across data centers (because overlays need underlays). We have implemented a workaround to this situation, but it would have been easier if the underlay was exposed across data centers.

Using an underlay/overlay approach in the Enterprise environment enables us to extend L2 VLAN everywhere over an L3 network, so there are no more looping outages. Also, we can isolate networks using VRF tables and extend that isolation throughout the fabric.

### Design Data Centers

The Design environment is a large-scale HPC infrastructure that did not immediately have a use case for overlay networks. However, we built the leaf/spine infrastructure using the same principles and practices as we would for an underlay that was going to support an overlay:

- Direct peer-to-peer peering to physical interfaces for faster convergence.
- Equal-cost multipath to help improve latency and optimize data flow.
- All devices have loopback interfaces that are in the global default VRF table, but are not used to peer with for underlay.

Consequently, when the Design environment did have a use case for an overlay network, we already had a large underlay at our disposal.

### Remaining Underlay Challenges

One of the key underlay challenges was to scale the multiple PODs interconnected in a mega data center. At one of our large data centers, we had to implement a five-stage CLOS architecture by introducing an eight-chassis-based super-spine layer with 256 100-Gbps ports each to maintain a minimum oversubscription between the spine and super-spine layers. Initially, we started with a four-switch super-spine and then scaled it to eight to accommodate network traffic growth. To scale beyond this, in our next iteration of underlay development, we plan to introduce 400 Gbps connectivity between spine and super-spine and scale the super-spine layer horizontally when needed.

The leaf layer poses different challenges. We noticed that top-of-rack (TOR) switches with 48 or 64 ports caused a sprawl in switch count. In the next iteration, we plan to address multiple challenges at this layer: the ability to natively support 10 Gbps and 100 Gbps connectivity and the ability to use a higher number of ports on TOR switches.

## Overlay Technology

An overlay network creates a logical structure on top of the physical structure of the underlay network. In our Enterprise data centers, we needed to provide L2 mobility across the underlay fabric, while in the Design data centers, we built an L3 secure enclave overlay network. Some important attributes of our overlay networks include the following:

- VxLANs allow encapsulation for cross-site network extensions, enabling both VLAN and VRF extensions. We use the BGP's EVPN extension for dynamic VxLAN learning. We are also currently conducting a proof of concept to explore the use of static VxLAN mapping for cross-site network extensions.
- The EVPN control plane is a distributed, dynamic learning plane that is not tied to a central controller.
- Distributed L3 means that within zones, we can use anycast IPs for distributed default gateways, which helps ensure the shortest routed path between systems in the same VRF table.

Our overlay networks build enclaves, which are networking environments that operate with a common set of security controls. The demilitarized zone (DMZ) is a networking environment that buffers between discreet networking environments and consists of a VPN and Proxy environments. Typically, one of these is untrusted, which usually is the internet. Then, we define separate security zones or enclaves (see Figure 6) for external and secure internal hosting (also called a secure internal zone, or SIZ). The enclaves include backup and recovery networks, pocket networks (dedicated network environments that are application-purpose-built and protected by a firewall) and network management.

### Enterprise Data Centers

Because the Enterprise domain covers a wide variety of use cases, each with its own set of security requirements, we deployed enclaves in our Enterprise data centers from the beginning. For internally hosted applications, over time we created multiple security zones to isolate components of two-tier and three-tier applications. In some cases, we also created specific application-level enclaves. Typically, we create separate security zones for each internet-facing service and application.

