White paper

Cisco confidential

The bridge to possible

# Cisco 8000 Powered By Cisco Silicon One: Foundation for Success

# Contents

## Introduction

The paradigm of today's networks is based on purpose-built devices designed for individual use cases. Because each role has its own unique requirements and organizational structures that were created around these deployments, each team solved for their specific requirements. This created unique Network Processing Units (NPU), unique Network Operating System (NOS), Software Developers' Kit (SDK), and buffer memory profiles. This role-based specialization now contributes to its own set of problems:

- Multiple operating systems and platforms make configuring and managing end-to-end networks complex.
- Multiple SDK/NPU combinations over different platforms mean any end-to-end automation is more expensive to develop, implement, and troubleshoot.
- Operational teams spend more time servicing the technical debt of disparate platforms than developing an efficient and agile network.

Network operators within hyperscalers, service providers, and enterprises see these limitations manifest in increased operational cost, the need for larger and more skilled workforces, and inflated project timelines.

Hyperscalers, in particular, have blazed the trail in maximizing operational efficiency by not just considering Capital Expenditure (CapEx) of procuring network equipment, but by recognizing that they would incur far greater costs in Operational Expenditure (OpEx)...unless they embraced operational fit as a top requirement for network devices. To that end, these operators have consistently standardized on common hardware/software architectures where possible – enabling simplified operations – to drive their multi-year network architectures. These efforts began in data centers but have spread beyond to other parts of the network infrastructure.

However, even these efforts remain confined to independent network domains – like DC Fabric or WAN – because the industry lacked a silicon architecture that could span their full network, forcing them to adopt unique architectures for specific roles.

In response to this role-based fragmentation, Cisco asked the question, "What if the same silicon architecture with a common SDK could span across the network?" Such an architecture would have to include:

- A radical rescoping of the solution from role-based silicon to a single silicon architecture extensible to all roles.
- Hardware flexibility in speed, scale, and buffering, plus expansive programmability for features.
- Support for a single SDK across all roles.

From this inquiry, the Cisco Silicon One architecture was born.

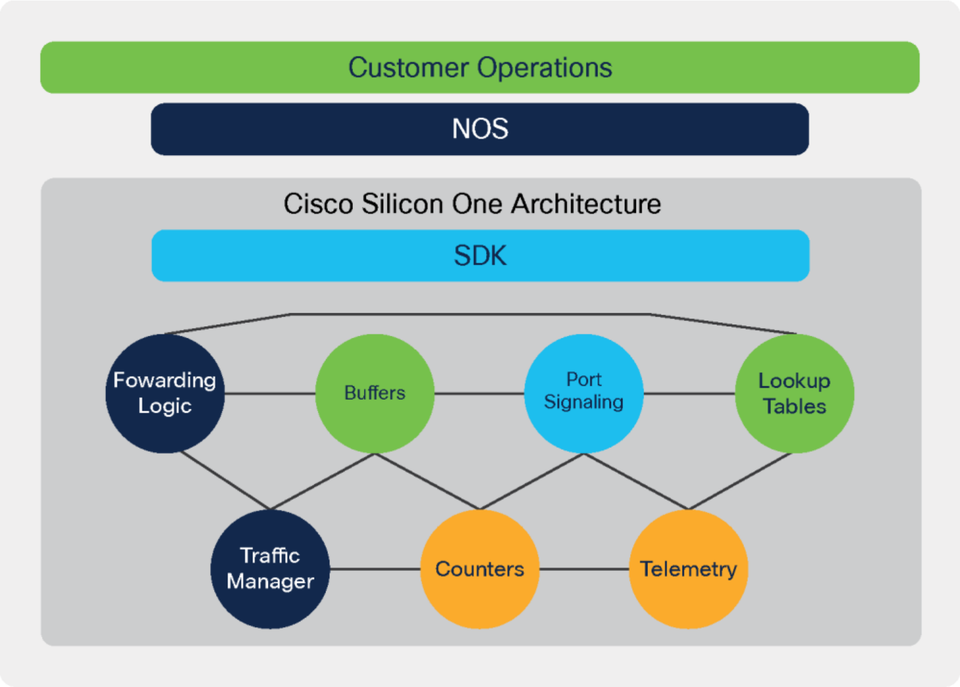

**Figure 1.**
Operational Hierarchy

It would be remiss to oversimplify the challenge that was taken on with Cisco Silicon One. The role-based architecture fragmentation derives from both organizational alignment in the market, but also deep technical challenges – both of which would need to be overcome to bring the reality of Cisco Silicon One to market.

It's also important to realize that although convergence is a strong offering, it is not enough by itself – the individual devices must also be 'best-of-breed'. This combination represents the value proposition provided by the Cisco 8000 Series: **Convergence without Compromise.**

This document starts by articulating how a single architecture leads to a more efficient operational model. Then, it will take a deep dive into the limitations of current architectures for evolving networks and how Cisco Silicon One radically addresses those limitations.

## Simplicity

Infrastructure operations involve constant iteration through a process of planning, developing, and deploying. A measurement and feedback loop – both qualitative and quantitative – informs each of these steps to make subsequent iterations more capable and more efficient. When operating at scale, commonalities across devices can make a substantial difference: a common operating system, common management interfaces, and common silicon architecture powering the infrastructure.

At the heart of this never-ending exercise is the silicon that powers the infrastructure, as show in Figure 1.

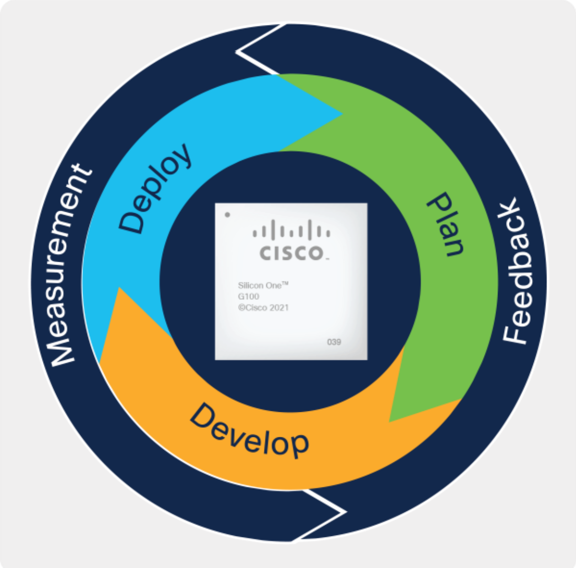

**Figure 2.**
Iterative Operational Model

Obviously, a number of factors – differing requirements, financial incentives, workforce experience, organizational realities – ensure that a single platform deployed across the network ("God box") can never exist. To advocate otherwise is folly. Yet, striving for consistency in a network through silicon architecture in as many roles as possible demonstrably delivers greater operational efficiency. Therefore, let's examine each step of the cycle in more detail and understand how systems powered by a common silicon architecture – like Cisco Silicon One – enables that efficiency.

