WHAT IS SOFTWARE DEFINED NETWORKING? SDN EXPLAINED



Software defined networking (SDN) is a network architectural model that allows programmatic management, control, and optimization of network resources.

SDN decouples network configuration and traffic engineering from the underlying hardware infrastructure, to ensure holistic and consistent control of the network using open APIs.

Let's take a look.

The limits of traditional network architecture

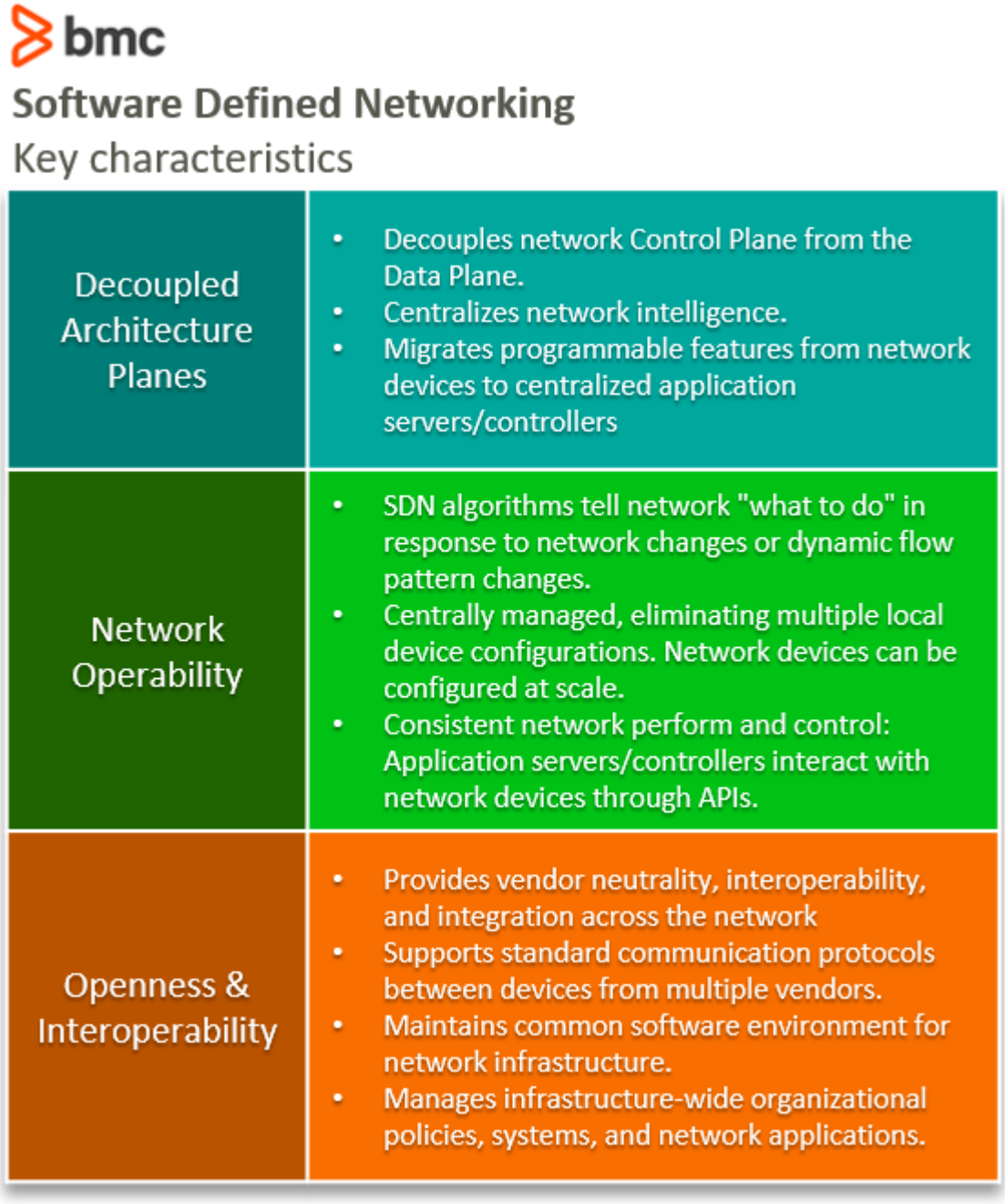
Traditional network architecture offers minimal flexibility to coordinate between fixed function network devices that must be configured manually. A single change can have a cascading effect on the network performance and has the potential to bring down the entire network.

The exploding volumes of data traffic, complex network architecture, and growing demands to improve network performance obsoletes the traditional approach to network management.

How software defined networking works

The static nature of the traditional network architecture model fails to meet the demands of modern business IT. Organizations require network infrastructure that allows the flexibility to scale and support dynamic computing environments based on rapidly evolving technology and business landscapes.

