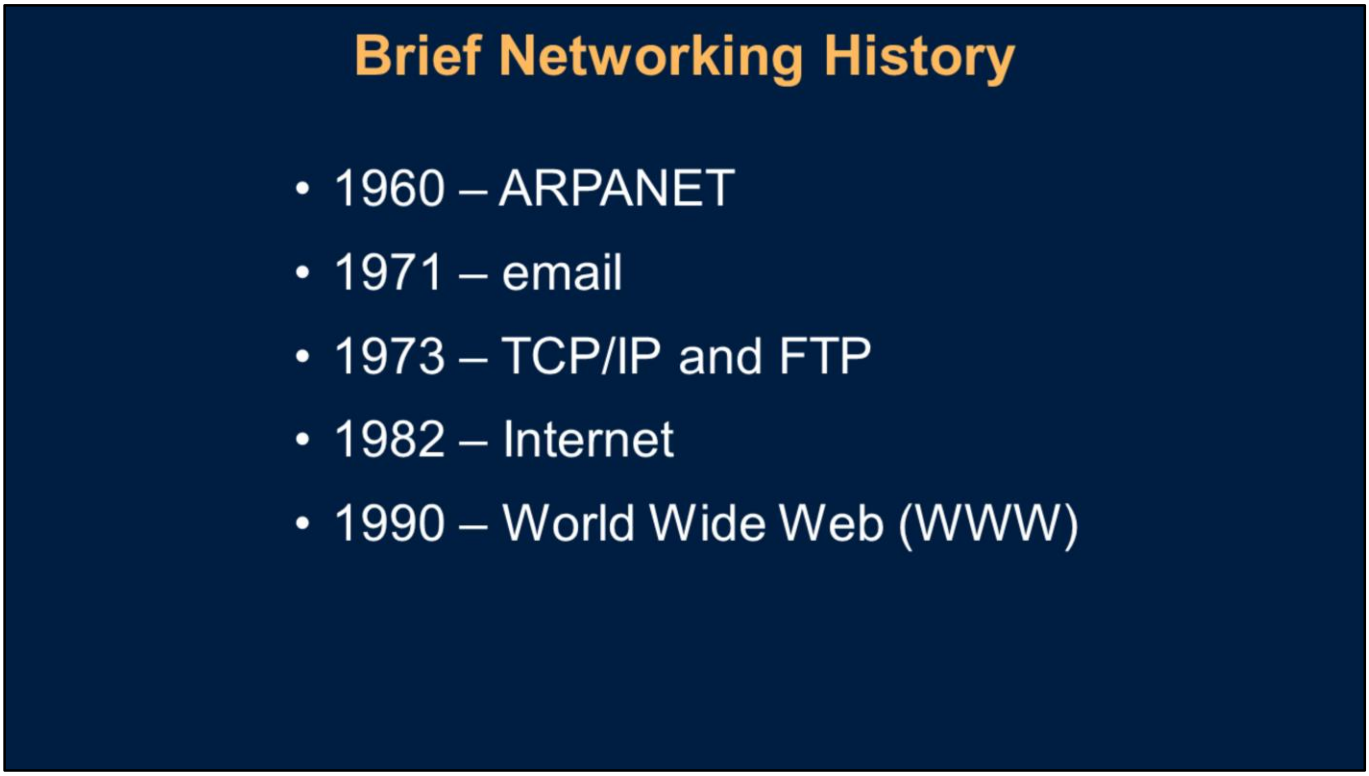
Simplifying the Substation Security With Software-Defined Networking



Brief Networking History

- 1960 – ARPANET
- 1971 – email
- 1973 – TCP/IP and FTP
- 1982 – Internet
- 1990 – World Wide Web (WWW)

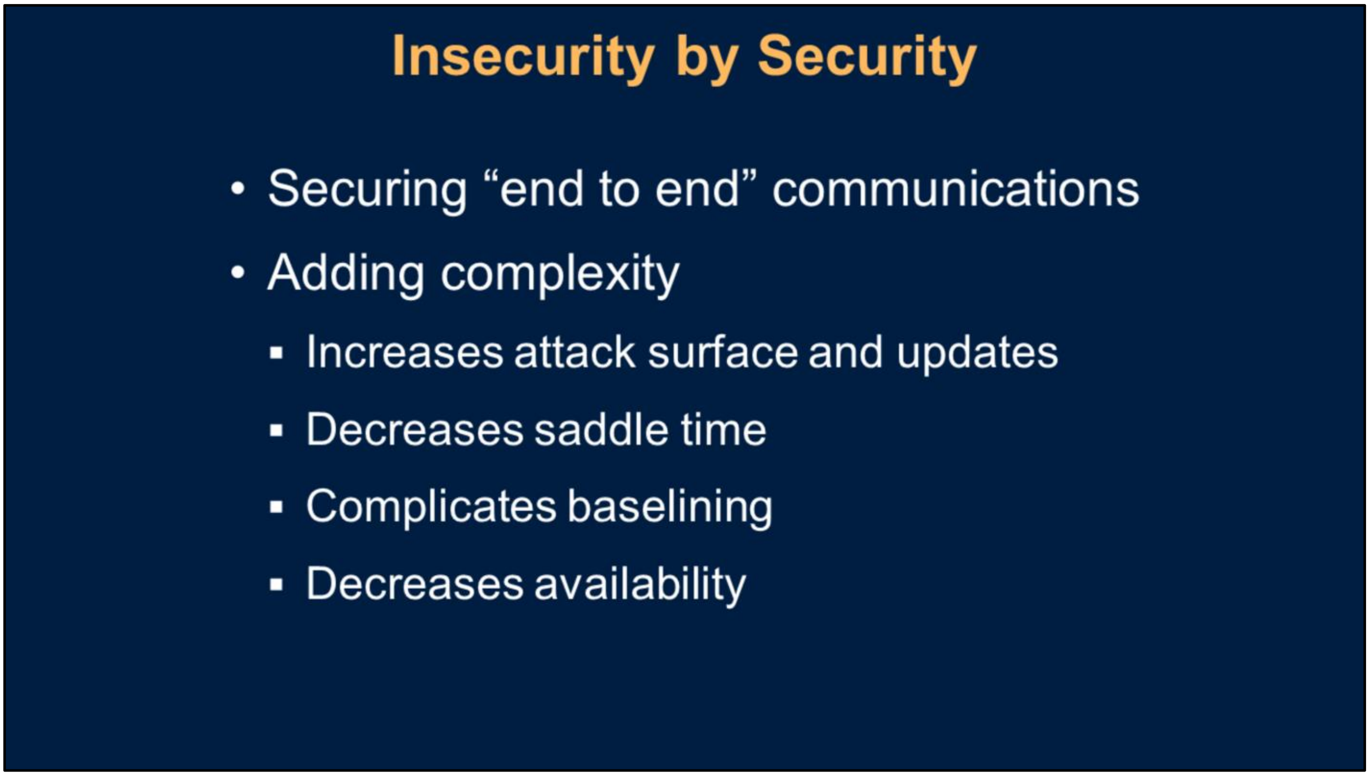
The Advanced Research Projects Agency Network (ARPANET) started with the Stanford Research Institute (now SRI International) and the University of California, Los Angeles (UCLA) in 1960. In 1970, ARPANET expanded with Harvard University and the Massachusetts Institute of Technology (MIT).

Ray Tomlinson invented email in 1971.

Transmission Control Protocol/Internet Protocol (TCP/IP) and File Transfer Protocol (FTP) came along in 1973 before Ethernet was realized.

In 1990, the World Wide Web (WWW) consisted primarily of .gif exchanges and bulletin boards, with HTML in its infancy.

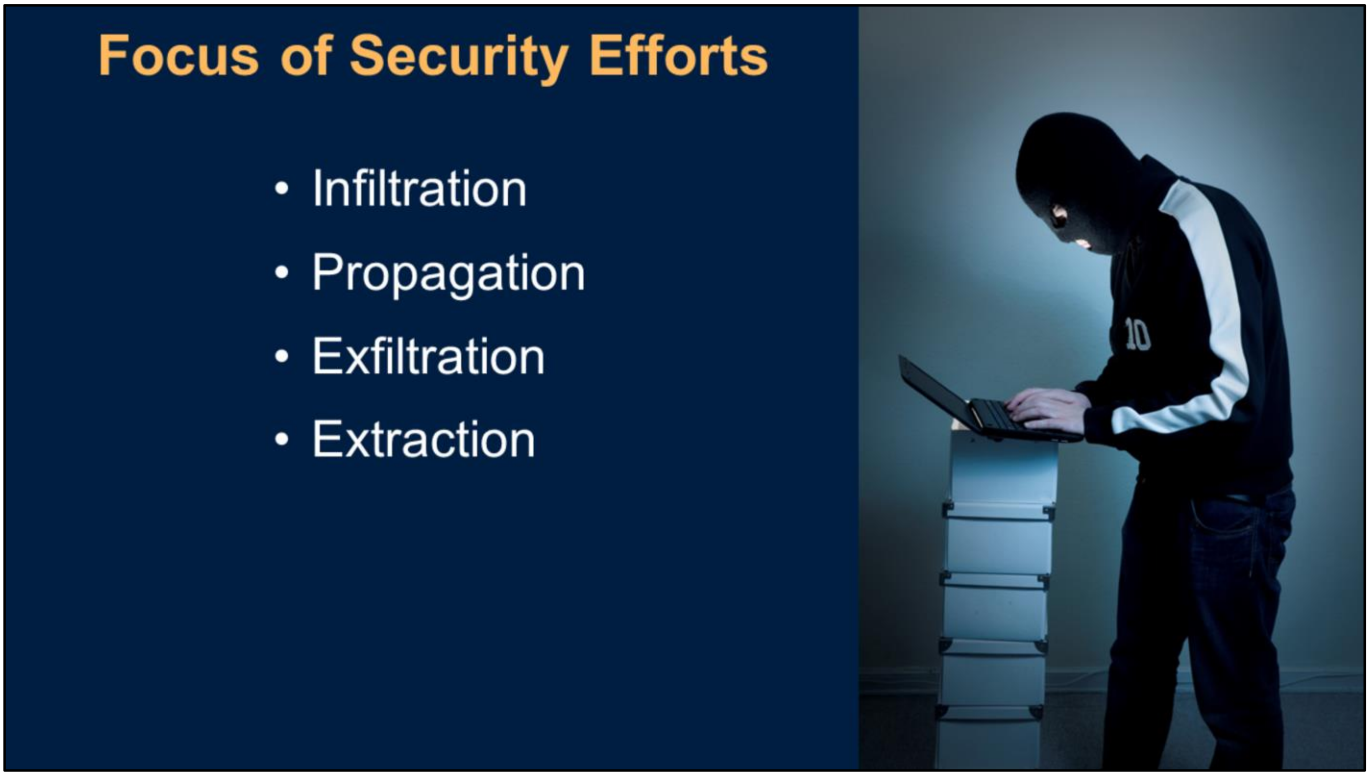
Cybersecurity was nonexistent for most network data exchanges until around 1994.



Insecurity by Security

- Securing “end to end” communications
- Adding complexity
  - Increases attack surface and updates
  - Decreases saddle time
  - Complicates baselining
  - Decreases availability

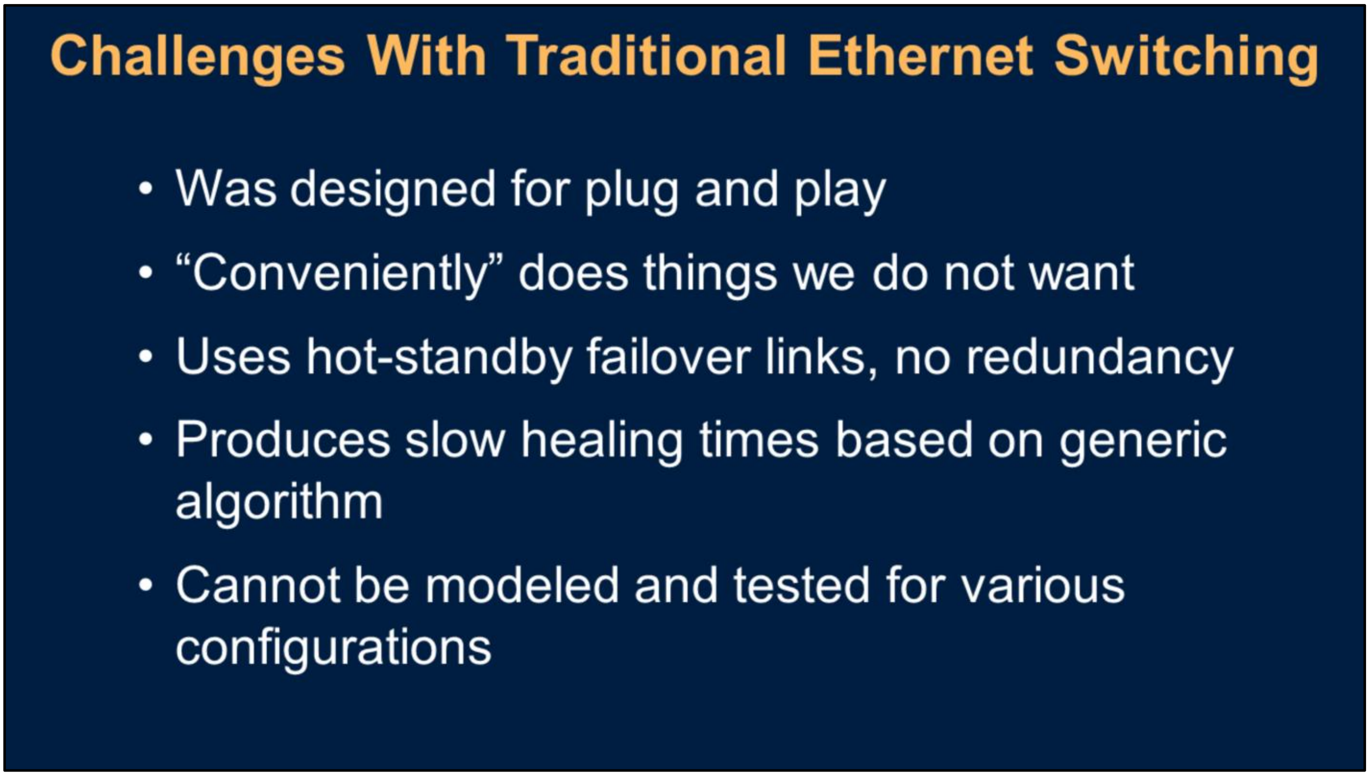
“End-to-end” security is loosely defined. Many solutions and applications today imply that full security controls are implemented on each of the communications channel end points and that data are exchanged without considering the data application or the network in between the end points. A tunneled exchange of data requires that the identities of the end points are known and verified on each end of the tunnel.



Focus of Security Efforts

- Infiltration
- Propagation
- Exfiltration
- Extraction

Most of our cybersecurity efforts (75 percent or more) are focused on preventing infiltration into the network. Typically, entities spend much less effort on detection and prevention of propagation, exfiltration, and extraction.



Challenges With Traditional Ethernet Switching

- Was designed for plug and play
- “Conveniently” does things we do not want
- Uses hot-standby failover links, no redundancy
- Produces slow healing times based on generic algorithm
- Cannot be modeled and tested for various configurations

Ethernet has become the dominant technology for local-area networking, but it has some specific challenges that limit its performance for applications that require low latency, fast failover, and predicable performance. These include the following:

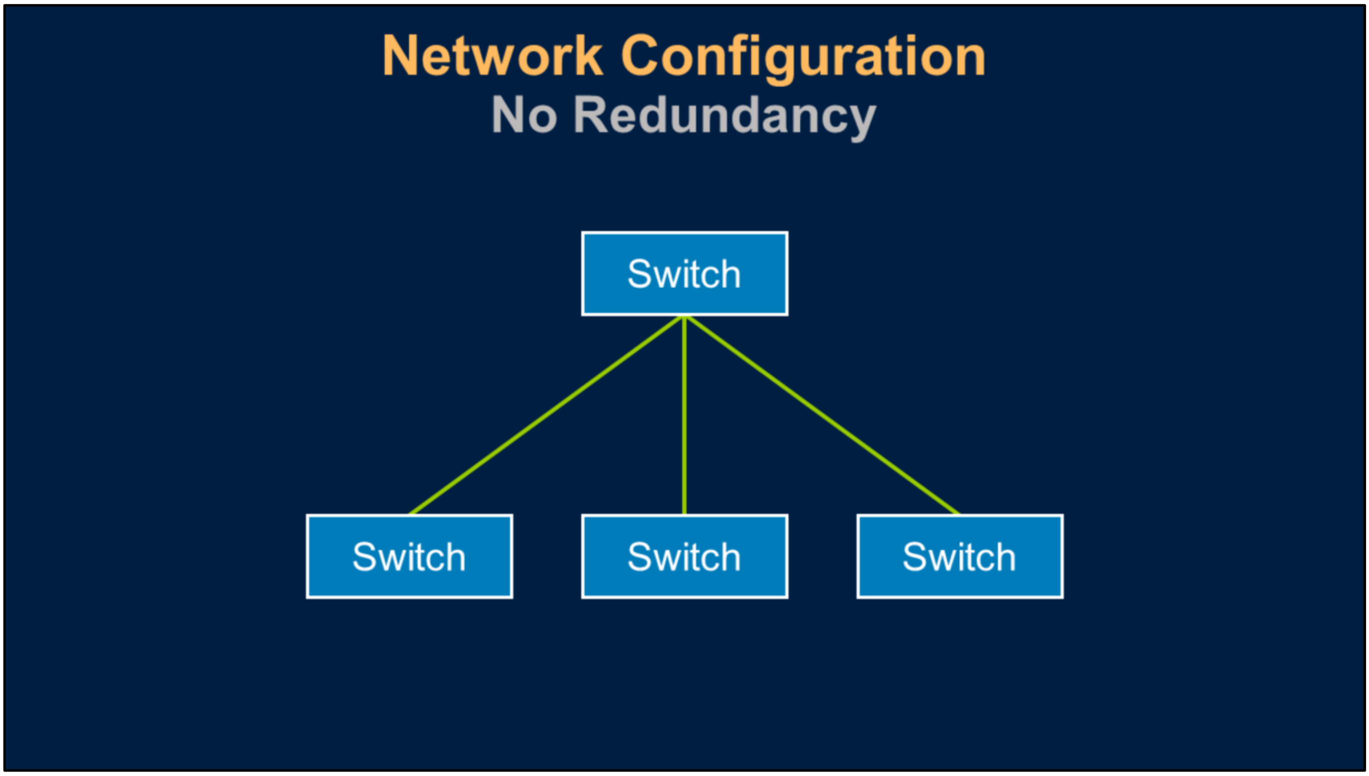
- Ethernet was designed with a plug-and-play philosophy that allows each Ethernet switch to discover neighboring devices and learn which ports to forward packets to.