## Planning

When deploying and maintaining infrastructure at scale, simplicity and homogeneity are the ruling principles. Of course, these traits always involve tradeoffs that must be clearly understood and accommodated based on the requirements of the applications, especially capacity and resiliency. Incorporating more systems with unique architectures makes that exercise dramatically more complex. Even different families from the same vendor have substantive differences across forwarding silicon that can make standardization challenging.

And let's not forget that long-term architectural planning to build networks includes the entire lifecycle: procurement, deployment, management, maintenance, and retirement. Each of those steps depends on a reliable supply chain, and Gartner has declared Cisco's supply chain to be the best in the world[1]. Whether you're buying components, full systems, or something in between, you benefit from the same award-winning supply chain.

## Development

You've chosen a silicon architecture and corresponding network design that accommodates your architectural goals around capacity, resiliency, and flexibility. Now it's time to develop software – whether NOS, automation, or tooling – to realize those goals. You want to maximize efficiency through consistency, and homogeneity is "The Easy Button".

But what do we mean by "consistency"? We're alluding to the efficiency of your operations strategy – the ways you create software and processes to deploy and manage your infrastructure. In any operations strategy, we can identify several attributes of a silicon architecture where consistency has a significant impact on operational overhead, and therefore efficiency:

**HW telemetry**

- Counters: Can these be correlated to pinpoint packet loss?
- Buffer Monitoring: Can you tell when you have congestion?
- Resource Utilization: Are forwarding or filtering tables increasing at a manageable pace?
- Inline Monitoring: Can you diagnose problems by embedding telemetry in packets?

**Traffic flow**

- Hashing: Are you getting efficient load balancing?
- Programmatic Steering: Can you influence traffic flow through an API?
- Forwarding Tables: Can you understand exactly how a packet will traverse the network?
- Life of a packet, including QoS behavior: Are you getting consistent performance?

[1] https://www.gartner.com/en/newsroom/press-releases/2021-05-19-gartner-announces-rankings-of-the-2021-supply-chain-top-25

**Provisioning**

- HW Feature Configurations: Are there special modes for certain data plane flows?
- Resource Carving: Do you have to pick between one feature and another?

Each of these items requires specific APIs, unique algorithms, and customized serviceability for every silicon architecture in your infrastructure. Plus, the process of developing software never really ends. New architectures within new roles with new features are constantly rolling out. And let's be honest: there's always technical debt because there are never enough people with enough resources to do all the work.

With the Cisco 8000 Series, you can minimize technical debt through fewer silicon architectures in more network roles. As a bonus, you can be confident in a long lifespan and diverse set of role adaptations.

## Deployment

In the end, you're balancing a lot of variables in your infrastructure. Cost. Security. Power. Uptime. At times, when juggling so many competing requirements, it feels like you're trapped in the old maxim: "Fast, cheap, or good. Pick two."

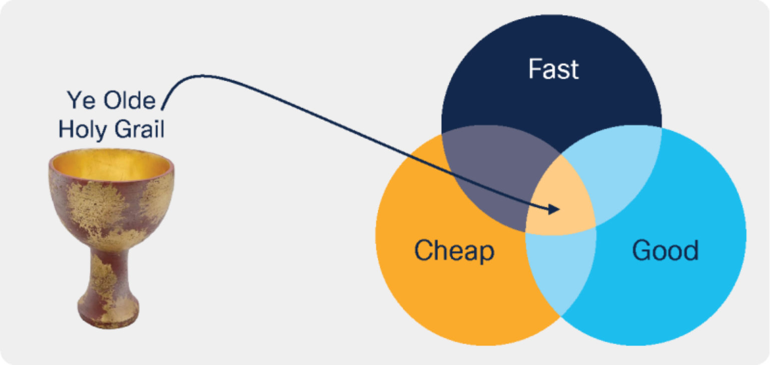

**Figure 3.**
Implementation Convergence

Every technology problem tends to involve these tradeoffs, but the implications differ. In terms of operational excellence, there are a few key variables to think about:

- How large can your operations workforce be?
- How long will it take to adopt a new platform?
- How much does the whole solution cost?

Some of the tradeoffs may be obvious, but it's useful to consider them. For example, adopting unique platforms with unique silicon families for each role in the network may provide some hardware cost benefit, but if **all** the tooling and testing needs to be redone for each, the cost savings quickly evaporate.

In a related vein, maintaining architectural flexibility is paramount. As we will explore later, some less programmable silicon architectures cannot adapt to changes in network requirements resulting from evolving technologies. The cost savings of rigid pipelines come at the expense of using the optimal approach: saving pennies in CapEx could lead to losing dollars from increased OpEx, declining differentiation, and a poor user experience.

With the Cisco 8000 Series, we're not claiming to deliver the holy grail. But we do think it gets you close to the middle of the eternal Venn diagram of technology.

# Cisco Silicon One in the Network

## Network Roles Today

As mentioned previously, the industry has evolved to the point where practically every network role has a custom-tailored device. However, there are several broad categories of requirements that we can aggregate. Of course, there are many more subsets of roles in the network but we will focus on DC fabric switching, core routing, and edge routing. Defining these roles and the feature/performance they require is key to understanding how the Cisco 8000 Series can perform so well where other platforms with other silicon architectures would struggle.