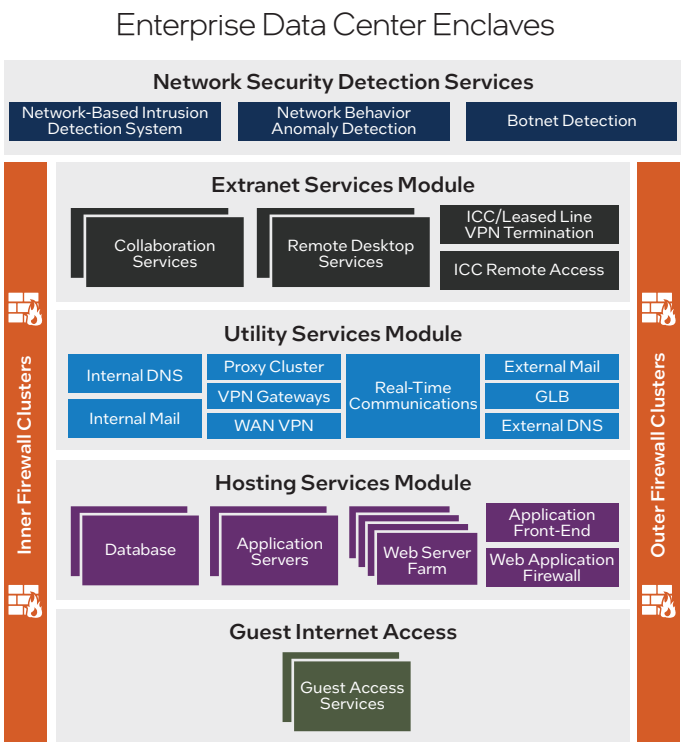


**Figure 6.** We use enclaves (indicated by the colored boxes), which are networking environments that operate with a common set of security controls, to increase security posture.

### Design Data Centers

Our Design environment originally had no need for overlay networks. But as new use cases were introduced, we needed to move beyond router ACLs to adding enclaves with special security features. In particular, adding next-generation firewall-grade security for select HPC networks was challenging, labor-intensive and often took weeks to complete because of the need for separate network switches and dedicated racks. We resolved these issues by creating a solution that is portable, granular and scalable:

- **Portable.** The ability to provide security to disparate, existing networks as well as new networks and be location-independent within a given data center.
- **Granular.** The ability to run select subnets through the firewall while letting others bypass.
- **Scalable.** The ability to support multiple tenants with low configuration overhead, where the security posture is handled by the Information Security Team (not by the network team).

Our solution starts at the leaf, where we use a VRF construct for our segmentation, providing security by routing (or lack thereof). All subnets inside the VRF can freely talk to each other, but cannot talk to anything outside the VRF. This solves the segmentation on the leaf, but the VRF isolation is only locally significant. The next component of our solution involved extending that VRF across the data center to wherever our firewalls were located, often several hops away. We used VxLAN to extend the VRF and used EVPN for the controller. Then, the service leaf pair that was connected to the firewall could serve as the VTEP, decapsulating the VRF traffic and sending it to the firewall policy for processing.

The HPC security solution provides the following benefits:

- We can use an app that spans multiple subnets, in multiple physical areas of the data center, and all these subnets can be in the same isolation bucket (VRF). The subnets can talk to each other without having to go through the firewall, but any other traffic in/out of that bucket must traverse the firewall policy.
- We are able to extend any enclave or secure network throughout the data center; there is no need to move enclosures.
- We have improved provisioning time; now it takes only two hours instead of eight days to secure the network.
- All security happens at the firewall with enhanced monitoring and logging.
- There is no impact on non-secure network traffic flow.

We are taking a phased approach to implementing this new security solution. We are starting with a single Design data center, using the design for all new enclaves and gradually migrating existing enclaves over the next year. We will then extend the solution to additional Design data centers as needed.

### Remaining Overlay Challenges

As we evolve from a monolithic security model including ACLs to a distributed security model using VRF, we plan to introduce additional overlay networks to support more security use cases over the common infrastructure. With the new L2 extension capability, we plan to support data center extension across the WAN for certain use cases.

## Integrating Intel® Silicon Photonics into Our Data Centers

### A Key Component of Intel IT's Data Center Strategy Is Network Innovation

In 2020, we evaluated Intel® Silicon Photonics as a way to move our data centers toward 100 Gbps and beyond. When we compared Intel Silicon Photonics to conventional optics-based technologies: We found that it uses less power (only 3.5 watts)[2], and is less expensive.[3]

Our adoption of Intel Silicon Photonics helps improve network utilization, reduces costs per port and enhances overall data center efficiency. We have reduced the costs associated with all network components (physical cabling, active switch equipment and optics), which has helped us to lower the overall cost of transitioning to 100 Gbps. What's more, we are poised to break new ground with even faster network technology. We are ready to meet the data explosion head-on and satisfy Intel's demand for data processing for the foreseeable future.