- Ethernet uses protocols to determine neighboring devices, establish the root switch, and recover from link failures. While many of these protocols conveniently manage the operation of the network, they can result in unwanted behaviors such as network healing choices determined by the Spanning Tree Protocol (STP) that have a longer than ideal number of hops through the network.

- Standard Ethernet does not have redundant paths to allow network recovery with no loss of data. STP and Rapid Spanning Tree Protocol (RSTP) are used to prevent network loops and to determine alternate network paths in the event of a link failure but take time to converge, resulting in packet loss during the healing process. Newer protocols such as Parallel Redundancy Protocol (PRP) and High-Availability Seamless Redundancy (HSR) have been added to Ethernet to address this issue.

- Healing times for standard Ethernet are based on STP and RSTP algorithms. This results in relatively slow healing times that increase with the number of nodes in a network. Many manufacturers have developed proprietary algorithms to gain advantages in healing times.

- It is not possible to effectively model and test different Ethernet network configurations and fault conditions. Testing typically requires physically breaking network connections and analyzing traffic statistics to determine the correct operation. It is not possible to test all fault conditions.



Network Configuration

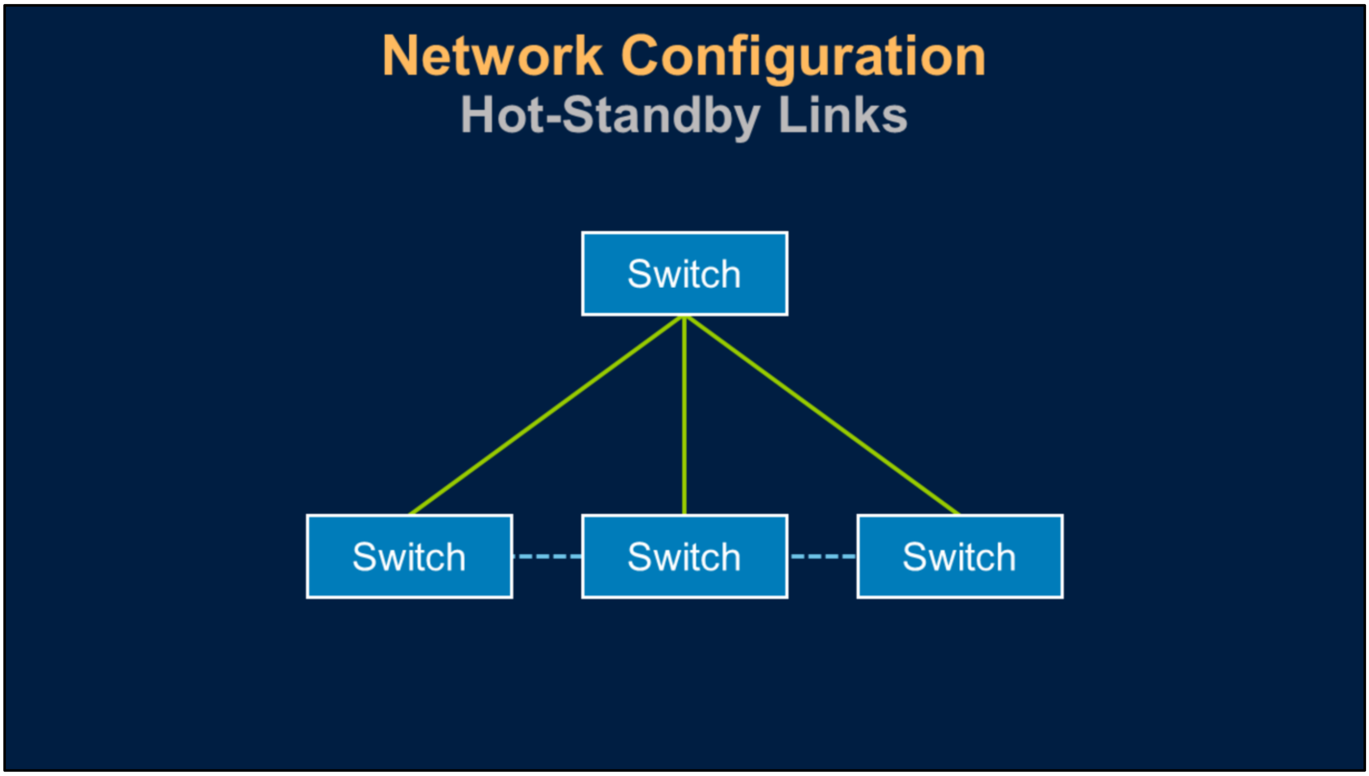
No Redundancy

Switch

Switch

Switch

Switch



Network Configuration

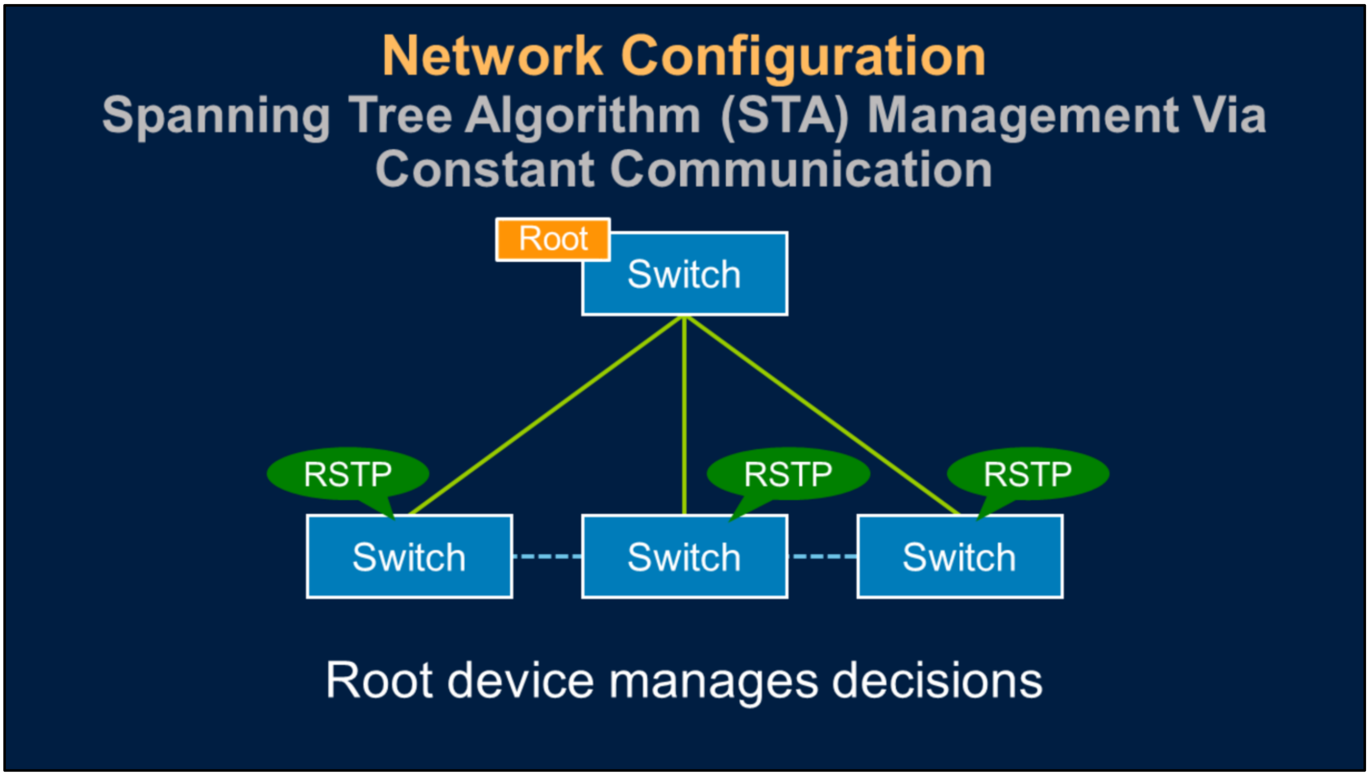
Hot-Standby Links

Switch

Switch

Switch

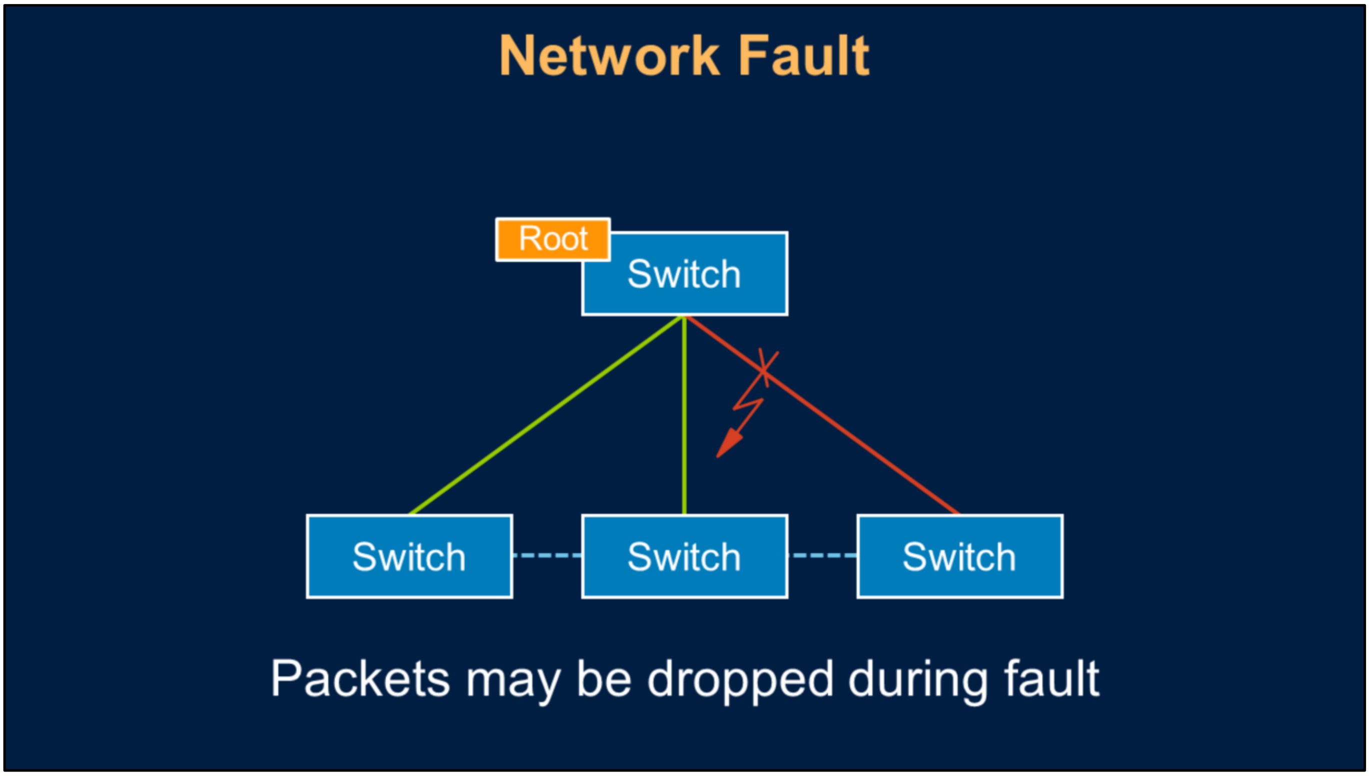
Switch



Network Configuration

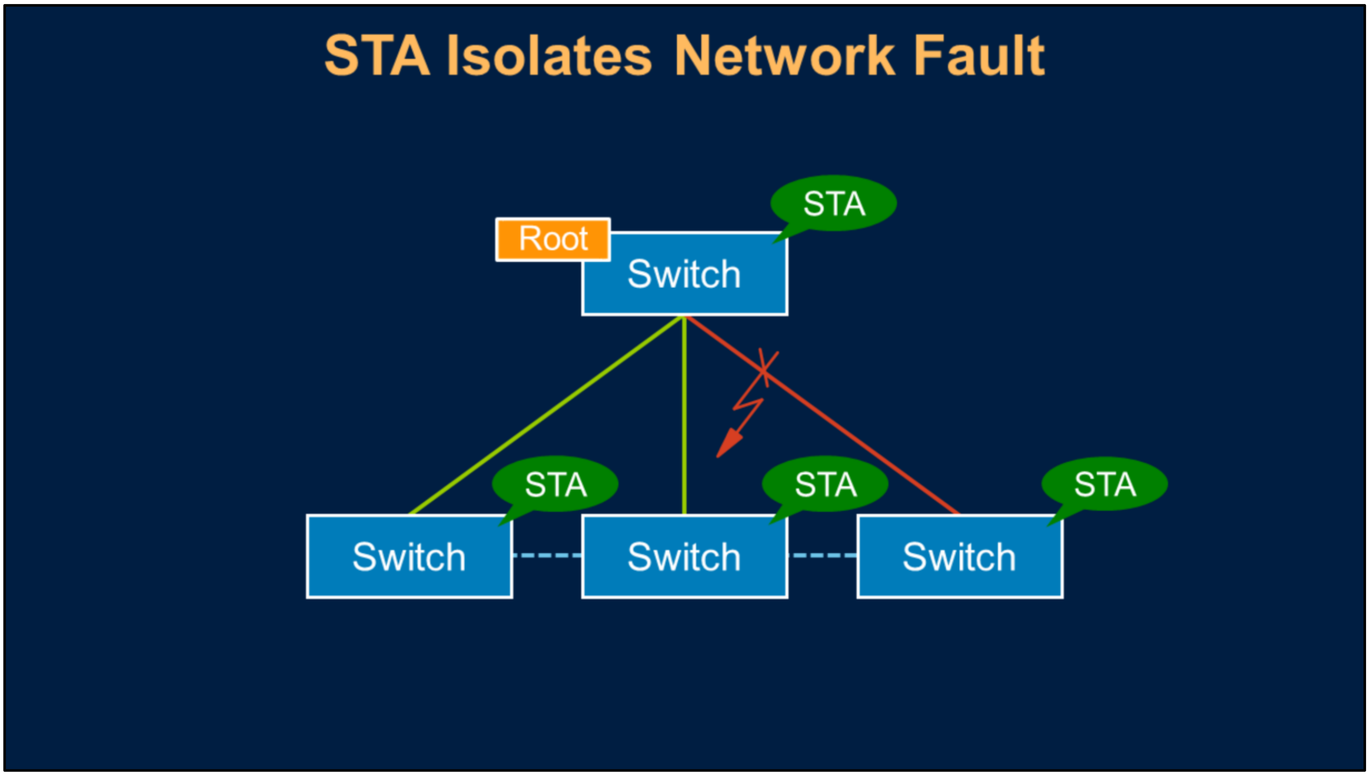
Spanning Tree Algorithm (STA) Management Via Constant Communication