### Cisco Silicon One Across Roles

How is the Cisco 8000 Series uniquely qualified to handle all three of these use cases? Cisco understood that the silicon industry was coming to an inflection point and a new architecture of silicon was needed. The 2016 acquisition of Leaba Semiconductor kickstarted the design from the ground up of the Cisco Silicon One family. Leveraging advanced intellectual property from Leaba, we combined a game changing memory management system, a Run-To-Completion (RTC) forwarding model, and innovative optimization to bring switching, routing, and fabric capabilities into the same architecture.

We accomplished several engineering feats:

- The new memory management system enables higher utilization efficiency of on-die memory than other architectures. Memory within the device constitutes one of the largest component by size, so small improvements can have a large overall impact.
- A Run To Completion (RTC) model with the efficiency of a programmable pipeline, leveraging P4.
- A flexible model where ports can be assigned to different roles – front-facing or fabric – depending on the chassis form factor. In traditional architectures a port is either standard ethernet or a proprietary fabric.

The beauty of the Silicon One Architecture is based on this premise: a holistic design that allows each component to scale to meet new requirements; yet all silicon still use a common language for implementation.

In the next section we will showcase how this forward-looking approach has allowed Cisco Silicon One to innovate new silicon so quickly.

## Cisco Silicon One Family

In December 2019, Cisco announced the first Cisco Silicon One silicon: the Q100. Less than 22 months later, we now have 11 devices (Figure 3) that span all of the aforementioned roles: DC, core, and several types of edge deployments. The work over the past 2 years demonstrates both the architectural breadth and velocity achievable by a single, flexible architecture and how that enables our customers to scale and innovate.

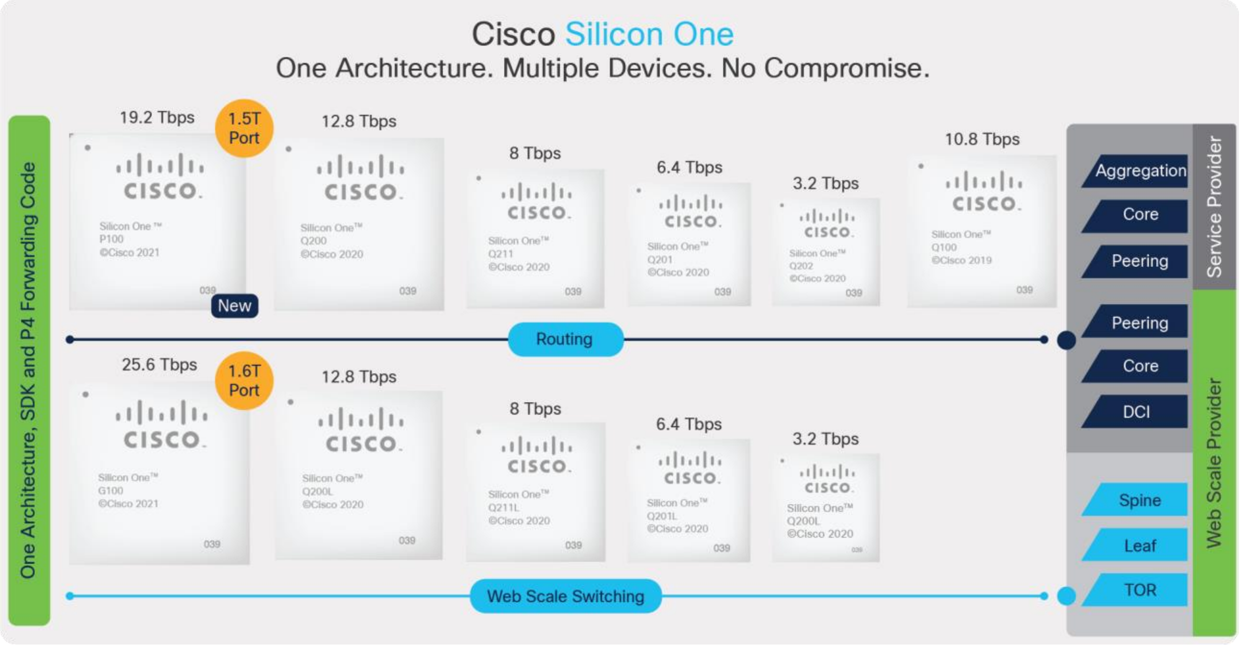
Cisco Silicon One
One Architecture. Multiple Devices. No Compromise.
One Architecture, SDK and P4 Forwarding Code
19.2 Tbps
1.5T Port
Silicon One™ P100 ©Cisco 2021
New
12.8 Tbps
Silicon One™ Q200 ©Cisco 2020
8 Tbps
Silicon One™ Q211 ©Cisco 2020
6.4 Tbps
Silicon One™ Q201 ©Cisco 2020
3.2 Tbps
Silicon One™ Q202 ©Cisco 2020
10.8 Tbps
Silicon One™ Q100 ©Cisco 2019
Routing
25.6 Tbps
1.6T Port
Silicon One™ G100 ©Cisco 2021
12.8 Tbps
Silicon One™ Q200L ©Cisco 2020
8 Tbps
Silicon One™ Q211L ©Cisco 2020
6.4 Tbps
Silicon One™ Q201L ©Cisco 2020
3.2 Tbps
Silicon One™ Q200L ©Cisco 2020
Web Scale Switching
Service Provider
Aggregation
Core
Peering
Web Scale Provider
Peering
Core
DCI
Spine
Leaf
TOR

**Figure 4.**
Cisco Silicon One Family

It takes a willingness to break from conformity to design a chip from the start that can incorporate DC, WAN, and Edge forwarding features. From the Q202 to the G100, customers have the ability to provision Cisco Silicon One in a range of Cisco 8000 platforms across their network from 3.2T up to 25.6T, and across routing and switching roles.

There are many advantages the Cisco 8000 Series inherits from a streamlined architecture family. Some benefits are operational, as we've already discussed, but others – like reduced manufacturing costs and optimized development cycles – are ultimately passed on to our customers.

As DC and WAN architectures shift to support new models of applications, network operators need hardware platforms that can evolve and innovate as well. Cisco's up-front investment in Cisco Silicon One has laid the foundation for delivering at a rate of a new piece of silicon every 2-3 months.