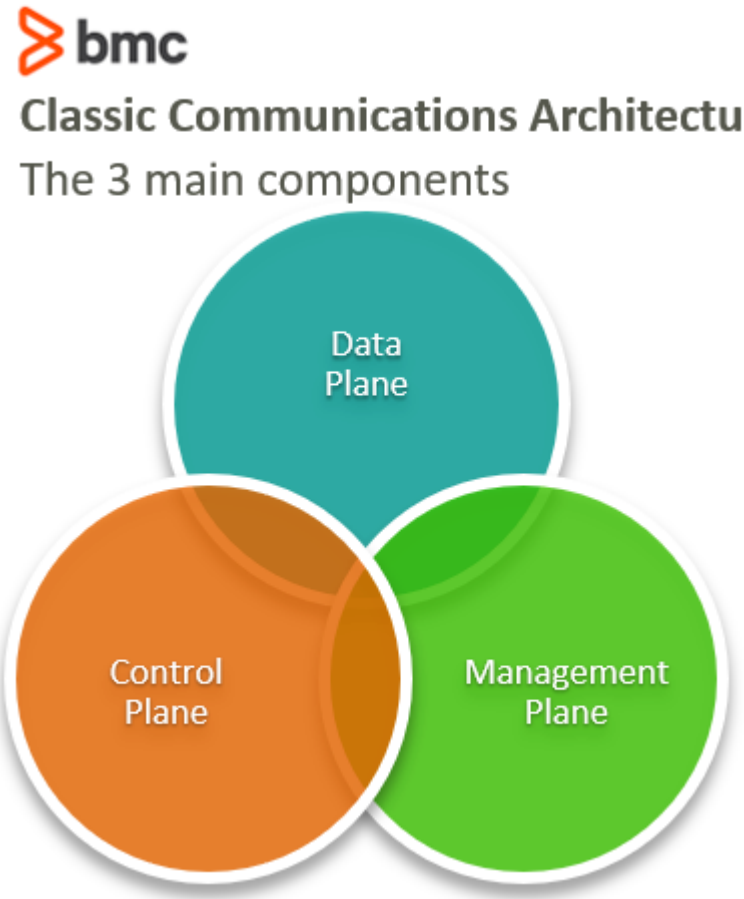
Software Defined Networking offers the following key characteristics to address these concerns:



Let's look at each characteristic in more detail.

Decoupled architecture planes

A classic communication network architecture consists of three main components:



- **The Control Plane** refers to the network architecture component that defines the traffic routing and network topology.
- **The Data Plane** is the network architecture layer that physically handles the traffic based on the configurations supplied from the Control Plane.
- **The Management Plane** takes care of the wider network configuration, monitoring and management processes across all layers of the network stack.

In traditional network architecture, the control plane and data plane are integrated. Any changes to the system are dependent upon configuring physical network devices, the protocols, and software they support. You can perform only limited changes to the overall system as the network devices bottleneck logical network traffic flows. Devices function autonomously and offer limited logical awareness toward the wider network.

In contrast, SDN decouples the Control Plane from the Data Plane and centrally integrates the network logic at the controller level. A controller separated between the two Planes logically centralizes the network intelligence such that users can choose which programmable features to move from network devices onto the application server or controller.

The logically centralized and decoupled controller operations allow organizations the enhanced agility to automate, extend, monitor, maintain, manage, extend, provision, and troubleshoot the network infrastructure.

Applications interact directly with the controller to obtain a global view of the network state. The dynamic, scalable and flexible network infrastructure leads to simplified operations and the ability to test new business opportunities that are otherwise limited due to network architectural bottlenecks.

Network Programmability

Imagine programming 10 routers, one by one, and then keeping track of how every device was configured and supposed to behave. Scale it to hundreds and thousands of devices, and the manual processes involved in device monitoring, configuration and management are rendered ineffective. Even the network management systems (NMS) will not suffice to eliminate network bottlenecks, errors, and performance issues.

With SDN algorithms, the number of repeatable device configuration and management processes is not a limitation. An SDN would see 1,000 routers in a similar way it sees 10 routers. Algorithms can incorporate the changing dynamics of the network configurations realized in scaled environments. The SDN capability essentially lets sysadmins tell the network "what to do" in response to network changes or dynamic traffic flow patterns.

Additionally, network programmability is key to achieve global optimization toward a consistent network-wide state. Networks do not converge to a single state when each component is programmed individually without awareness of the surrounding:

- Network environment
- State
- Configuration patterns

SDN lets users replace local optimization with a logically centralized intelligence and control of network resources. The underlying dependencies and limitations do not apply (at least in the same way) considering the decoupled nature of the Data and Control Plane components of the network architecture. Instead of having to tweak settings on hundreds of individual management consoles, applications can interact with the appropriate network devices through APIs and maintain consistent overall network performance, functionality, and control.