Root device manages decisions

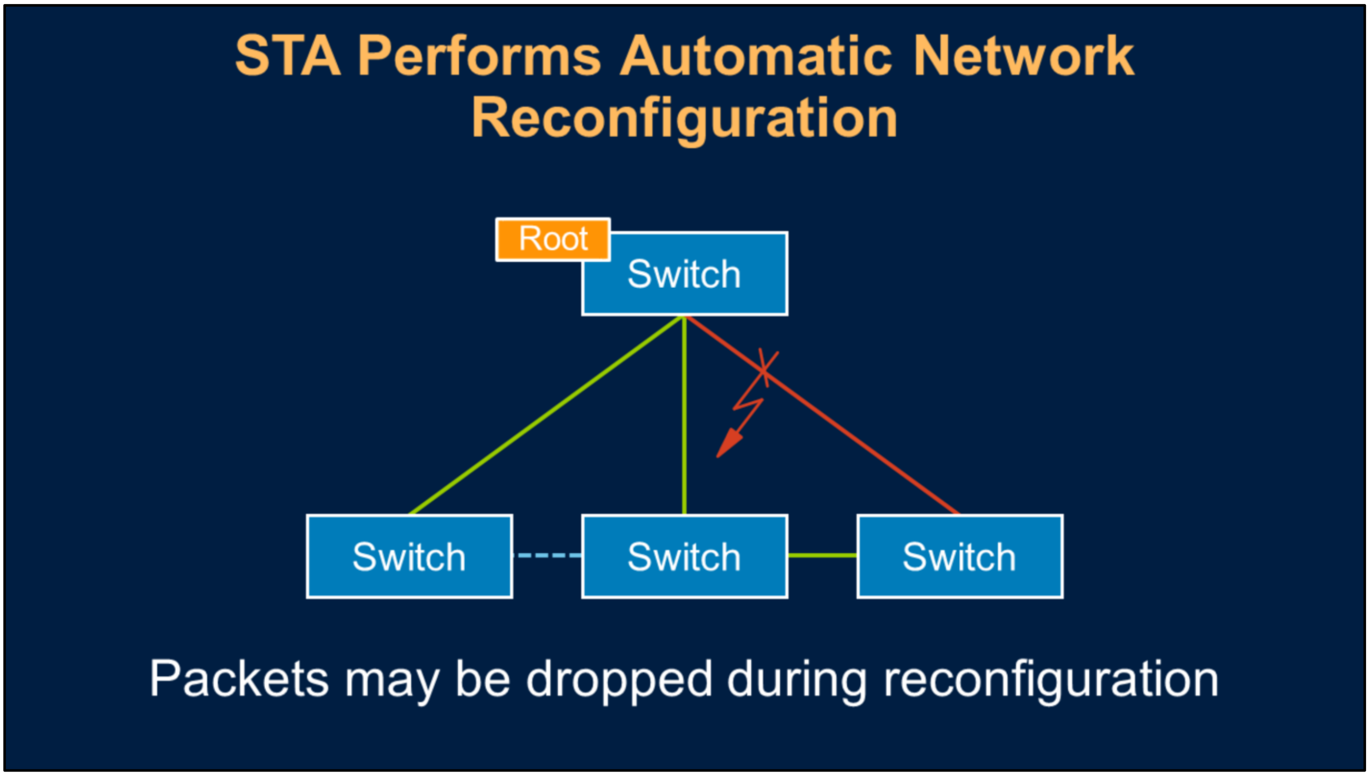


Network Fault

Packets may be dropped during fault

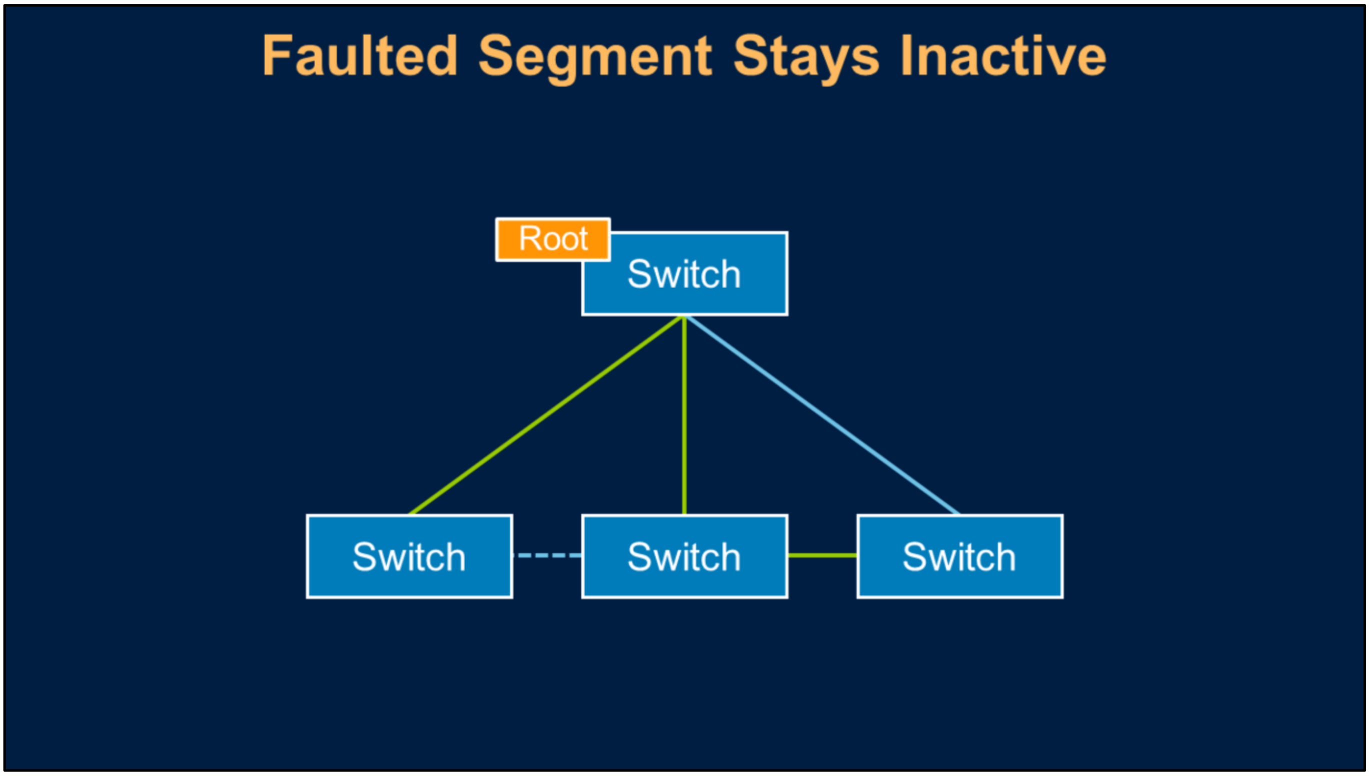


STA Isolates Network Fault

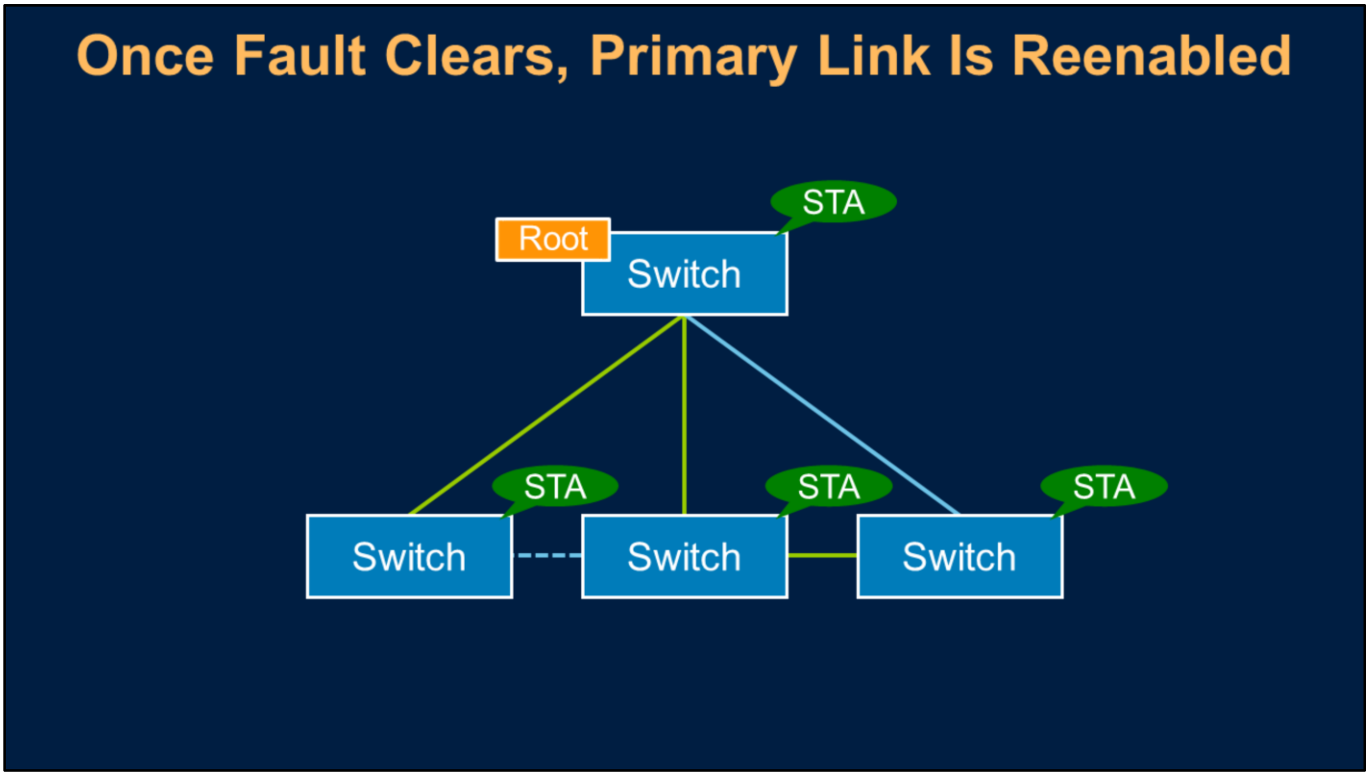


STA Performs Automatic Network Reconfiguration

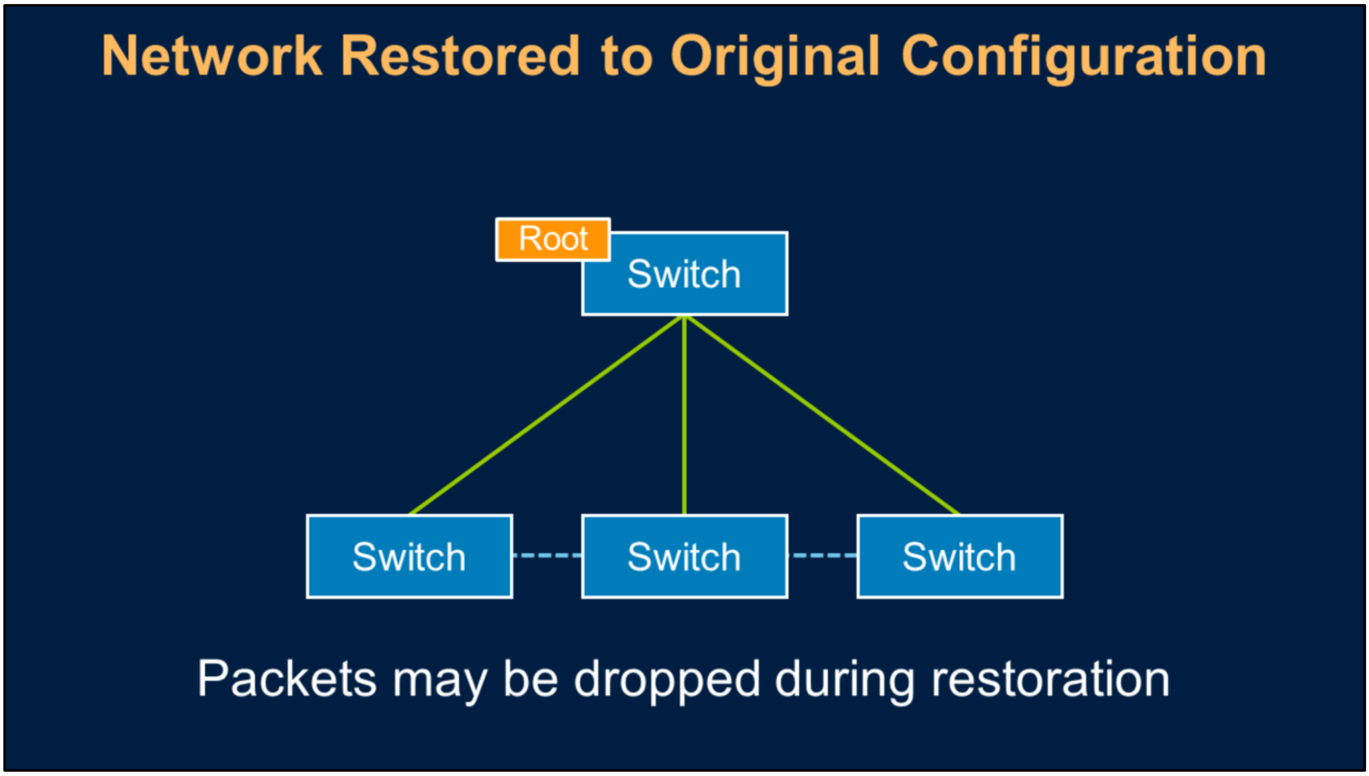
Packets may be dropped during reconfiguration



Faulted Segment Stays Inactive

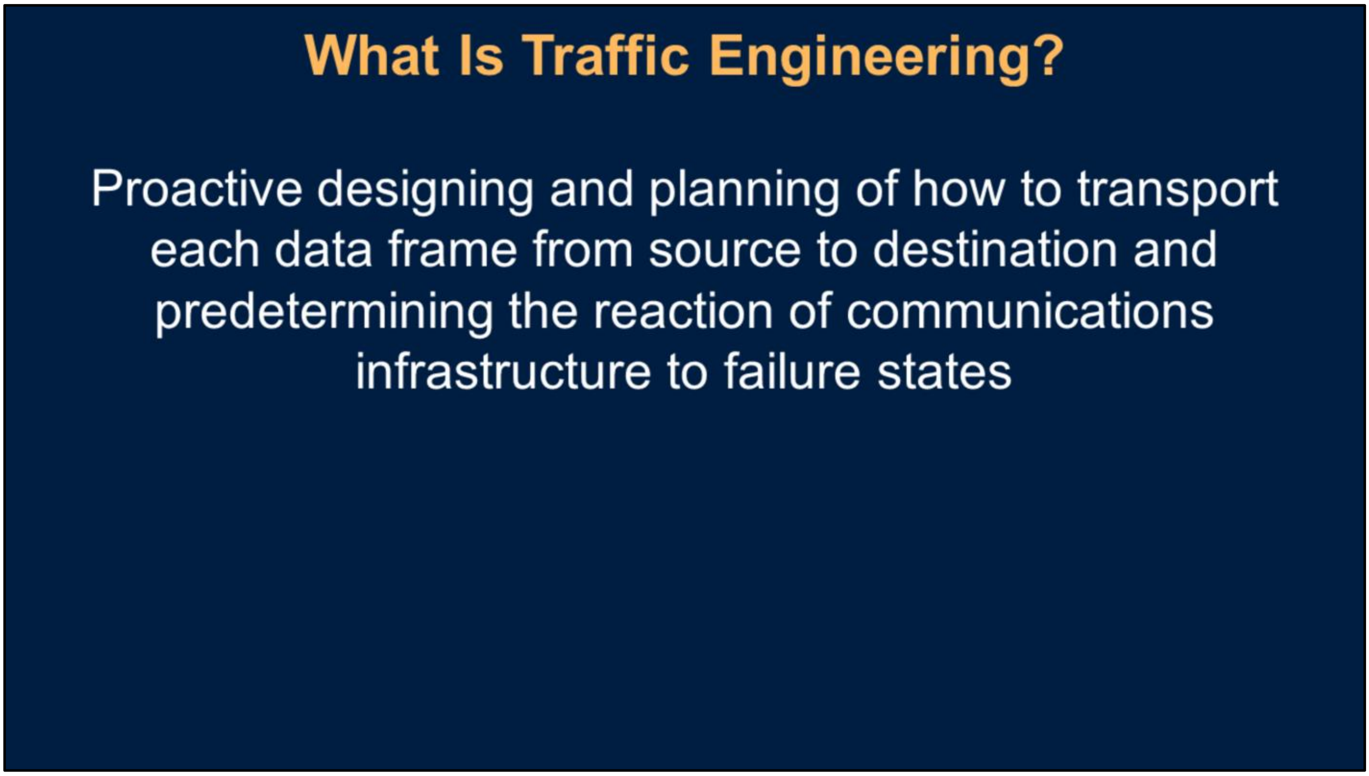


Once Fault Clears, Primary Link Is Reenabled



Network Restored to Original Configuration

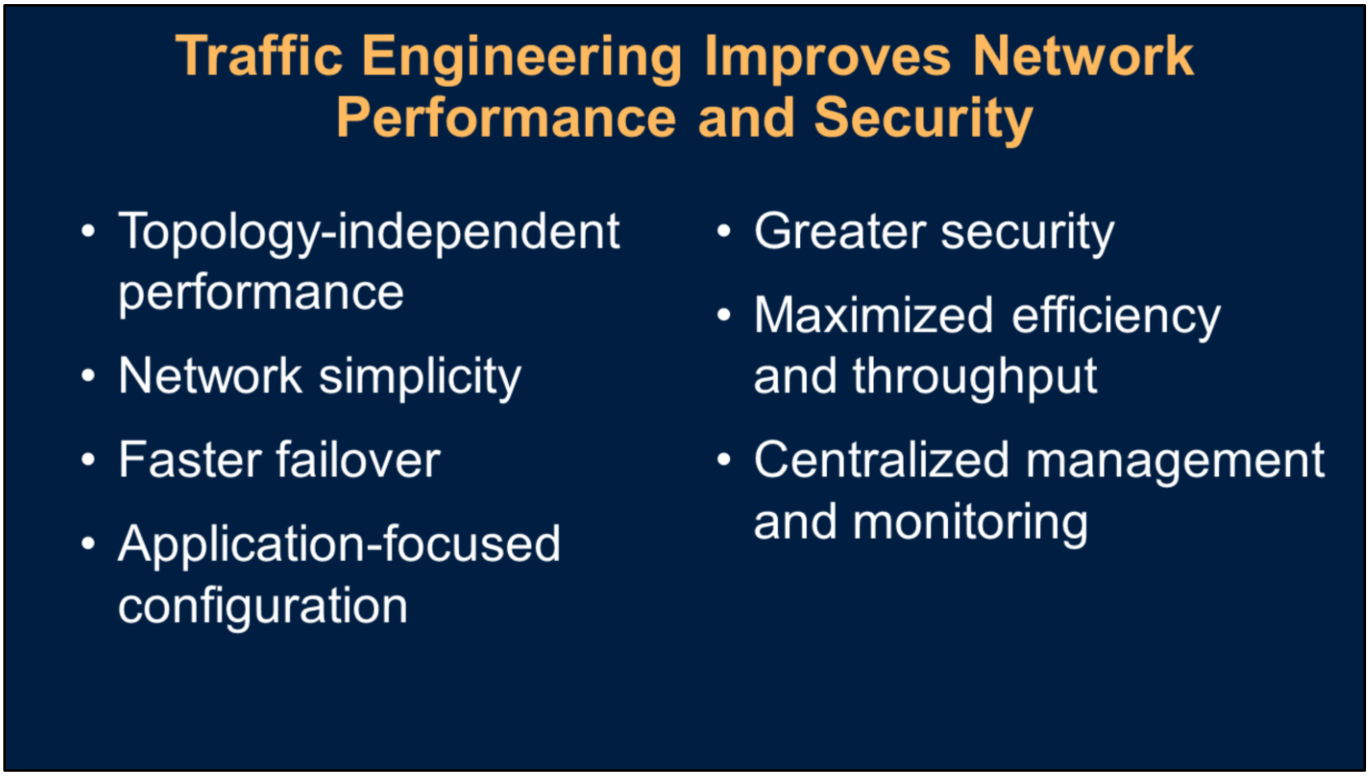
Packets may be dropped during restoration



What Is Traffic Engineering?