## Migration Strategy

To move our Enterprise and HPC data centers to the new SDN architecture, we built the new IP fabric in parallel in the data center. Any new systems were deployed directly to the new fabric while we began migrating existing racks a few at a time to the new fabric. For the HPC environment, there was no downtime for a compute rack move, while file server migration was done without downtime also by working closely with the file server administrator team. In Enterprise data centers, we used quarterly scheduled downtime to migrate L3, firewalls and load balancers to the new fabric (only one period of downtime per data center), and migrated one row at a time to the new fabric. We found that it takes six to eight hours of downtime for large data centers and three to four hours for medium data centers for the enterprise migration. Figures 7 and 8 on the next page, respectively, show our migration strategy for the Enterprise and Design data centers. Since the data centers have a low tolerance for outage windows, we adopted a phased migration approach, where layers from the legacy environment are removed first, with the client connections still intact. Subsequent phases involve staging redundant connections to the newly built infrastructure and then simultaneously cutting the links to the legacy environment while bringing up the new links.

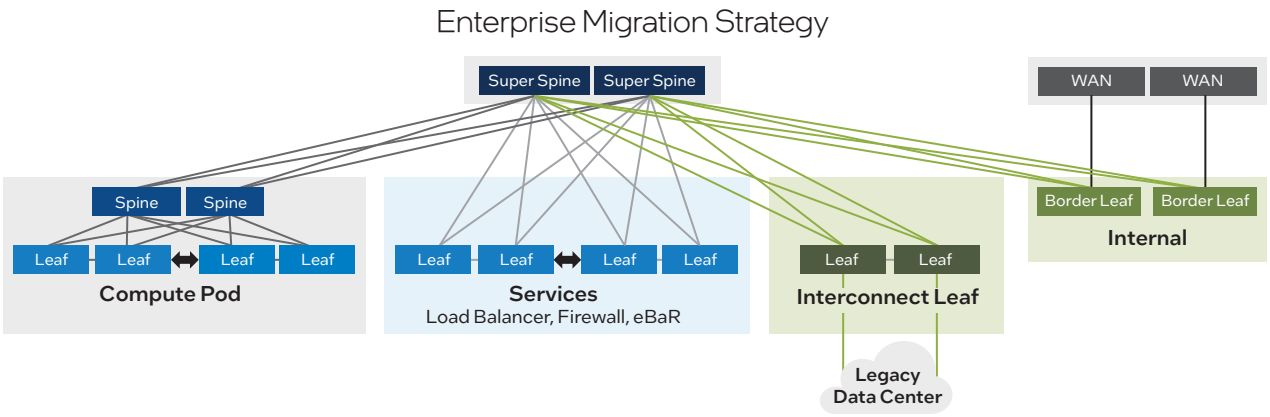


**Figure 7.** Interconnect leaf layer connecting legacy data center with new fabric.

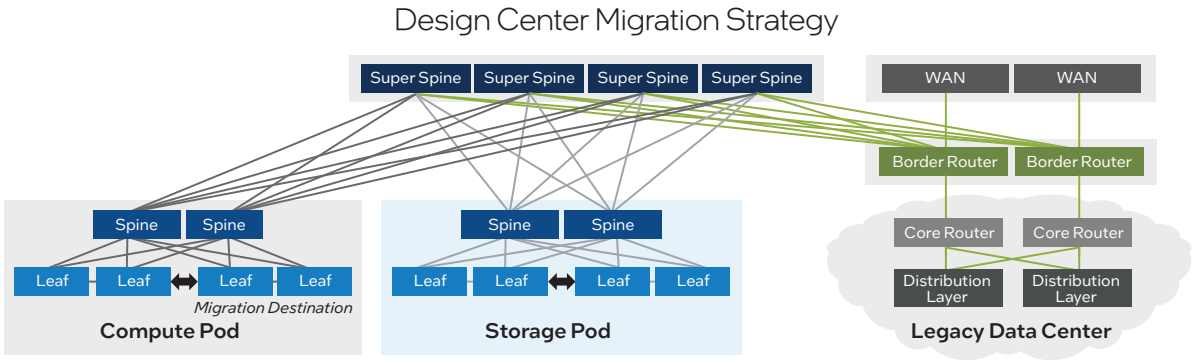


**Figure 8.** Border router connecting the old topology with the new topology.

## Results

Our adoption of SDN and automation architecture has provided numerous benefits to Intel:

- **Network provisioning improvement.** It used to take nearly eight hours to provision networks for entire racks of servers from TOR switches. Multiple manual components contributed to the long lead time, including initial switch standard software and baseline configuration, provisioning of L2 and L3 networks, configurations of L2 and L3 redundancy, setup of DNS records and, finally, giving the correct persona to a switch. With the new software-defined and automated architecture, all these components are built as part of baseline configurations and integration with IP address management, which has reduced the provisioning time to less than two hours.
- **Improvement in reliability and stability.** In the last two years, we have improved the reliability of the data center and reduced the number of performance-related incidents by 70%. Multiple factors contributed to these improvements, such as using multiple 100 Gbps connectivity (which increased the bandwidth by 2-8x) and standard deployment automation that eliminates human errors in configuration. In 2021, we had over nine months without any network-caused incident issues across all of Intel's data centers worldwide — amply illustrating the robustness of the solution.
- **Efficiency.** In the last two years, we saw 25% year-over-year growth in Intel's Design data centers. Our network team was able to support this higher volume of work without increasing staff. This was only possible due to the direct value of SDN and automating Day 0 and Day 1 tasks. We have achieved greater than 20% efficiency improvements with the SDN architecture to date.
- **Flexibility.** An additional benefit of the open, standards-based SDN and orchestration layer is the ability to add custom network layers to meet unique business requirements. In contrast, a closed-loop, supplier-centric SDN solution offers very limited ability to make these types of changes. Over the years, we have made multiple value-add changes to the automation to adjust to the architecture changes in the data center.

Over the next 18 months, we plan to finish migrating the remaining 40% of the Access network of our data centers to the new leaf-spine architecture to fully realize the value of the design and SDN.

## Conclusion

We believe our choice of open, standards-based technologies to build underlay and overlay networks with an orchestration layer has been critical in providing us with maximum flexibility to adapt to business needs and realize the value of a larger ecosystem. Our network architecture and strategy are intentionally created and data-driven to help provide the performance levels and network availability that our customers require to be successful.

The leaf-spine-based underlay architecture with open, standards-based protocols and an SDN environment allows us to fulfill the 25% annual network growth with a 4x reduction in provisioning time. We can converge separate security use cases on a common infrastructure and are able to onboard new security use cases with minimum additional effort. The backbone for all these activities was to holistically build automation and standardize the data center architecture elements so that they can easily be reproduced in building blocks.

## Related Content

If you liked this paper, you may also be interested in these related stories:

- Affordably Increase Network Bandwidth at 100 Gbps and Beyond brief
- Intel IT's Multi-Cloud Strategy: Focused on the Business white paper
- Building a Multi-Cloud-Ready Enterprise Network white paper
- Adopting Software-Defined Networking in the Enterprise white paper
- Intel IT's Software-Defined Infrastructure Experience podcast
- Preparing Intel's Data Center Network Architecture for 100 GbE podcast

For more information on Intel IT best practices, visit intel.com/IT.

## IT@Intel

We connect IT professionals with their IT peers inside Intel. Our IT department solves some of today's most demanding and complex technology issues, and we want to share these lessons directly with our fellow IT professionals in an open peer-to-peer forum.

Our goal is simple: improve efficiency throughout the organization and enhance the business value of IT investments.

Follow us and join the conversation on **Twitter** or **LinkedIn**. Visit us today at **intel.com/IT** if you would like to learn more.

[1] IT@Intel, "Increase Business Velocity with Enterprise Database as a Service," intel.com/content/www/us/en/it-management/intel-it-best-practices/increase-business-velocity-with-enterprise-dbaas-paper.html.

[2] Intel® Silicon Photonics 100G CWDM4 Brief," intel.com/content/www/us/en/architecture-and-technology/silicon-photonics/optical-transceiver-100g-cwdm4-qsfp28-brief.html.

[3] Based on internal Intel IT measurements.