## Future

We’ve established the value of having the same silicon family across multiple roles in the network, which provides benefits at any snapshot in time – creating designs, developing, and optimizing software, understanding diagnostics and serviceability – but what about the future? Won't the speeds just increase?

In fact, there’s a subtle but important aspect of futureproofing that goes beyond faster links: evolving application infrastructure requirements lead to sometimes drastic differences in network requirements. Here are some examples:

- Adopting a distributed storage architecture in the data center would result in increased capacity and buffering requirements to ensure lossless transmission.
- Adding capacity to respond to increased bandwidth demands could mean moving from fixed to modular systems.
- As hybrid work arrangements proliferate, edge networks will need higher capacities with more complex data plane capabilities to accommodate the bandwidth demands of remote workers.
- As speeds increase and optics dominate the cost and power requirements of network ports, on-board or in-package optics could provide substantial savings.

The Cisco 8000 Series anticipates a wide range of architectural directions that will help adopters amortize the onboarding and development costs across a number of roles in the network and several generations of technological maturity:

- As bandwidth and buffering needs grow, the ability to scale our architecture to incorporate more ports or larger buffers means a smooth transition to higher speeds.
- Multiple products within the Cisco 8000 Series – both fixed and modular – are using Silicon One, enabling architectural consistency regardless of form factor.
- The flexible silicon architecture means the right device can work optimally at its place in the network, while retaining the “common language” of the architecture that drives the operational efficiencies discussed earlier.
- At some point on the bandwidth curve, the power required to drive copper signals to transceivers will start to exceed facility budgets, especially in larger data centers. Cisco Silicon One anticipated this architectural shift and will efficiently incorporate on-package optics when the industry needs it.

In summary, the future of your network’s architecture will be dictated by the same important considerations as always: application performance, power, and cost. You can accommodate any future more efficiently by putting the Cisco 8000 Series with Cisco Silicon One at the center of your strategy.

## Cisco Silicon One – The Hardware

### Divergent Evolution of Role-Based Devices

Cisco teams have been developing networking silicon for decades. We deeply understand where we've been, where we are, and where we're going as an industry. While Cisco has built many different product lines based on many different variants of architectures to tackle many different network requirements, this model comes with inefficiencies. For operators, the overhead of designing, developing, and deploying so many different hardware architectures required a substantial workforce, highly complex automation, or both.

So why not just build **one** chip that does everything all at once? Because the size of silicon has an upper bound based on the reticle limit[2] – the area of visibility under the lens used to etch transistors and wires into silicon. Want to add more packet processors? More memory? Greater bandwidth? Crypto capabilities? Sure, you can do any of these...but what will you remove to make room? Of course, the reticle limit is a long-standing constraint in chip design – nearly all chips make full use of the available space. Improvements typically come through a new process (shrinking transistors) or innovating individual components (like new memory technology).

Cisco Silicon One brought a major innovation: flexibility of individual components of the chip to scale up or down while retaining a "common language" in the architecture; in other words, without building an entirely new forwarding paradigm. As an example, compare the two diagrams in Figure 4, which offer approximate representations of a DC Fabric device and a WAN device.

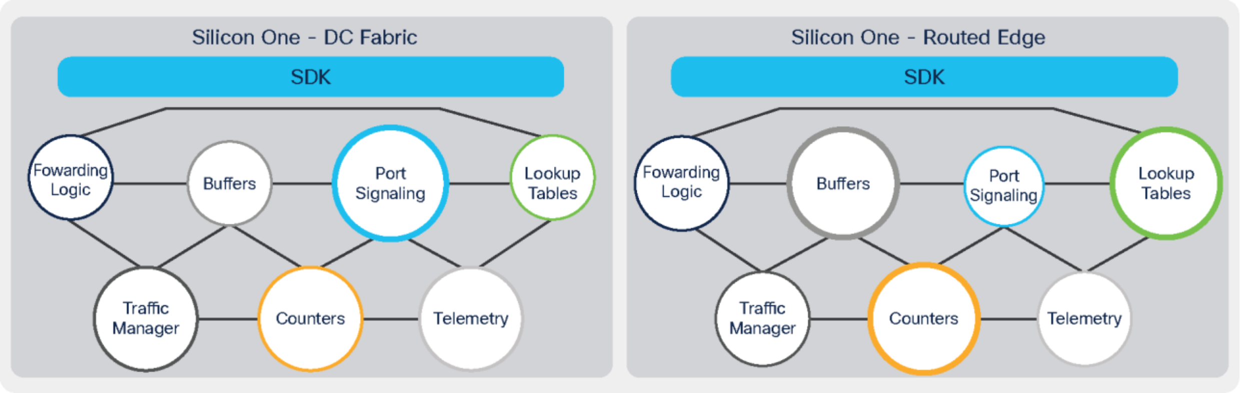

**Figure 5.**
DC Fabric device vs. Routed Edge device

[2] https://semiengineering.com/chip-dis-integration/#:~:text=The%20reticle%20size%20limits%20the,or%20imperfections%20in%20the%20mask.

The DC Fabric design emphasizes Port Signaling for greater bandwidth, while scaling back on Buffers, Lookup Tables, and Forwarding Logic, which need to be far smaller in a DC Fabric. The WAN design decreases emphasis on Port Signaling, while beefing up Buffering, Lookup Tables, and Forwarding Logic – reflecting the need for these devices to accommodate a wide range of port speeds, hold large forwarding and filtering tables, and perform more complex packet operations.

Both devices would run right up against the reticle limit. However, before Cisco Silicon One, these two were two different architectures, with different connectivity and SDKs. An operator would deploy a unique architecture in its corresponding role in the network. With the Cisco 8000 Series, an operator can deploy different flavors of the same architecture.

With these topics considered, are there certain niches that require extreme capabilities like hyper-low latency or elaborate per-flow state that would benefit from highly customized devices? Absolutely. But for everywhere else, you could gain substantial consistency if you had the same silicon logic. Silicon designed with flexibility in mind from the outset will let you address all but the most "corner" of the corner-cases.

Consistent behavior, consistent performance, consistent operations, all through consistent deployment of Cisco Silicon One.