Proactive designing and planning of how to transport each data frame from source to destination and predetermining the reaction of communications infrastructure to failure states

This slide provides the definition of traffic engineering, which is a key principle in communications network design. It provides the mechanisms required to ensure that circuits meet specific performance criteria required by the applications running across the network.



Traffic Engineering Improves Network Performance and Security

- Topology-independent performance
- Network simplicity
- Faster failover
- Application-focused configuration
- Greater security
- Maximized efficiency and throughput
- Centralized management and monitoring

Traffic engineering provides the following benefits:

- Topology-independent performance. Depending on the communications network technology, different topologies affect the latency, throughput, and determinism of traffic. The addition of new switches or multiplexers can change the performance of a circuit. Traffic engineering allows circuit performance to be maintained regardless of network topology and network changes.

- Network simplicity. Understanding how network topologies can affect circuit performance and how to minimize potential problems can get complicated. Using traffic engineering removes this complexity.

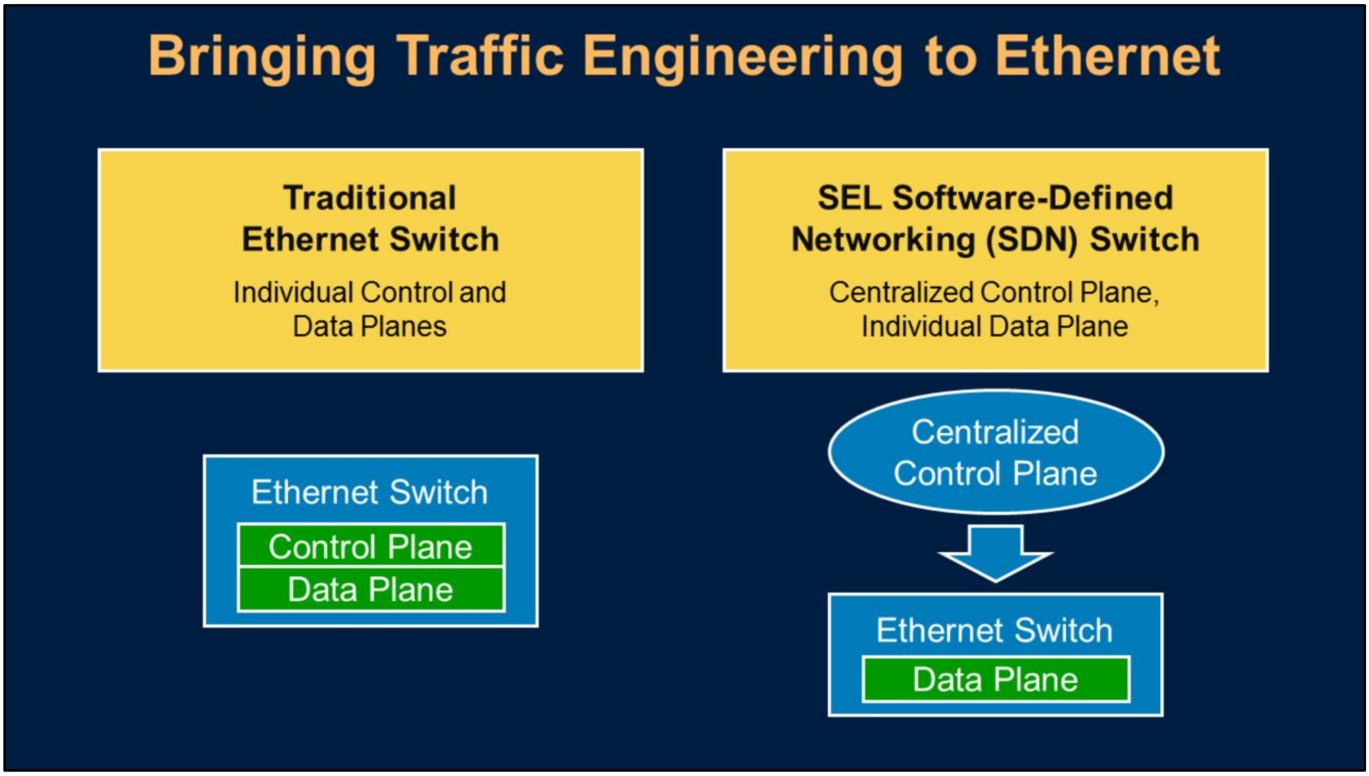
- Faster failover. Technologies like Ethernet use protocols such as STP to manage network failover in the event of a cable or fiber break. These protocols have convergence times that are dependent on the failure path, network topology, and network size. Traffic engineering allows primary and failover paths to be specified for each circuit, thereby reducing failover times.

- Application-focused configuration. Circuits can be implemented with performance characteristics that are defined based on the application.

- Greater security. Traffic engineering provides enhanced network security that includes rule-based, application-specific attributes and deny-by-default policies.

- Maximized efficiency and throughput. With traffic engineering, the available network bandwidth and capacity can be optimized to make the most efficient use of available resources.

- Centralized management and monitoring. This attribute supplies visibility of the network status to the network manager and provides a centralized resource for configuring and managing changes to the network.



Bringing Traffic Engineering to Ethernet

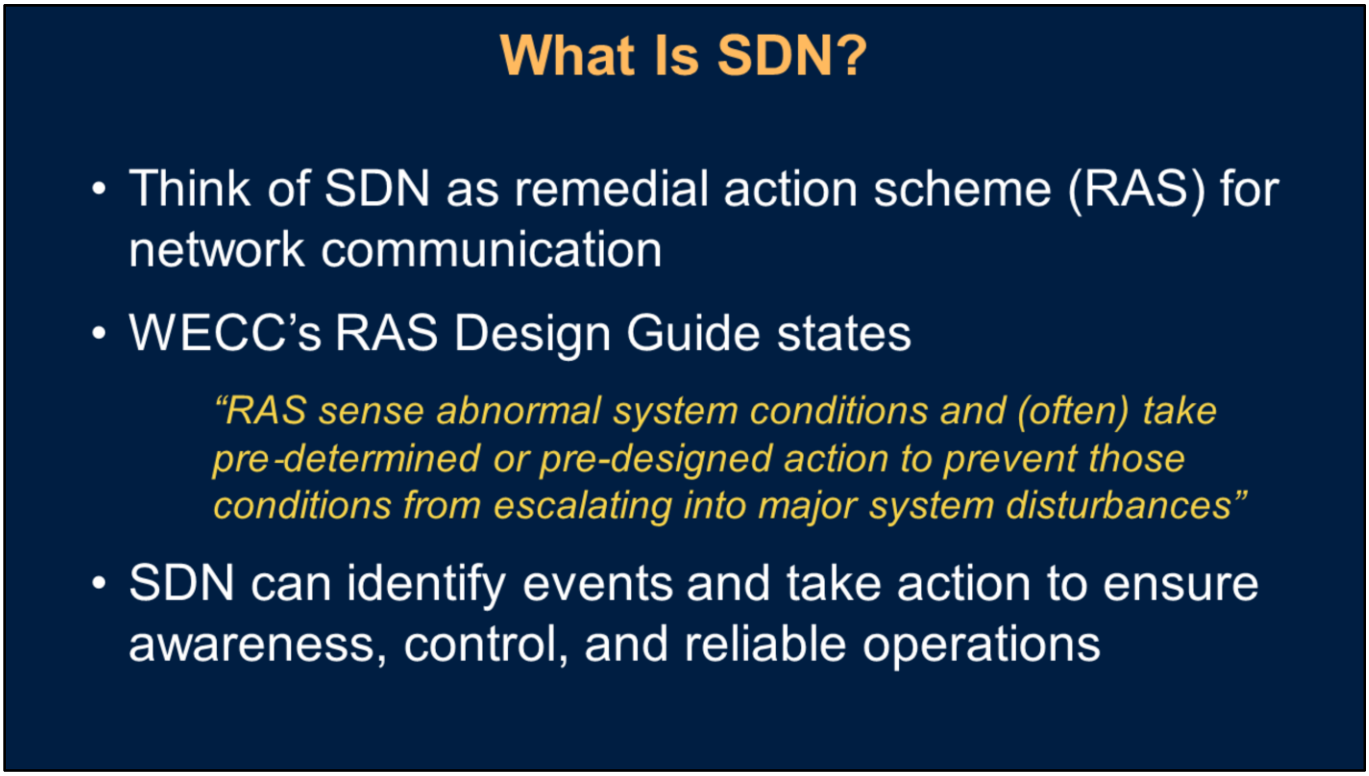
Traditional Ethernet Switch

Individual Control and Data Planes

SEL Software-Defined Networking (SDN) Switch

Centralized Control Plane, Individual Data Plane

Software-defined networking (SDN) is a new approach to the management, configuration, and operation of network systems. In a traditional Ethernet network deployment, application data must flow in the direction of the routes determined by the routing and switching protocols. Each network device (switch or router) has integrated control logic and data-forwarding logic. The control plane in a traditional network is distributed in the switching fabric (network devices), and as a consequence, changing the forwarding behavior of a network involves changing the configurations of many (potentially all) network devices. SDN is a new architecture in networking that simplifies network management by abstracting the control plane from the data plane.



What Is SDN?

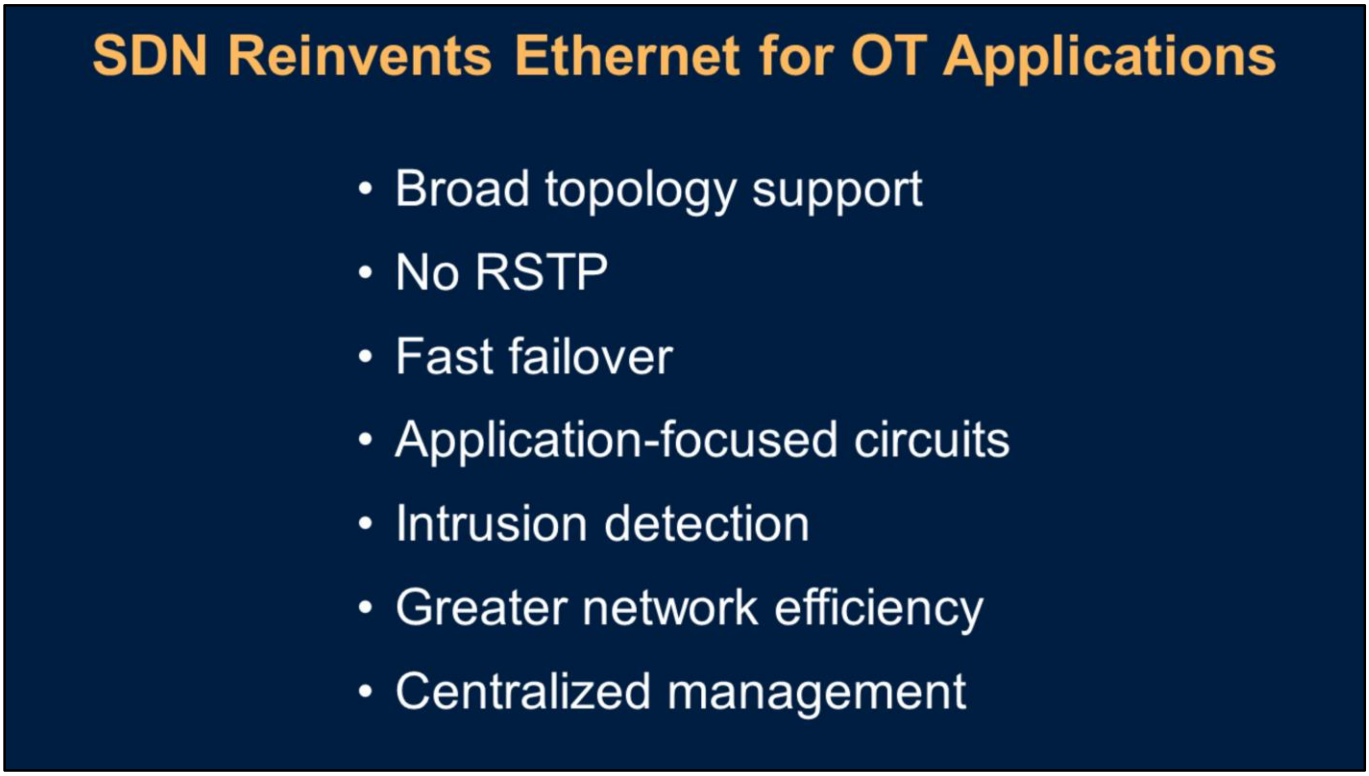
- Think of SDN as remedial action scheme (RAS) for network communication
- WECC's RAS Design Guide states

> *"RAS sense abnormal system conditions and (often) take pre-determined or pre-designed action to prevent those conditions from escalating into major system disturbances"*

- SDN can identify events and take action to ensure awareness, control, and reliable operations

SDN enables the network to be engineered with the same professional practices as the power systems. The network can be pre-engineered and configured not only to all the primary communications flow paths but for every failure case as well. This improves the network performance by enabling the devices to know what to do when any of the failure cases happen without the need to "discover" what should be done. This also provides a reliability validation to network engineering that has been difficult to achieve. Network reliability can be engineered and exercised systematically to validate that the performance conditions are achieved under all expected system states.

For example, a remedial action scheme (RAS) takes into account power flow between multiple sources to the region's loads and adjusts power flow paths based on the condition of the power system assets if they are safe and operational or if there are fault conditions that must be taken out of service. SDN allows these engineering principles to be applied to network engineering to keep as many communications flows up and as operational as possible during any fault condition.

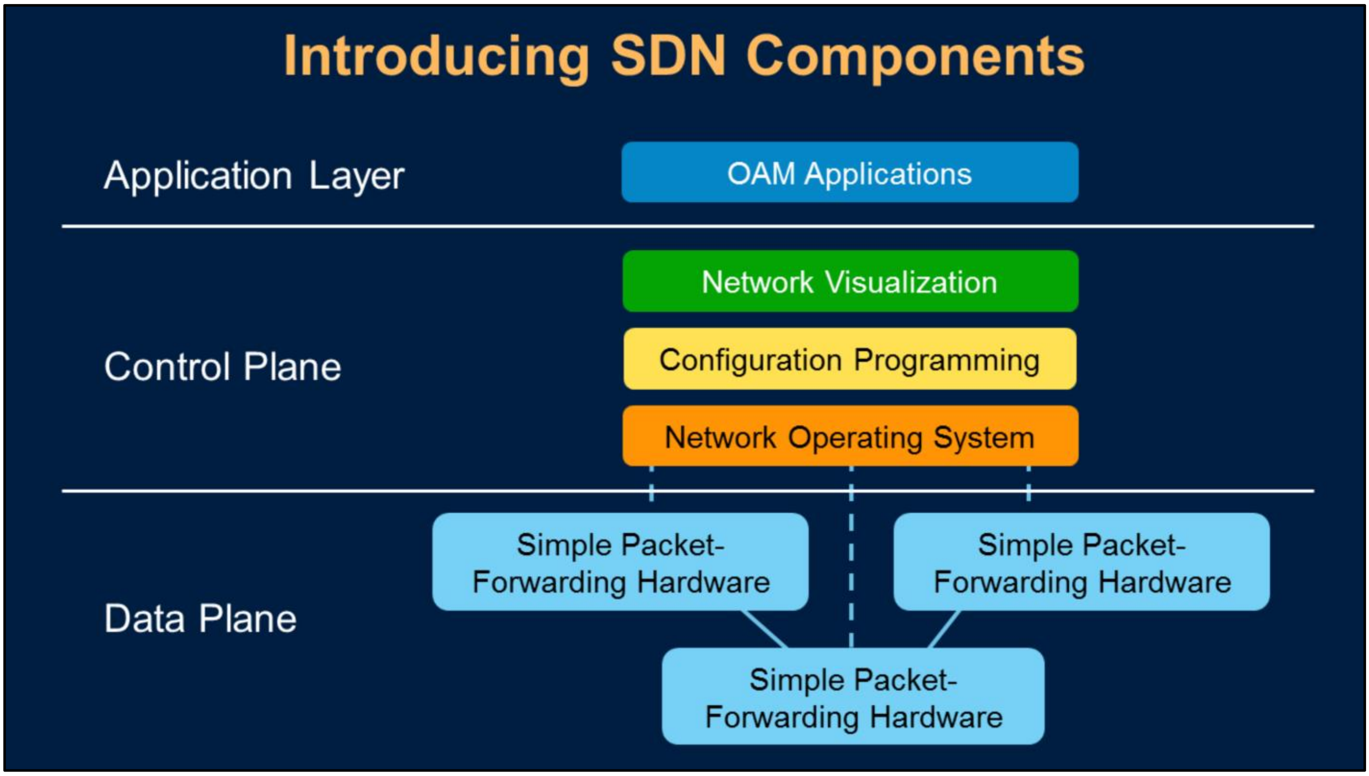


SDN Reinvents Ethernet for OT Applications

- Broad topology support
- No RSTP
- Fast failover
- Application-focused circuits
- Intrusion detection
- Greater network efficiency
- Centralized management

Critical infrastructure operational technology (OT) networks require high reliability, deny-by-default security, guaranteed latencies, and determinism. SDN addresses the deficiencies of traditional Ethernet for real-time OT applications by performing the following:

- Supporting a wider range of network topologies that are driven by the applications and services using the network.
- Removing dynamic protocols such as RSTP and Open Shortest Path First (OSPF) that increase network design complexity.
- Supporting faster failover through the pre-engineering of primary and backup paths. With SDN, the switches know the backup path for a failure on the primary path and automatically forward the next packet to the backup path without requiring RSTP protocols to determine a failover network configuration. This enables faster healing times; SDN provides millisecond network healing times.
- Enabling each circuit or flow to be configured based on the applications using the network.
- Providing stronger security by using matching rules that only allow approved services and denying all other data traffic by default.
- Maximizing network efficiency and throughput. With no blocking ports, all links can be used for flow circuits. Flow transports can be balanced by maximizing the use of every port and link on the system. Traffic congestion can be controlled by logically and physically isolating high-priority flows from lower-priority flows.
- Providing centralized network management and monitoring, which provides performance information and visibility of the entire network.