Openness and Interoperability

Vendor interoperability and network integration is a primary criterion for network traffic engineering—ranging from device planning and purchase to configuration and management. Vendor neutrality allows organizations to optimize infrastructure investments for technical and business requirements.

Consider the two layers of SDN interoperability: the infrastructure and the service:

- At the infrastructure layer, SDN supports standard protocols for communication between devices from multiple vendors and maintain a common software environment.
- At the service layer, SDN manages infrastructure-wide organizational policies, systems, and network applications. These can be divided into multiple open and modular dimensions.

In an SDN system, the open network orchestration, service, and network management systems simplify deployment. For instance, RESTful APIs can be used for communication via open protocols supported by all network device vendors. As a result, the network infrastructure is easy to extend, and the overall system does not run into network performance and scalability bottleneck.

The architecture abstraction layer allows controllers to manage traffic easily using open standards such as OpenFlow or other extensible protocols. Finally, the SDN controller platform itself operates as an open system that allows users to extend functionality, add applications via APIs or modules to

perform specific network management functionality.

SDN misconceptions

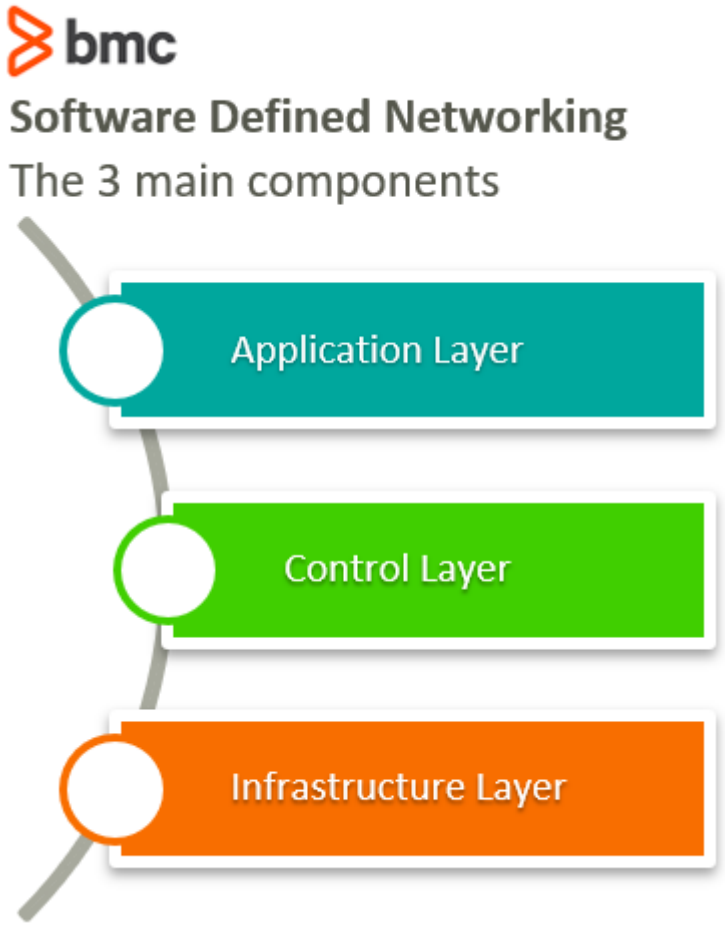
SDN is a significant architectural change over traditional networking infrastructure. However, the technology does not require you to disrupt your existing network and replace it with entirely new hardware and software resources that constitute SDN.

In fact, organization can begin with specific SDN use cases such as:

- Optimizing the network by monitoring
- Identifying account affinities and access control limits
- Orchestrating specific workloads as you adopt the technology at scale

In essence, SDN does not replace traditional networking devices or only apply to switches through automation. Instead, SDN only reshapes the architecture to enable centralized control with multiple levels of automation across the network.

The Open Networking Foundation identifies three main parts of the SDN model:



The major architectural differences between SDN and traditional network infrastructure are identified within the Control and Infrastructure layers. However, it is the SDN programs within the Application layer that define the new approach of data communication between controllers and services that run over the network.

The Controller and centralized Control Plane define how SDN is different from traditional network architecture. Application layer technologies are responsible for translating those differences into performance improvements, that result in tangible business results.

Related reading

- BMC IT Operations Blog
- OSI Model: The 7 Layers of Network Architecture
- Introduction To BYON (Bring Your Own Network)
- N-Tier Architecture: Tier 2, Tier 3, and Multi-Tier Explained
- What Does a Network Operation Center (NOC) Engineer Do?