## Buffering

On the surface, understanding buffer memory in a network device seems simple: look at any two devices and compare packet buffer sizes. In actuality, the analysis should not stop there. Packet buffer memory architecture is often overlooked but can have an outsized impact on network performance during congestion.

On modern systems there are primarily two models of buffer architecture SRAM (on-die and fast) and external DRAM (off-die and slow). Like specific-purpose architectures, these two buffer architectures were designed within the constraints of the devices they were meant to service.

On-die packet buffers are used due to their fast speeds and low power consumption which makes SRAM the ideal choice. To minimize costs, ethernet switches used exclusively on-die packet buffers, which enables high speeds at low cost. However, because SRAM is on-die, the amount of die area required for this packet buffering directly competes with other vital components like forwarding logic.

So how does the memory architecture translate to practical application? Buffers handle bursts of traffic that exceed an interface transmission rate. Instead of dropping a packet, it gets temporarily stored in buffer memory. The faster the interface, the quicker the buffer fills. For short links, like in a data center, host traffic congestion protocols can throttle quickly, so fairly shallow buffers are needed. However, for routed links such as WAN backbones that span many kilometers, much deeper buffers are needed to sustain long bursts of traffic before host traffic congestion protocols engage. The buffering needed for even a single interface can exceed the total on-die SRAM.

To accommodate both scenarios, another solution was needed.

## Hybrid HBM

In 2013, a new technology came onto the scene called High Bandwidth Memory (HBM). HBM allowed for off-die external buffering using Dynamic Random Access Memory, or DRAM. With the memory moved off-die, HBM could be hundreds of times larger than on-die SRAM allowing for very deep buffers. The trade-off is that accessing this remote memory is slower than accessing local SRAM.

In addition to buffer size and architecture, modern applications pose a third challenge to buffer performance. Complex applications such as big data, AI training, and High-Performance Compute (HPC) can create unpredictable conditions that can saturate a network. While such conditions are transient, swamping a network beyond its ability to buffer results in loss of packets and degrades application performance. It might also result in the loss of valuable data.

To address both the issue of buffer oversubscription and transient congestion, two solutions were developed--a hybrid HBM system and a shared memory management system.

The first advancement was Cisco's hybrid HBM[3]. In a hybrid HBM design the memory scheduler works with both the SRAM and the HBM to achieve optimal performance. The majority of the flows can be handled easily by the SRAM. The memory manager monitors for large, aggressive flows that might contribute to short term congestion, migrating those flows to HBM as needed (figure 5). When congestion has passed, these flows can be returned from HBM to the internal SRAM. This scheme achieves the peak performance for most flows but has the flexibility to absorb short term congestion without affecting all flows.

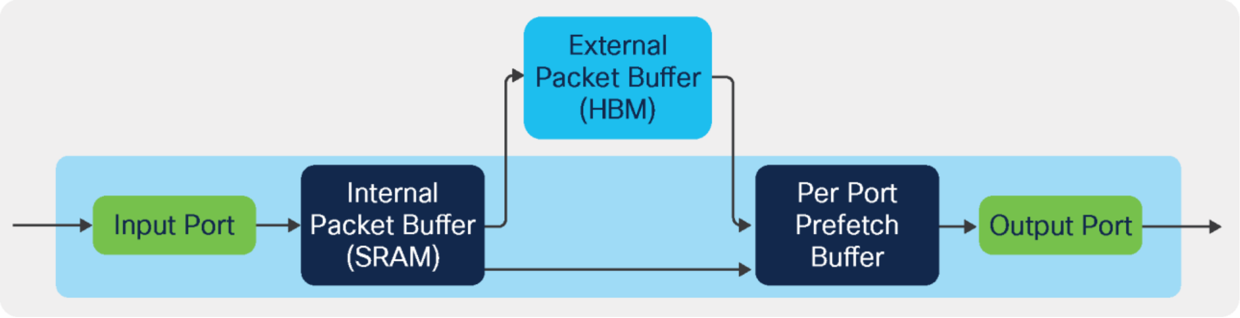

**Figure 6.**
Hybrid HBM

[3] https://www.cisco.com/c/dam/en/us/solutions/collateral/silicon-one/white-paper-sp-hybrid-buffer-architecture.pdf

## Shared memory manager

Not all memory sub-systems are created equal. Many manufacturers claim a fully shared buffer architecture for their on-die SRAM, but when you peel back the layers you find many constraints. You can characterize these architectures usually into two types: input buffering or output buffering[4]. In both cases (as you see in figure 6) the memory, though technically accessible by all ports, is broken down into pools.

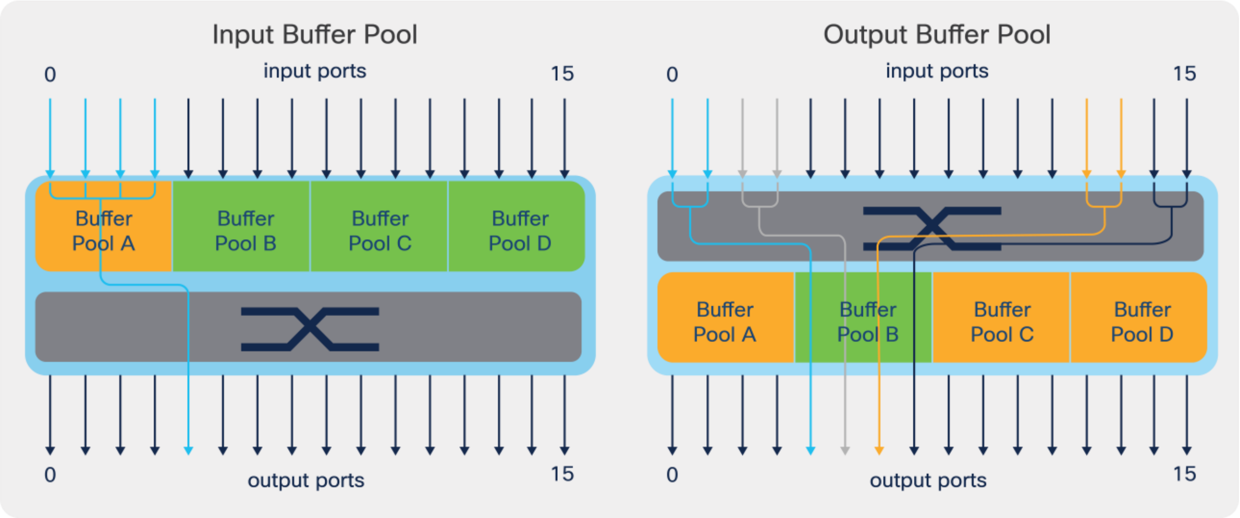

**Figure 7.**
Buffer Restrictions

Where constraints become apparent is when you have oversubscribed traffic patterns, and a particular pool can easily be saturated while other pools can sit idle. In the example on the left, input buffer pool A is saturated even though only 25% of total buffer is consumed. On the right example, output buffer pool B is saturated with only 25% of total shared buffer utilized.