Introducing SDN Components

Application Layer

OAM Applications

Control Plane

Network Visualization

Configuration Programming

Network Operating System

Data Plane

Simple Packet-Forwarding Hardware

Simple Packet-Forwarding Hardware

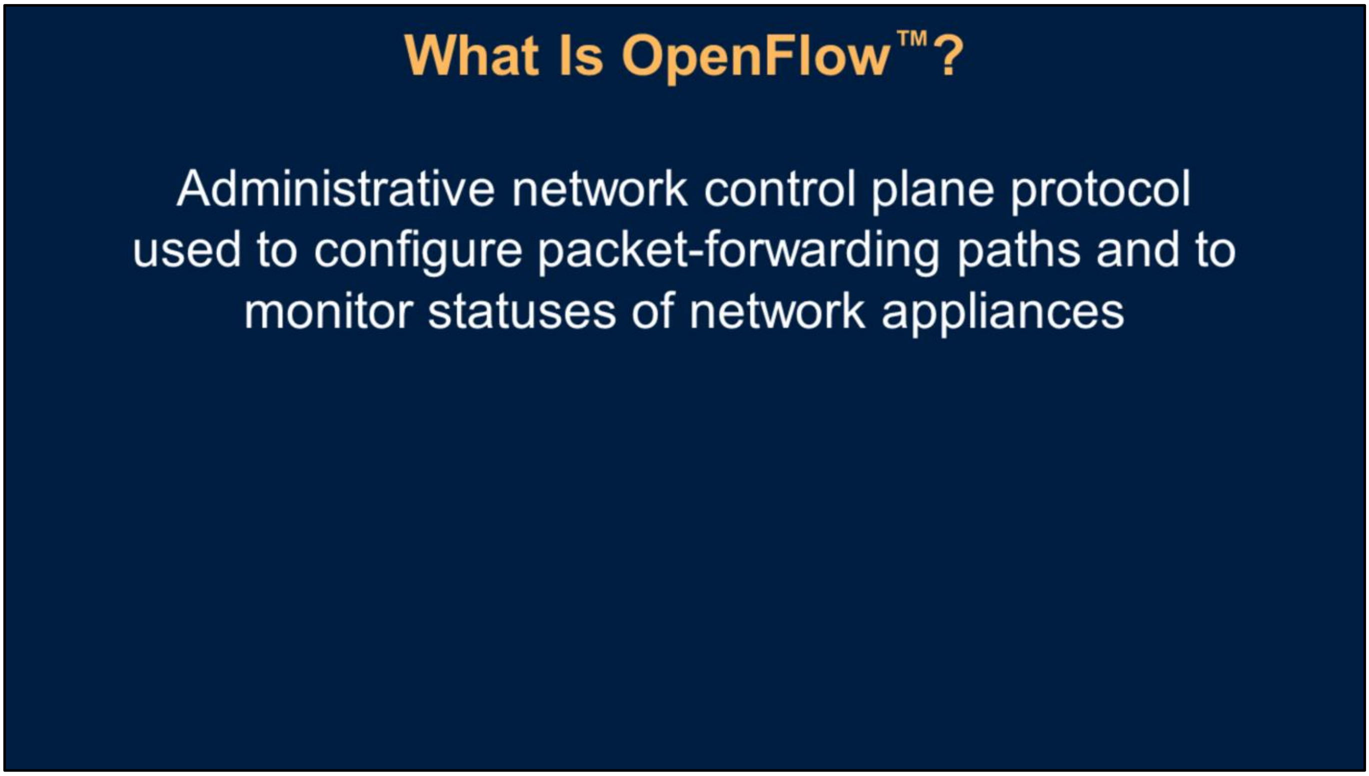
Simple Packet-Forwarding Hardware

The heart of SDN is the SDN controller that contains the control plane. The controller software determines how packets should flow or be forwarded in the network. The controller communicates this information to the network devices, which compose the data plane, by setting their forwarding tables. This enables centralized configuration and management of the network.

The data plane is composed of network devices that replace switches and routers. In SDN, these devices are very simple Ethernet packet-forwarding devices, each with a communications interface to the controller to receive forwarding information.

In the SDN architecture, the controller can identify an application programming interface (API) that other services or applications can use to configure the network. This is represented by the application layer in the diagram shown on the slide.

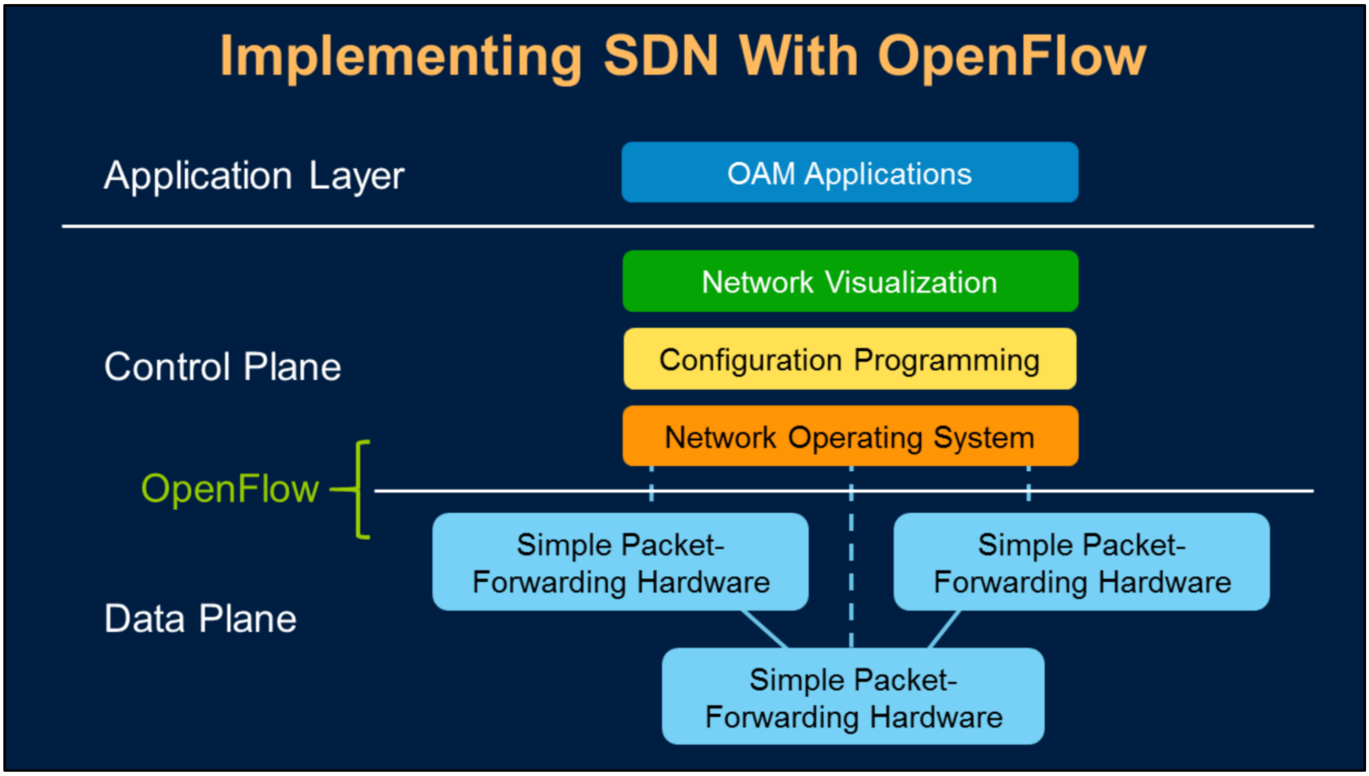
The application layer can be tailored to meet the requirements of the application domain, such as a data center networking application or an automation and control networking application. These applications are generically referred to as operations, administration, and maintenance (OAM) functions.



What Is OpenFlow™?

Administrative network control plane protocol used to configure packet-forwarding paths and to monitor statuses of network appliances

SDN requires a communications interface between the network devices and the controller. A standardized interface allows a controller to interoperate with different types of network devices. The OpenFlow™ protocol is a standardized interface that is managed by the Open Networking Foundation (ONF) and has been adopted by major switch and router manufacturers.



Implementing SDN With OpenFlow

Application Layer

OAM Applications

Control Plane

Network Visualization

Configuration Programming

Network Operating System

OpenFlow

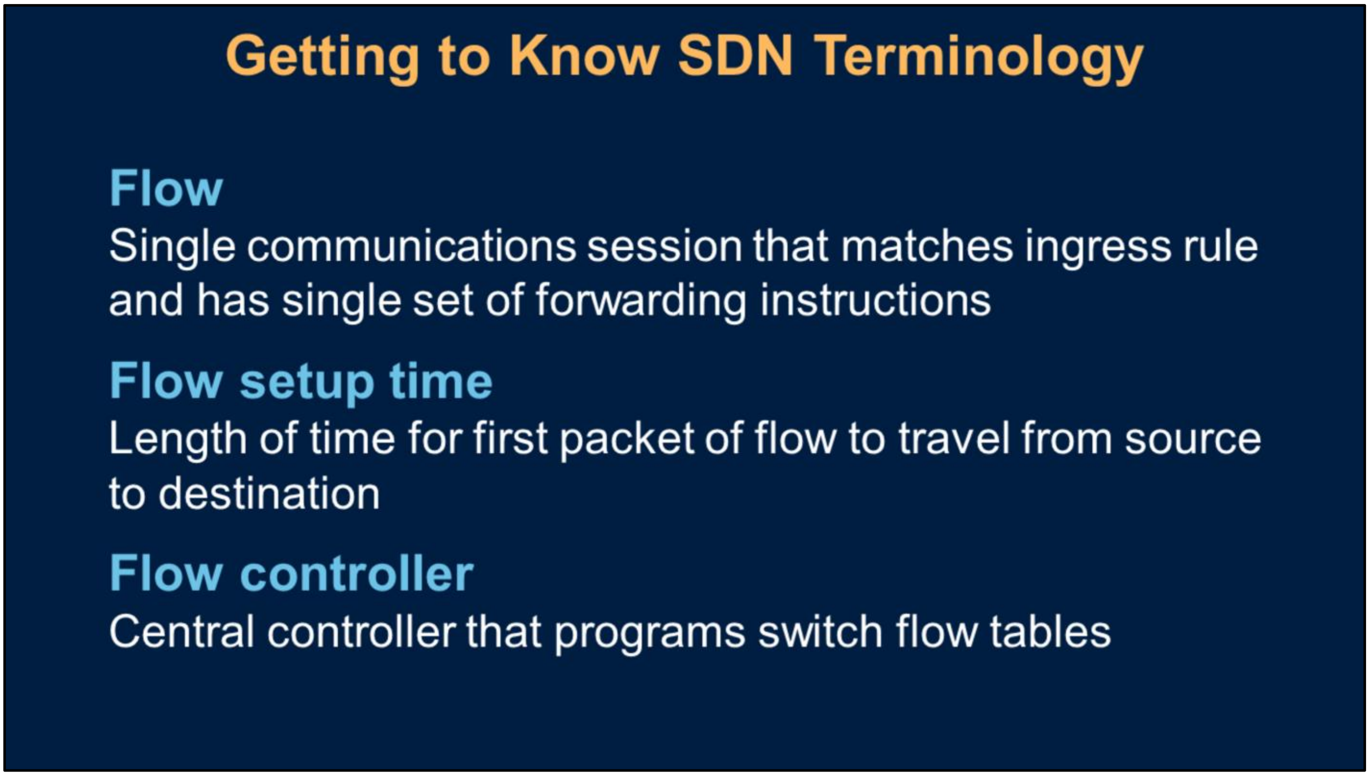
Data Plane

Simple Packet-Forwarding Hardware

Simple Packet-Forwarding Hardware

Simple Packet-Forwarding Hardware

OpenFlow is an interface standard between the control plane and data plane, and it can be considered a building block in the overall SDN architecture.



Getting to Know SDN Terminology

Flow

Single communications session that matches ingress rule and has single set of forwarding instructions

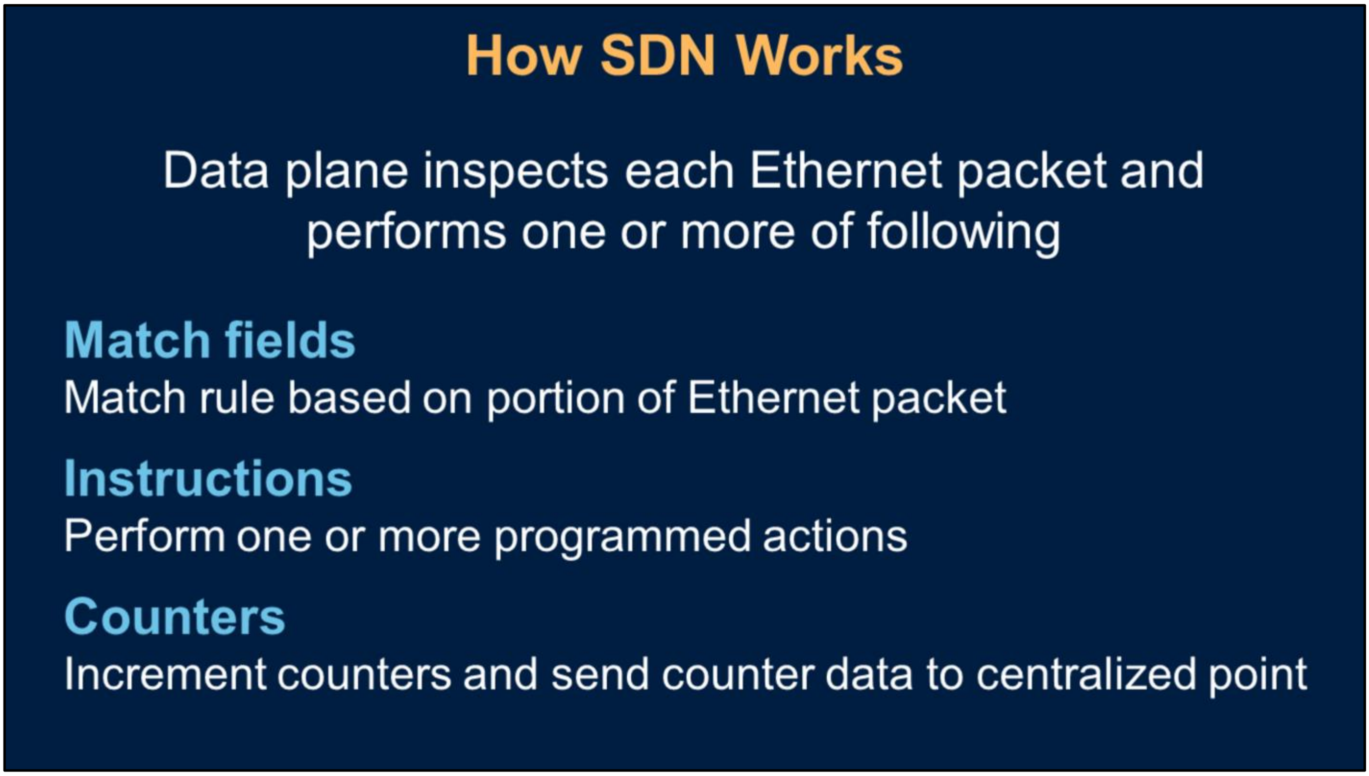
Flow setup time

Length of time for first packet of flow to travel from source to destination

Flow controller

Central controller that programs switch flow tables

There are many new terms that SDN has introduced. This slide explains the terms flow, flow setup time, and flow controller.



How SDN Works

Data plane inspects each Ethernet packet and performs one or more of following

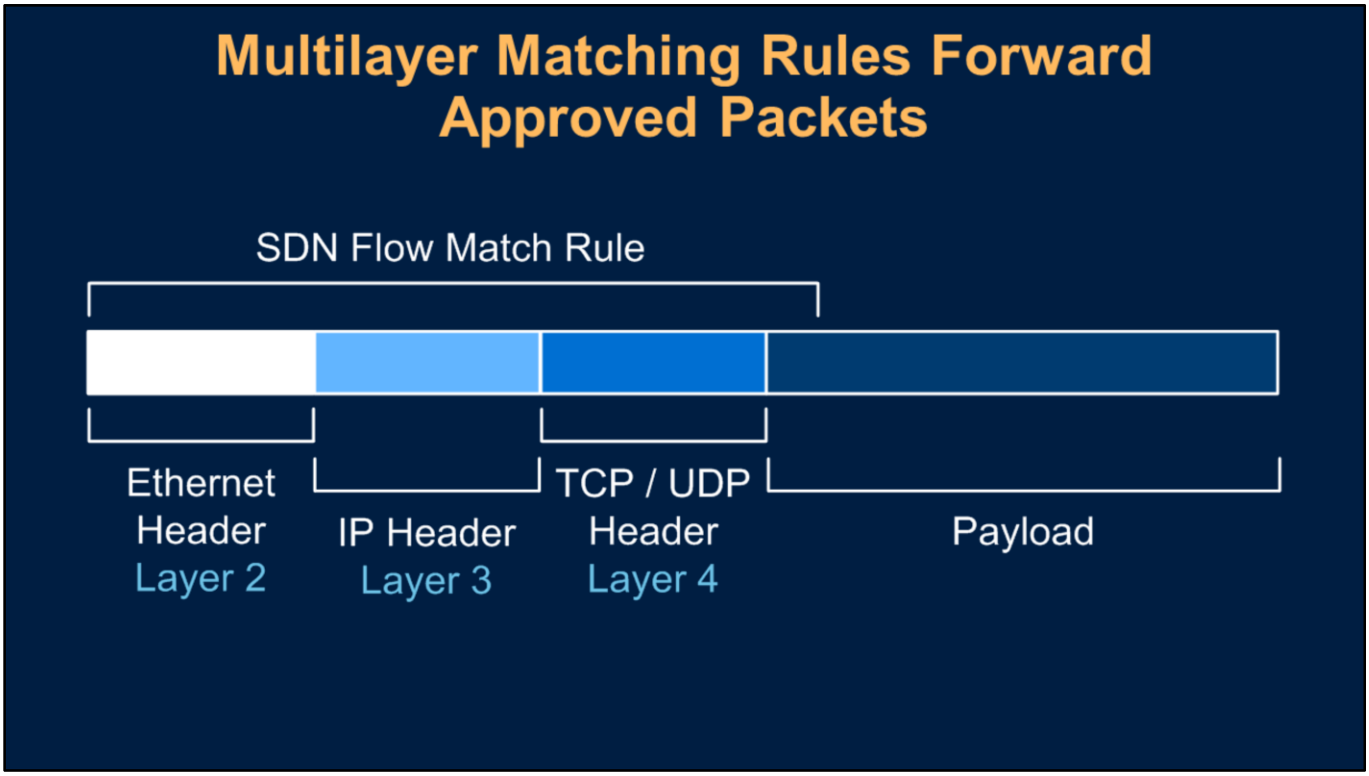
Match fields
Match rule based on portion of Ethernet packet

Instructions
Perform one or more programmed actions

Counters
Increment counters and send counter data to centralized point

The SDN switch and controller communicate via the OpenFlow protocol. The switch is given a set of rules by the controller that is used to determine which operations should be performed on each packet received. Each flow table in the switch contains a set of flow entries, and each flow entry consists of match fields, counters, and a set of instructions or actions to apply to matching packets (such as send out port, modify field, or drop). If a matching entry is found for a packet, the actions associated with the specific flow entry are executed. When an SDN switch receives a packet it has never seen before (i.e., for which it has no matching flow entries), it sends the packet to the controller. The controller then makes a decision on how to handle the packet. It can drop it or add a flow entry that tells the switch how to forward similar packets in the future.

Counters are maintained for each flow entry and used to maintain statistics on traffic by flow, port, queue, and so on.



Multilayer Matching Rules Forward Approved Packets

SDN Flow Match Rule

Ethernet Header Layer 2

IP Header Layer 3

TCP / UDP Header Layer 4

Payload

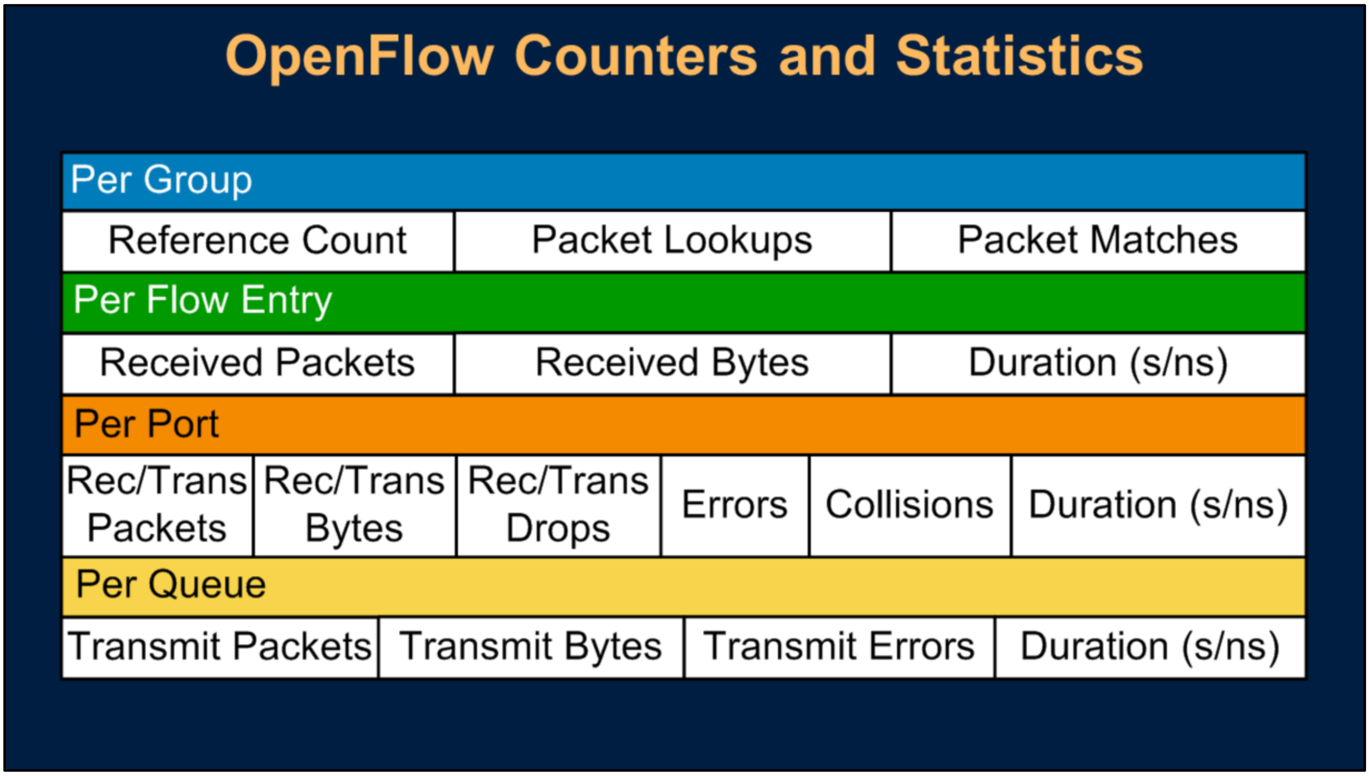
With SDN matching rules, the switch has the ability to look at the first 128 bits of the Ethernet frame and use the information to determine a rule and an associated action. This makes SDN switches multilayer devices that can operate as Layer 1 to Layer 4 switches.

SDN is able to make Layer 1 port association rules to determine how to route packets based on the ingress port and can direct frames to specific egress ports.

The Ethernet header contains the destination and source media access control (MAC) address, Ethertype field, and optional IEEE 802.1Q tag to support virtual local-area networks (VLANs). Layer 2 switches only look at the Ethernet header information.

The IP header contains information on the IP version, source IP address, destination IP address, and time to live. Layer 3 switches or routers use the IP address information contained in the IP header to make routing decisions on packets. A Layer 3 switch also looks at the MAC address information in the Ethernet header to make associations between the IP addresses and MAC addresses of source and destination devices.

The Transmission Control Protocol/User Datagram Protocol (TCP/UDP) header contains protocol information to manage data communications using either TCP, which is a connection-oriented protocol, or UDP, which is a connectionless protocol. Layer 4 devices look at both Layer 3 IP address information and TCP/UDP header information to make switching decisions.



OpenFlow Counters and Statistics

Per Group

Reference Count	Packet Lookups	Packet Matches

Per Flow Entry

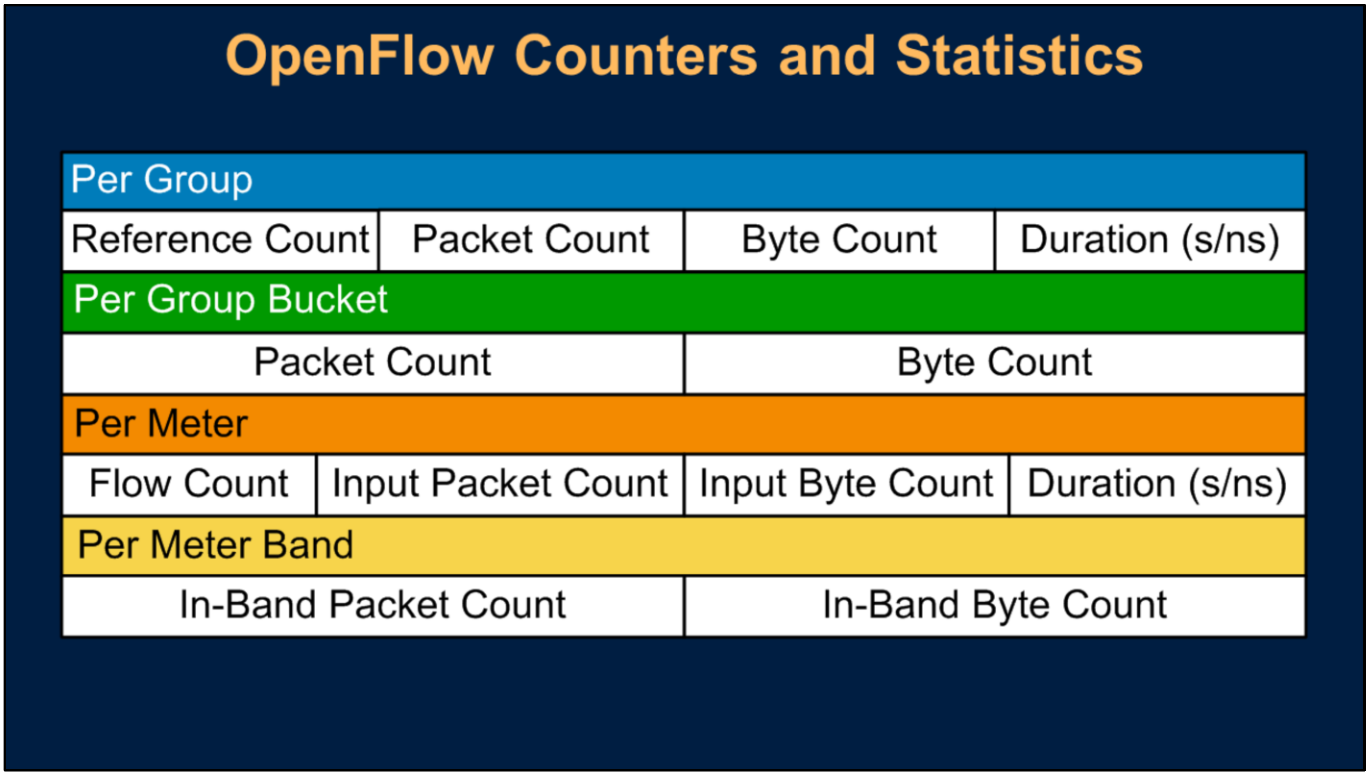
Received Packets	Received Bytes	Duration (s/ns)

Per Port

Rec/Trans Packets	Rec/Trans Bytes	Rec/Trans Drops	Errors	Collisions	Duration (s/ns)

Per Queue

Transmit Packets	Transmit Bytes	Transmit Errors	Duration (s/ns)



OpenFlow Counters and Statistics

Per Group

Reference Count	Packet Count	Byte Count	Duration (s/ns)

Per Group Bucket

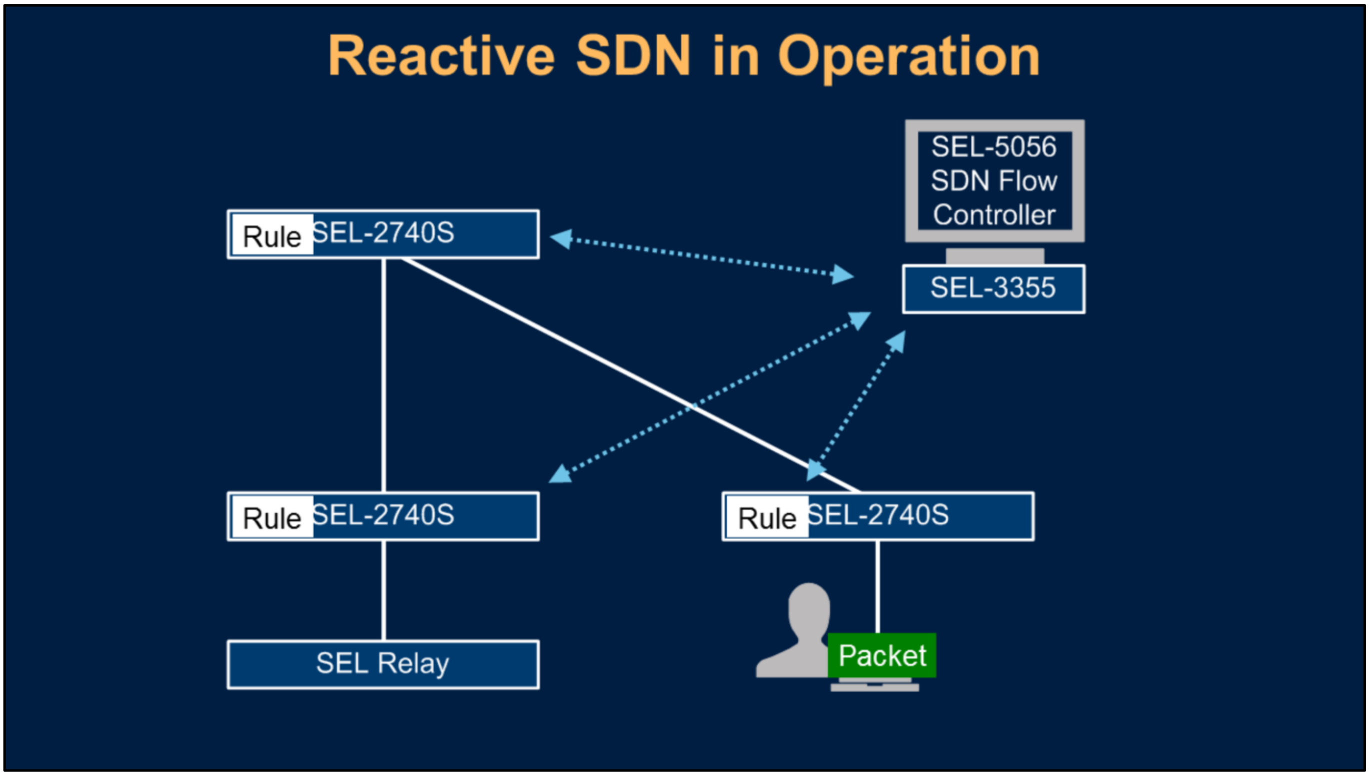
Packet Count	Byte Count

Per Meter

Flow Count	Input Packet Count	Input Byte Count	Duration (s/ns)

Per Meter Band

In-Band Packet Count	In-Band Byte Count



Reactive SDN in Operation

SEL-5056 SDN Flow Controller

SEL-3355

Rule SEL-2740S

Rule SEL-2740S

Rule SEL-2740S

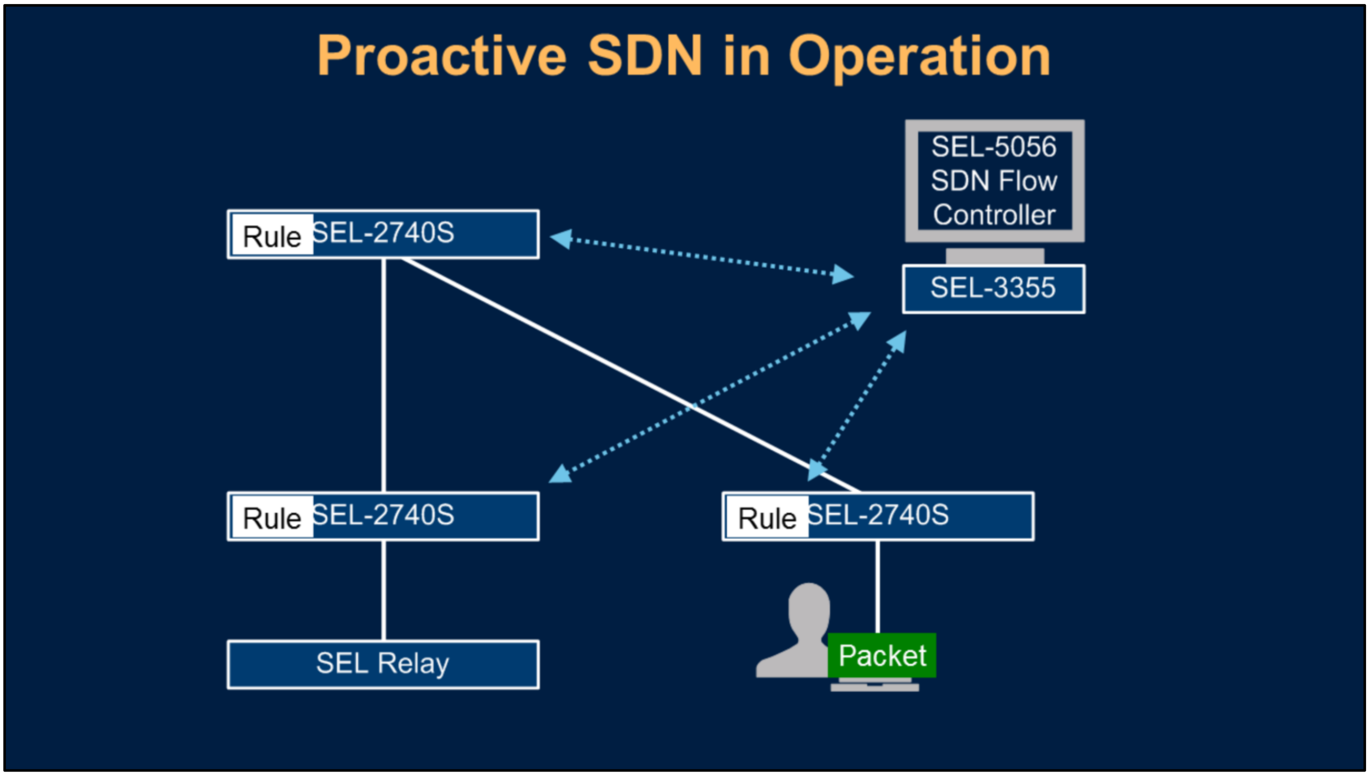
SEL Relay

Packet

The SEL-5056 Software-Defined Network Flow Controller can add, update, and delete flow entries in flow tables both reactively (in response to packets) and proactively.