In Silicon One we have a fully shared buffer architecture (figure 7). The buffer space is fully shared between both the ingress and egress ports with no restrictions. There is no buffer pool carving as seen in traditional architectures, allowing for burst absorption for flows that would saturate other architectures.

[4] https://blogs.cisco.com/sp/optimize-real-world-throughput-with-cisco-silicon-one

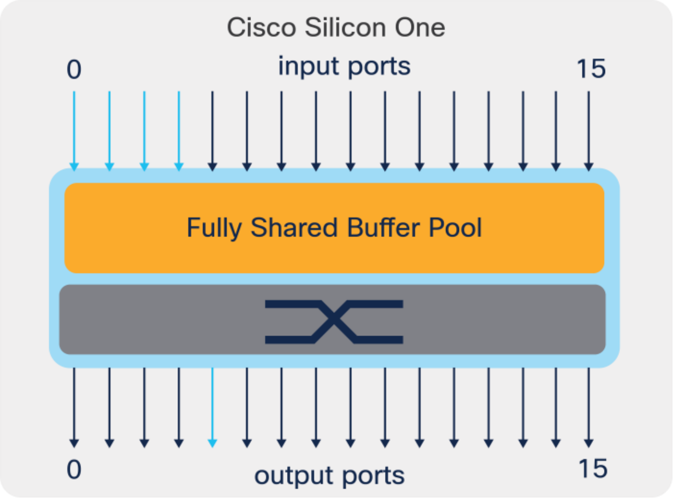

**Figure 8.**
Fully Shared Buffer

Additionally, the fully shared buffer pool allows for a massive power reduction. Significant power is consumed when writing and reading packet data into and out of memory. With Cisco Silicon One's unique memory architecture, we write the data once and read the data once.

The combination of a hybrid HBM with a fully shared buffer allows Cisco 8000 Series systems to achieve an effective buffer size much larger than the alternatives.

## Power

### Network Device Power Consumption

Power consumption over time is often overlooked in network implementations. The reality is that power consumption has both significant upfront cost in provisioning capacity and in the ongoing operational cost in kilowatt hours (kWh) consumed over the lifetime of the device.

Traditionally, network equipment selection has focused more on interface speeds and density than power consumption. Hence, compared to compute, power consumption by network devices has been relatively small. However, as network port speeds surpass 400Gb, more power is needed to drive high bandwidth connections between system components and to drive high-speed optics. The networking power consumption that was easily overlooked before now translates into significant Operational Expenditure (OpEx) incurred year after year.

Nowhere is this a more critical issue than data centers where power is a zero-sum game. The more power allocated for network infrastructure; the less power available for revenue-generating workloads.

Cisco Silicon One impacts OpEx and therefore ROI for facilities by realizing a significant reduction in the amount of power needed through a few key innovations.

### Holistic Architecture Optimization

The Cisco engineering teams team took a relentless approach to reduce power at every opportunity. Achievements were made in die layout, memory technology, and the RTC engine to reduce power consumption. A recent analysis of routing systems built on 12.8 Tbps performance found that a Silicon One Q200-based system consumed 390W and the next closest competitor system would consume more than 1000W. The analysis[4] found the typical colocation facility charges $6,300/kW annually so savings add up quickly. Simply put, the Cisco 8000 device just requires less power to run.

### System Design

Most engineers understand that network devices generate heat that must be dissipated by the cooling subsystem, but the wide range of components and deployment environments makes optimal heat dissipation incredibly challenging. To optimize system design, the Cisco Silicon One team incorporated advanced thermal telemetry into the device itself. This telemetry has allowed the Cisco hardware and software engineering teams for the Cisco 8000 to create sophisticated dynamic fan control algorithms. The ability to adaptively drive the system fans at a slower speed has a significant impact on power consumption, as seen in Figure 8.

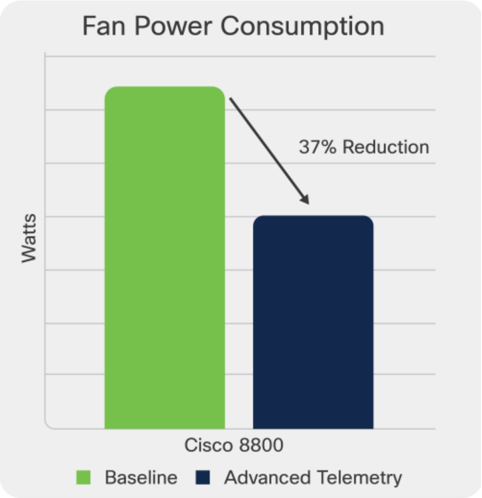

**Figure 9.**
Fan Power Consumption

In real world customer deployments of the modular Cisco 8800 systems powered by Cisco Silicon One, lower power consumption and smarter fan algorithms result in tangible savings. A study[5] found that a single 12.8Tbps router could potentially save $143,000 over a 10-year life span in power costs alone.

While the savings of a single kWh might seem insignificant, large networks running hundreds or even thousands of devices can realize power savings into the millions of dollars – a substantial positive environmental and financial impact.

[5] https://www.cisco.com/c/dam/en/us/solutions/collateral/silicon-one/white-paper-sp-fixed-box-router.pdf

# Cisco Silicon One – The Software

## Software

Today the Cisco 8000 Series supports many devices and different network operating systems such as Cisco's IOS XR7, and Software for Network in the Cloud (SONiC[6]). This support allows the customer to choose the NOS stack that makes the most sense for their use cases and operations model.