Reactive operation is when a switch receives a packet that does not match any of the flow table entries. In this situation, the switch forwards the packet to the flow controller. The controller then makes a decision about how to handle the packet. It can drop the packet, or it can add a flow entry that tells the switch how to forward similar packets in the future.

In the example shown on this slide, the SEL-5056 adds a new flow entry to all of the SEL-2740S Software-Defined Network Switches in the network, providing the match rule and action for these types of packets. The example shows an engineering access operation to an SEL relay where there was no flow entry for the type of data packet.



Proactive SDN in Operation

SEL-5056 SDN Flow Controller

SEL-3355

Rule SEL-2740S

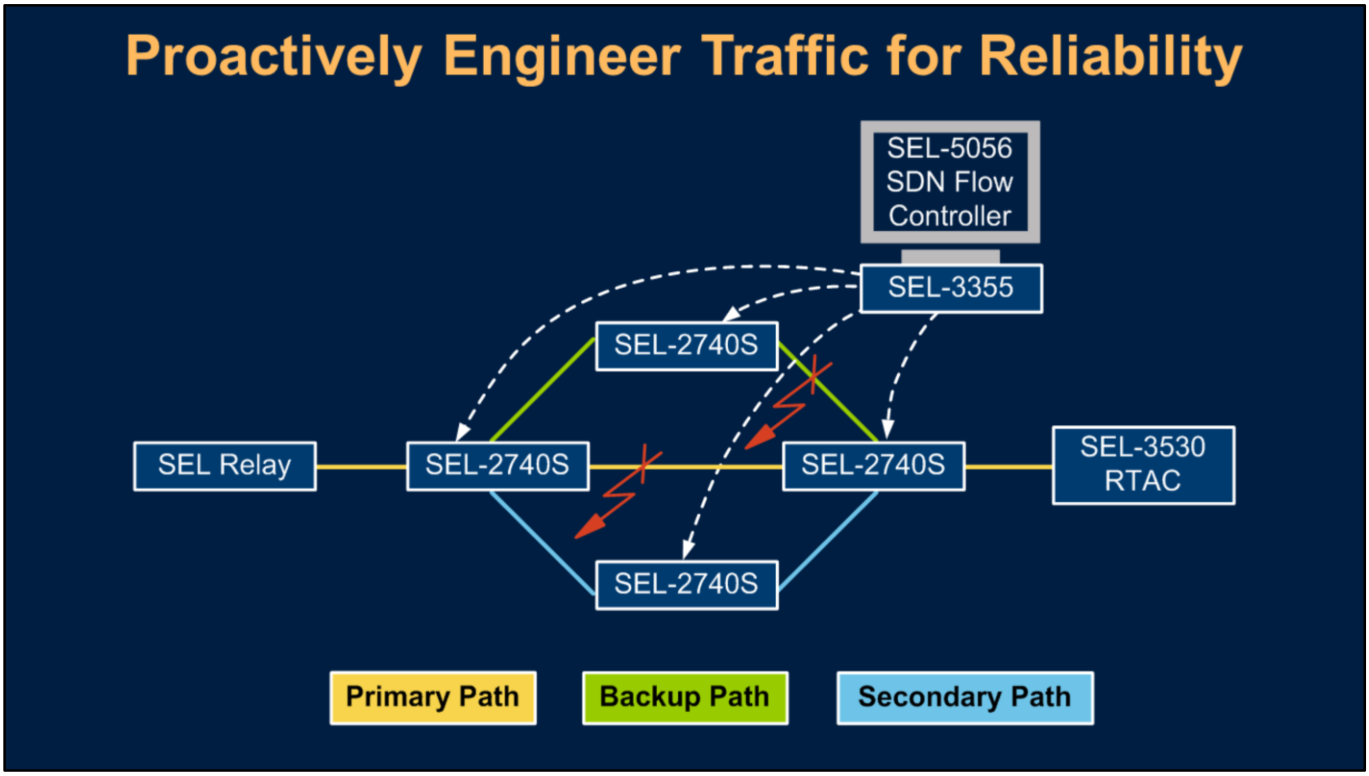
Rule SEL-2740S

Rule SEL-2740S

SEL Relay

Packet

In proactive operation, the flow tables in each switch are updated by the SEL-5056 before engineering access traffic is sent across the network. This method is best suited for the machine-to-machine communications commonly found in industrial control systems.



Proactively Engineer Traffic for Reliability

SEL-5056 SDN Flow Controller

SEL-3355

SEL-2740S

SEL Relay

SEL-2740S

SEL-2740S

SEL-3530 RTAC

SEL-2740S

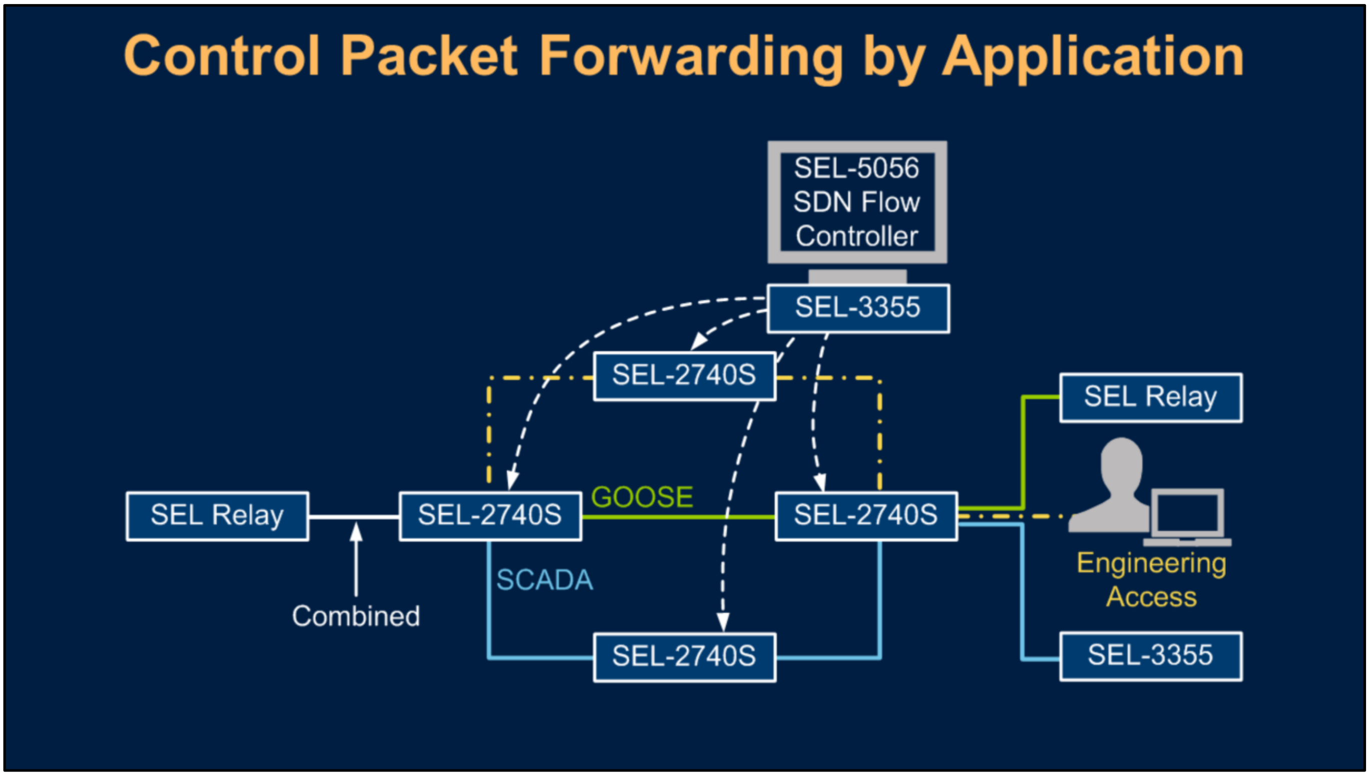
Primary Path

Backup Path

Secondary Path

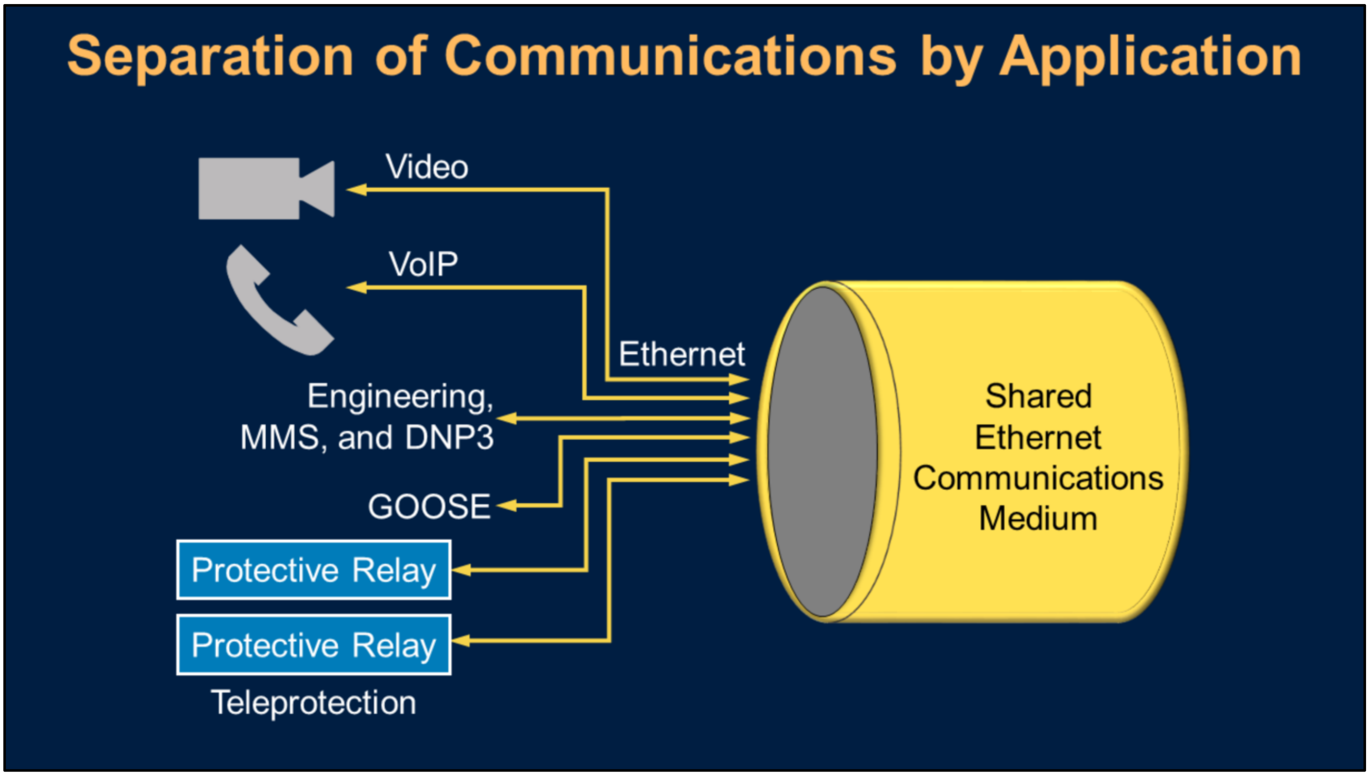
This slide illustrates the traffic engineering of the primary, backup, and secondary paths in a simple network configuration. In the event of a primary path link failure, the network will automatically failover each flow to the backup path. Similarly, if a link problem occurs on the backup path, all flows are reconfigured to the secondary path.

SDN improves the reliability of the network by providing the ability to engineer the primary and backup network paths for each forwarding path on the network. This enables faster link recovery times that are predetermined. Network change control is difficult to manage in standard Ethernet when protocols such as RSTP handle the forwarding decisions based on physical or logical topologies and not the services running on the circuits. SDN allows the forwarding decisions and failover paths to be based on the applications using the network.



Control Packet Forwarding by Application

Because packet forwarding in SDN is based on the applications using the network, it is possible to define different forwarding paths for different applications. In the example shown on this slide, engineering access, supervisory control and data acquisition (SCADA), and Generic Object-Oriented Substation Event (GOOSE) traffic have each been given different forwarding paths across the network.



Separation of Communications by Application

Video

VoIP

Ethernet

Engineering, MMS, and DNP3

GOOSE

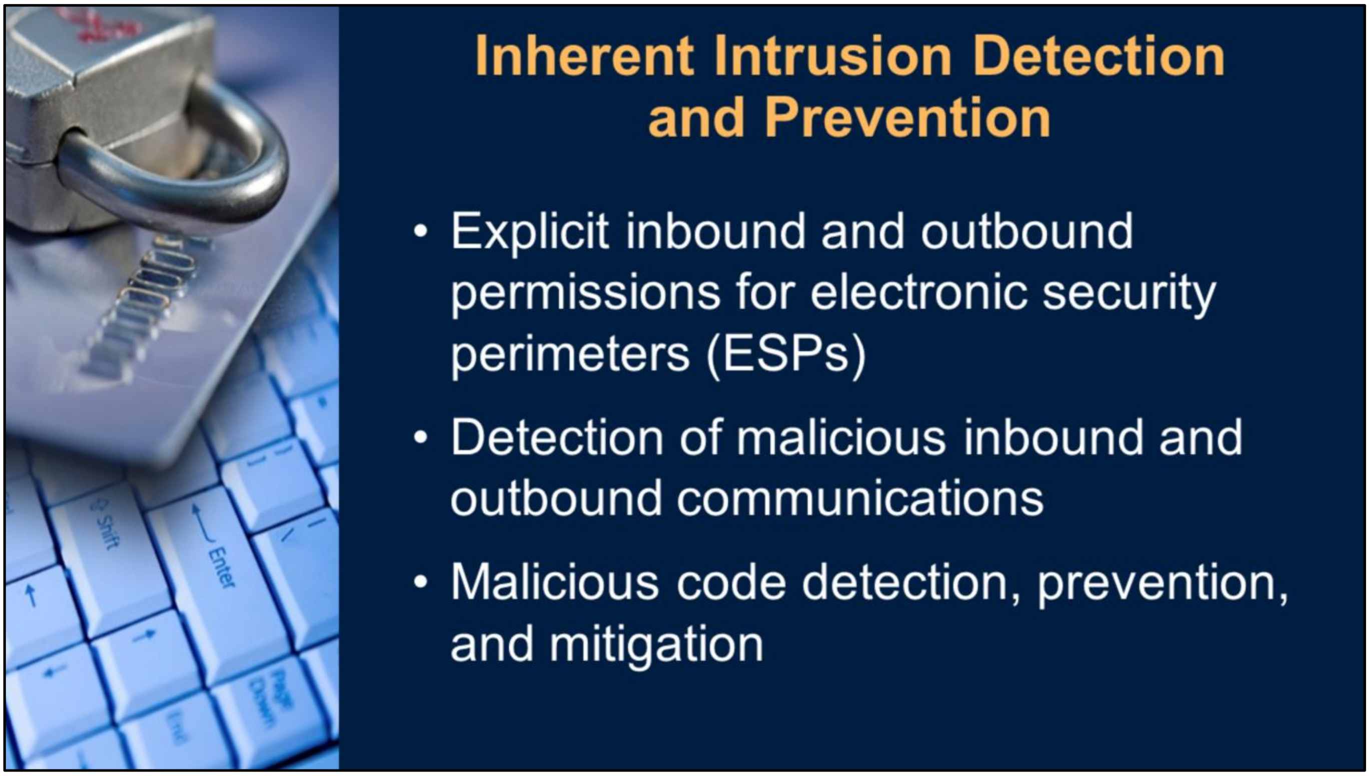
Protective Relay

Protective Relay

Teleprotection

Shared Ethernet Communications Medium

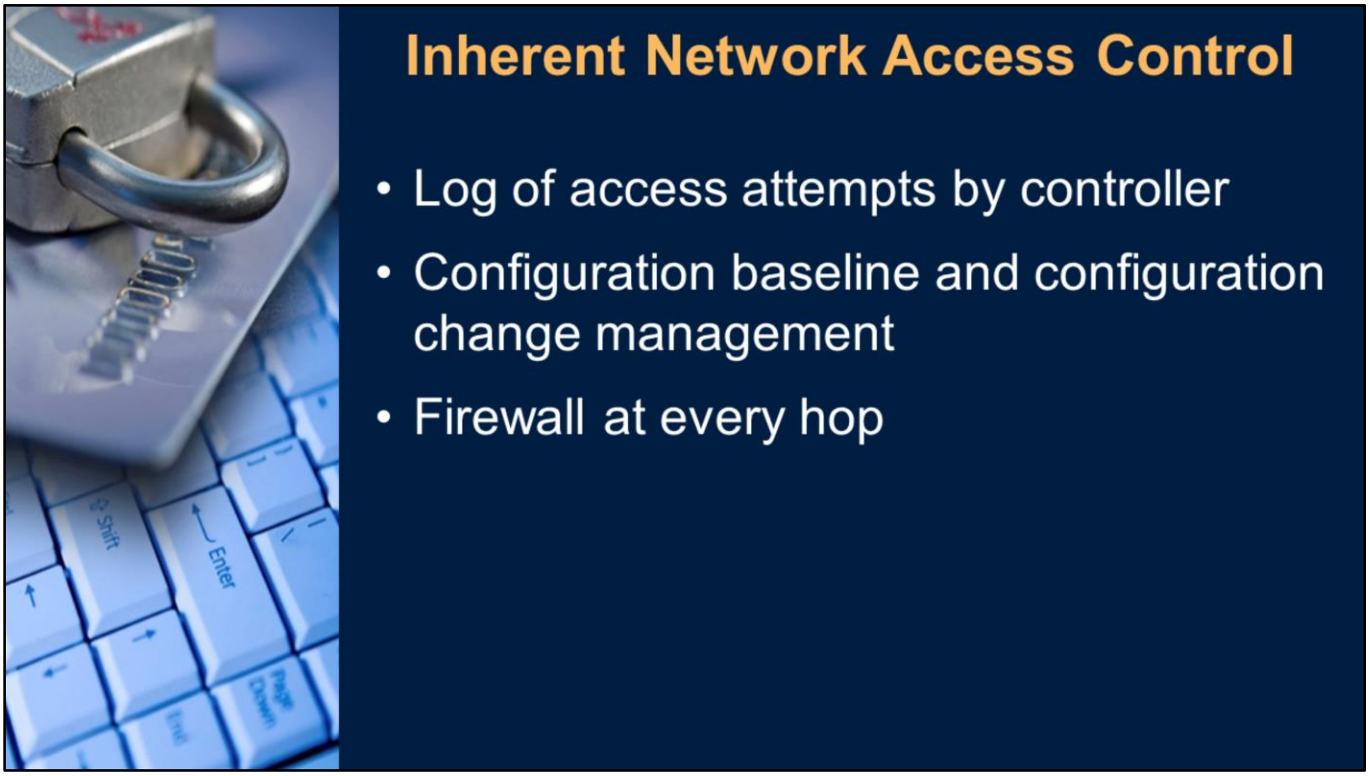
Each application has a corresponding path and subsequently "flows" through the network based on the application for the data. End devices can receive application data from a server over different paths through the industrial control system (ICS) network that are predefined along with backup paths.