**Furthermore,** the software stack model for Cisco Silicon One is designed to be extensible and create an easy path to onboard new operating systems. An example of this extensibility has been demonstrated in Silicon One's support for FBOSS, the Meta-developed NOS.[7] Although not a Cisco 8000 Series platform, it highlights the flexibility provided by the hardware and software stacks of the Cisco 8000 Series.

This portability was achieved through the software stack seen in Figure 9. At the bottom is the actual bytecode, derived from a P4 program that is compiled and installed on the Cisco Silicon One device. The top layer of the SDK is the API, which serves as an abstraction layer for a wide range of networking methods and objects that are familiar to network engineers. In between, the driver translates between the higher-level API and NPU-specific instructions, ultimately resulting in data retrieval from the various hardware memory tables to supply back to the caller.

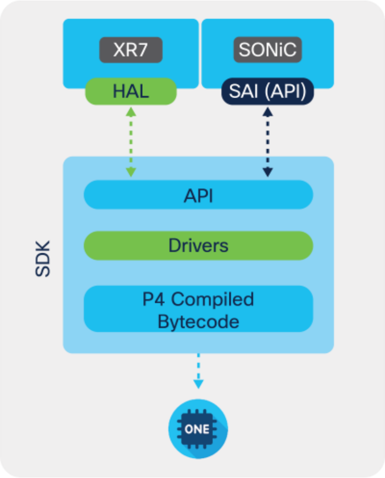

**Figure 10.**
Software Stack

Above the SDK, the software stack diverges to interact with different abstraction layers. For IOS XR7 a Hardware Abstraction Layer (HAL) is used which interacts directly to the SDK through the northbound API. SONiC (and FBOSS on the Wedge400C) utilize a model known as Switch Abstraction Interface (SAI). As such the SDK API interacts with the SAI translation layer.

There are strengths in both Cisco 8000 Series operating systems. Which option is right for you depends on your operational model and your approach to software development and support. The benefit for customers and the community at large is that the software diversity model is proven and will continue to flourish.

[6] https://blogs.cisco.com/sp/cisco-goes-sonic-on-cisco-8000

[7] https://blogs.cisco.com/sp/cisco-and-meta-partner-on-wedge400c-data-center-switch

## Value of a Single SDK

The never-ending challenge in technology is deciding where to generalize and where to differentiate. For network devices, this challenge is exacerbated by the relatively high level of vertical integration that is prevalent across the industry. Consider the technology stacks that typically exist across a run-of-the-mill production network in the past decade, as portrayed in Figure 10.

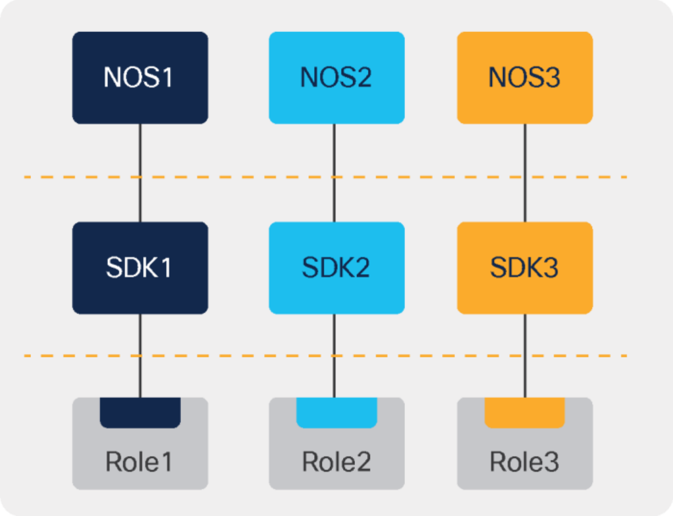

**Figure 11.**
Disparate stacks with different silicon architectures

As we discussed earlier in the "Simplicity" section, multiple SDKs across multiple NOS don't have a consistent abstraction layer from which to gain operational simplicity and efficiency. One approach for operational efficiency involves adopting a common NOS, as portrayed in Figure 11.

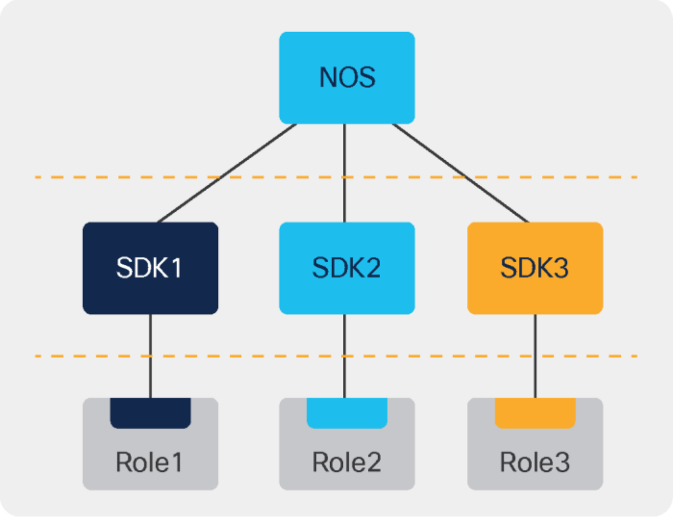

**Figure 12.**
Common NOS abstracting different silicon architectures

Even with an NOS abstracted across SDKs, the differences in forwarding behavior, provisioning, and resource monitoring among the different devices can still result in operational overhead and technical debt in operations. With the flexibility of deployment of Cisco Silicon One, modern networks can adopt a new abstraction layer to drive operational efficiency. Consider the transformation brought to the technology stack, as in Figure 12.

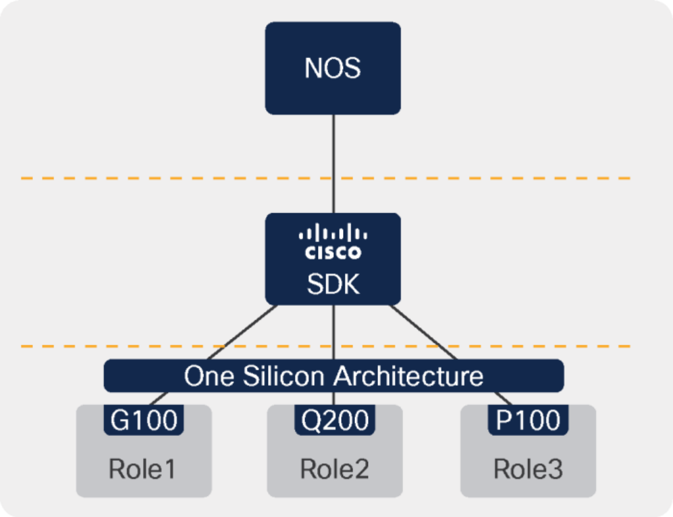

**Figure 13.**
One SDK for One silicon family

As discussed, benefits of this One Architecture model include significant reuse of network operations artifacts, including provisioning agents, Methods of Procedure (MoPs), and monitoring schemes. Operators also benefit from Cisco Engineering's improved code reuse. And if an operator is taking on more of the NOS development work themselves, they're going to need the savings of code-once-deploy-in-many-roles to justify their efforts. Either way, they can be confident that the behavior and performance will be consistent across their network, resulting in greater overall efficiency through simplicity.

Put another way, network device development used to be a marathon – a long stretch of software releases in a tightly-integrated stack for **each platform**. Now, it's more of a relay sprint – the SDK team can develop once, then hand it off to other NOS or customer teams, who can then deliver new capabilities on their own timelines. That means velocity, agility, and scale...all with higher quality from less code.

## P4 and Packet Forwarding Model

The Cisco Silicon One architecture introduces a new paradigm in packet processing gaining the benefits of Run To Completion (RTC) but with the efficiency of a programmable pipeline. This enables a single device to provide the optimal packet processing for the packets flowing through your network at any one point – never too much, never too little, and always power-efficient.

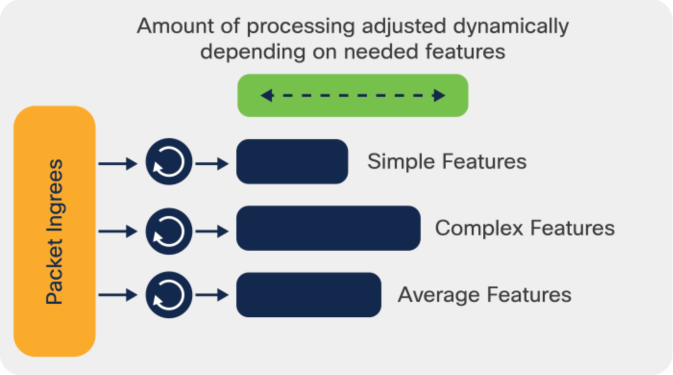
Amount of processing adjusted dynamically depending on needed features
Packet Ingrees
Simple Features
Complex Features
Average Features

**Figure 14.**
RTC Forwarding Model

As we've all learned in this industry, it's one thing for the hardware to be capable. It's quite another to have software that unlocks that capability. So how do Cisco 8000 platforms do it?

P4 is a data plane specification language intended to separate intent from implementation. It uses the fundamental notion of a match-action table as a building block for defining increasingly complex per-packet operations. The benefits of P4 for data plane development include:

- A consumable, readable, and writable syntax for expressing packet forwarding behaviors
- Concise definition of behavior that can rapidly be validated in simulation
- An efficient abstraction layer with a consistent API across a silicon family
- User-defined extensions that can, for example, gather additional telemetry or network visibility

All of these benefits can be realized without requiring a change to the silicon architecture. It's almost like a get-out-of-jail free card for network functionality – new features without rip-and-replace.

## Conclusion

The Cisco Silicon One architecture was designed to liberate customers from the traditional way of operating networks – a patchwork of vendors and operating systems that spanned across many life cycles. This legacy created operational and financial debt for customers maintaining infrastructure in an increasingly complex world. The Cisco 8000 Series powered by Cisco Silicon One now frees IT teams to regain the agility they have lost through a unified architecture that reduces the hardware/software permutations in their network. This leads to an across-the-board increase in operational efficiencies from software qualifications through automation development.

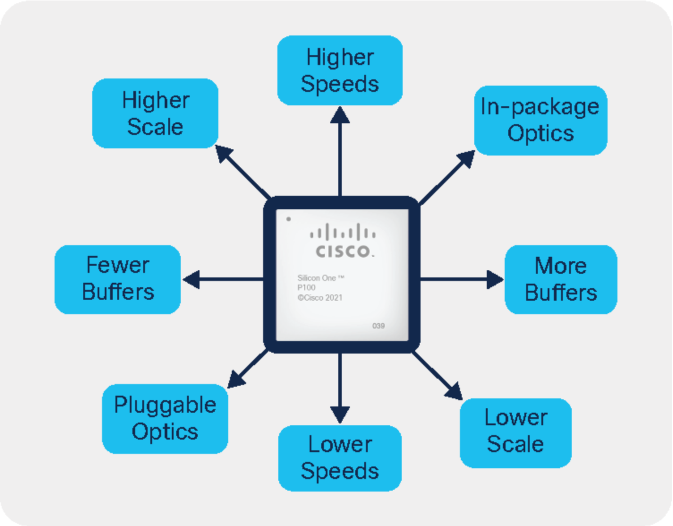

**Figure 15.**
Role flexibility

By incorporating groundbreaking technology such as our advanced memory sub-systems, P4 + RTC engine, and large-scale power savings, Cisco has created the world's first architecture that can be deployed throughout your network. The Cisco 8000 Series powered by Cisco Silicon One empowers operators to consume hardware and software on their terms and frees operational teams to innovate at the pace their business demands. Operators have never had more control over their own network infrastructure.

**Americas Headquarters**
Cisco Systems, Inc.
San Jose, CA

**Asia Pacific Headquarters**
Cisco Systems (USA) Pte. Ltd.
Singapore

**Europe Headquarters**
Cisco Systems International BV Amsterdam,
The Netherlands

Cisco has more than 200 offices worldwide. Addresses, phone numbers, and fax numbers are listed on the Cisco Website at **https://www.cisco.com/go/offices**.

Printed in USA C11-2696412-01 05/22