Inherent Intrusion Detection and Prevention

- Explicit inbound and outbound permissions for electronic security perimeters (ESPs)
- Detection of malicious inbound and outbound communications
- Malicious code detection, prevention, and mitigation

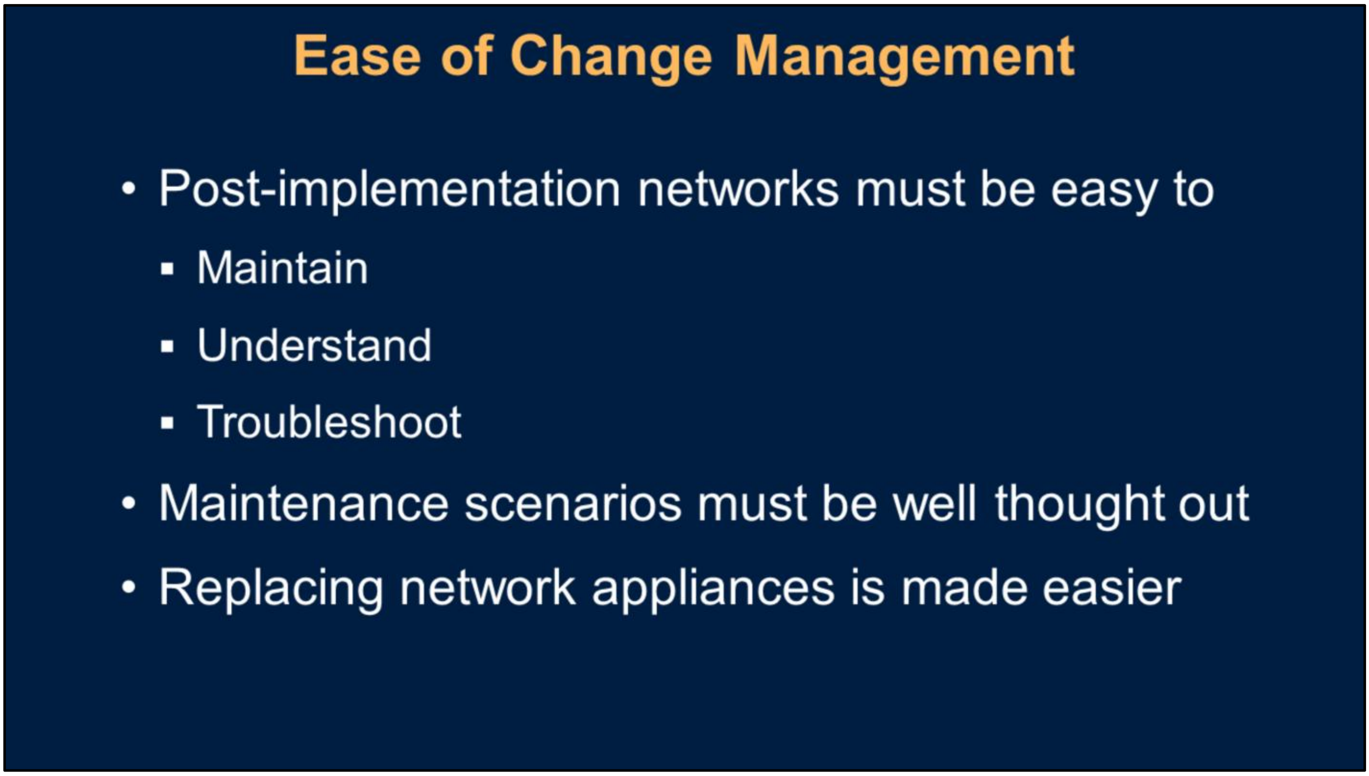
Intrusion detection is typically a bolt on a network device that requires read access to all traffic in the network in order to analyze the traffic for suspicious, unwanted, unauthorized, or malicious frames. The traffic-engineered SDN whitelist approach inherently provides many intrusion detection system (IDS) capabilities, and an SDN controller can serve as a deeper packet analyzer to determine that the payload integrity of the machine-to-machine communications is intact.



Inherent Network Access Control

- Log of access attempts by controller
- Configuration baseline and configuration change management
- Firewall at every hop

Network access control (NAC) includes application-level control and information. Firewall capabilities are no longer just at the perimeter of the network; they are at every hop along the path for data exchange and are no longer simply focused on the TCP/IP rules. Firewall rules include up to the first 128 bytes of the frame header. NAC baselining includes the flow, communications application, and protocol and not just the open and unused ports of each end device.



Ease of Change Management

- Post-implementation networks must be easy to
  - Maintain
  - Understand
  - Troubleshoot
- Maintenance scenarios must be well thought out
- Replacing network appliances is made easier

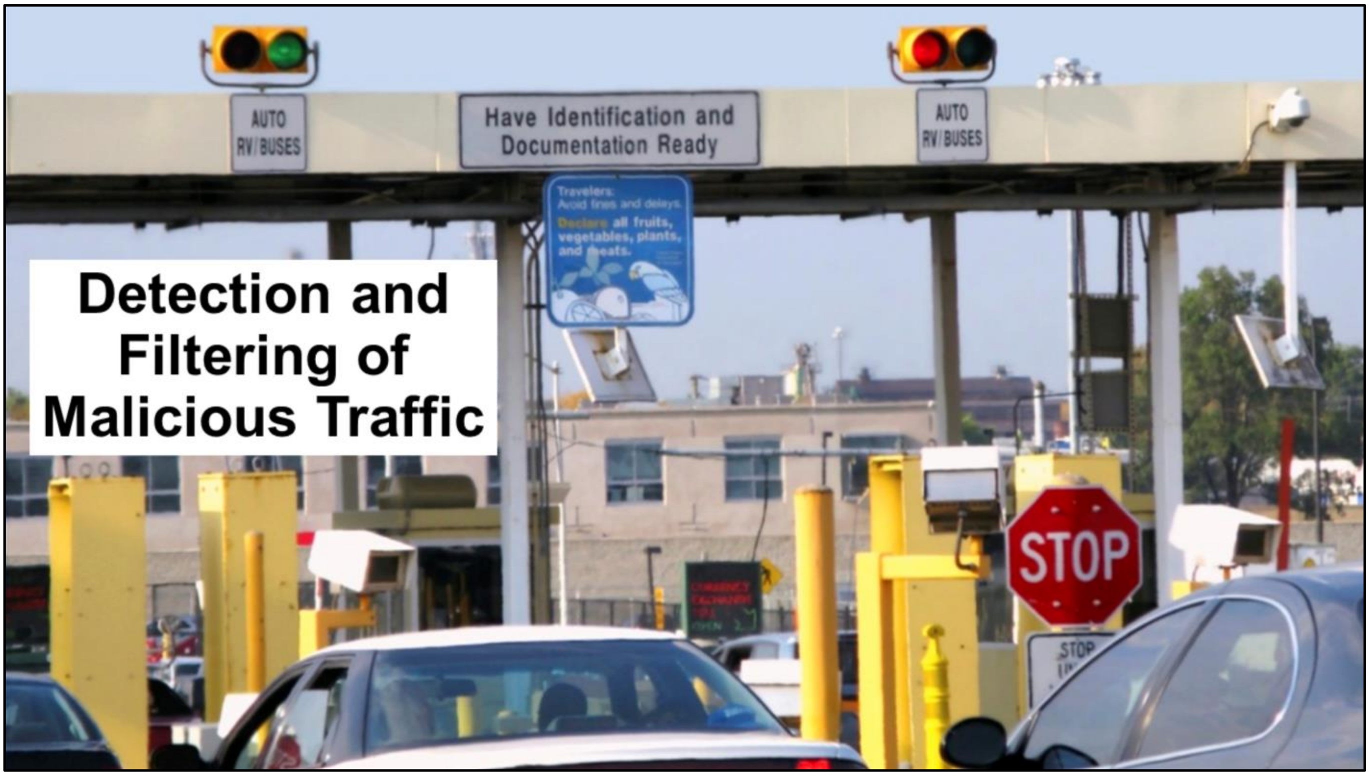
Topology independence comes from the ability to engineer every forwarding hop. The asset owner does not have to design a network to optimize a general healing algorithm. In fact, the network could be any shape, and no port should be unused.

Network simplicity is achieved by thinking at the application level of what applications need to do, and the controller makes sure the path for the communications is provisioned.

Faster failover performance is realized by telling the network appliance what to do before the failure condition is experienced. Then the appliance does not have to ask anyone what to do; it just knows the right thing to do on the very next packet.

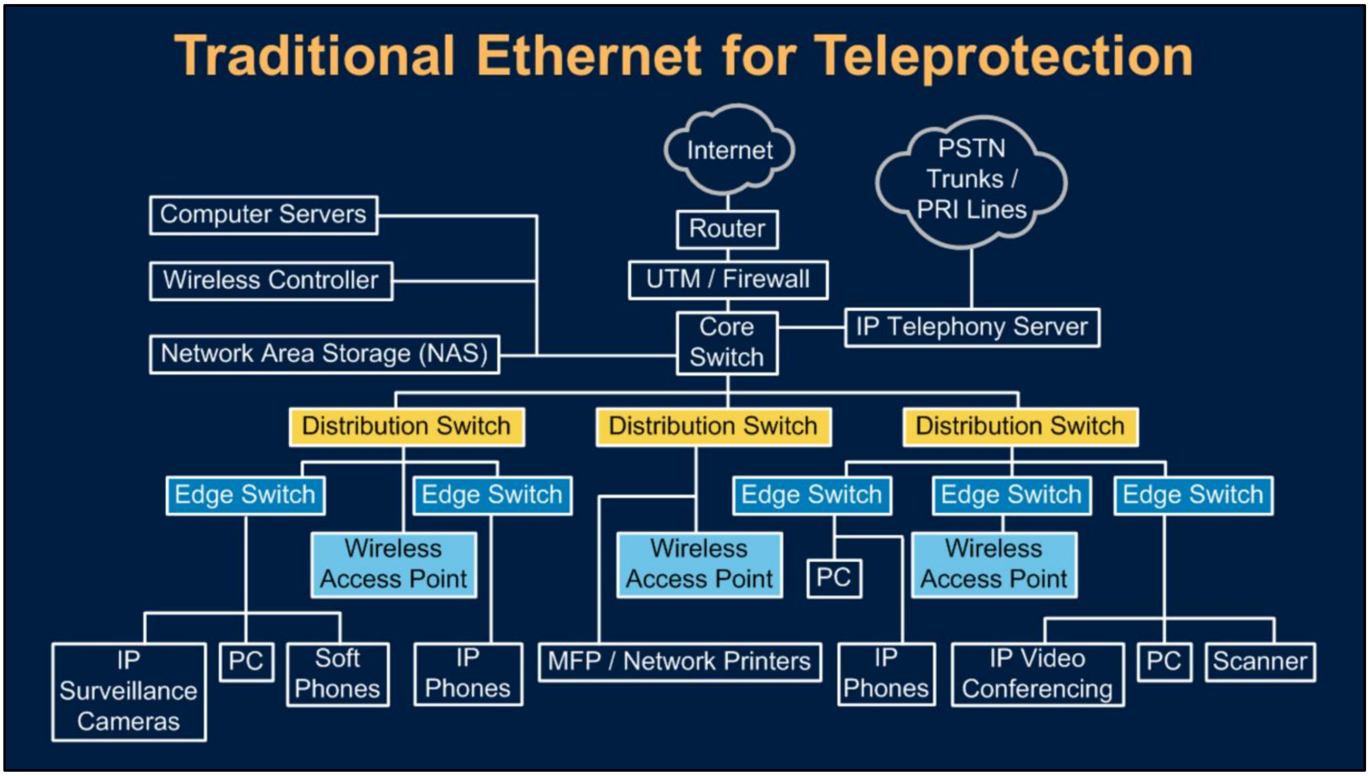
Efficiency increases by eliminating blocked ports for loop mitigation. Loops are controlled by flow paths, and with the ability to use all ports and not tradeoff redundancy, maximum throughput traffic increases.

Central control and monitoring of the network as a single asset improves reliability and safety.



AUTO
RV/BUSES
Have Identification and Documentation Ready
AUTO
RV/BUSES
Travelers: Avoid fines and delays. Declare all fruits, vegetables, plants, and meats.
STOP

Detection and Filtering of Malicious Traffic



Traditional Ethernet for Teleprotection

Internet

PSTN Trunks / PRI Lines

Router

UTM / Firewall

Computer Servers

Wireless Controller

Network Area Storage (NAS)

Core Switch

IP Telephony Server

Distribution Switch

Distribution Switch

Distribution Switch

Edge Switch

Edge Switch

Wireless Access Point

Wireless Access Point

Edge Switch

Edge Switch

Edge Switch

PC

Wireless Access Point

IP Surveillance Cameras

PC

Soft Phones

IP Phones

MFP / Network Printers

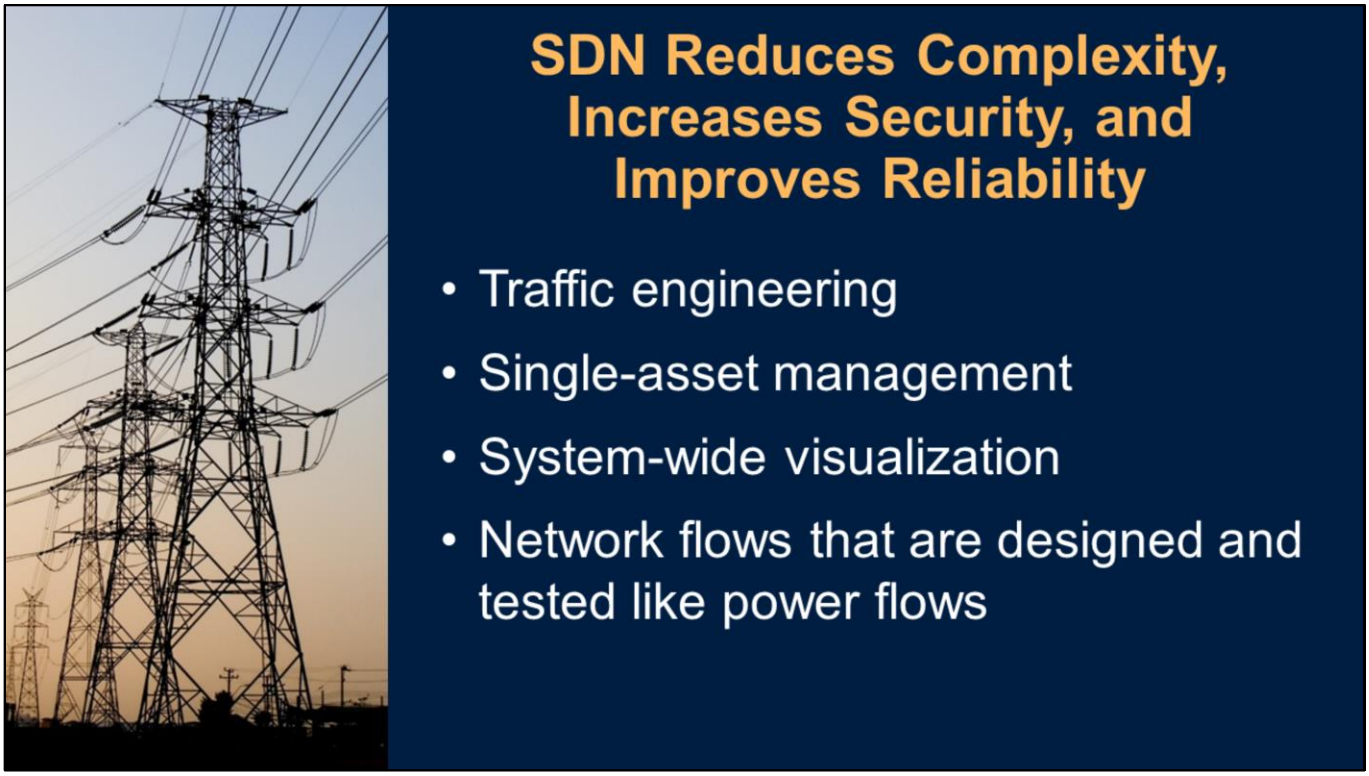
IP Phones

IP Video Conferencing

PC

Scanner

Today’s traditional Ethernet is complex, expensive, and incomplete (with a lack of embedded cybersecurity).



SDN Reduces Complexity, Increases Security, and Improves Reliability

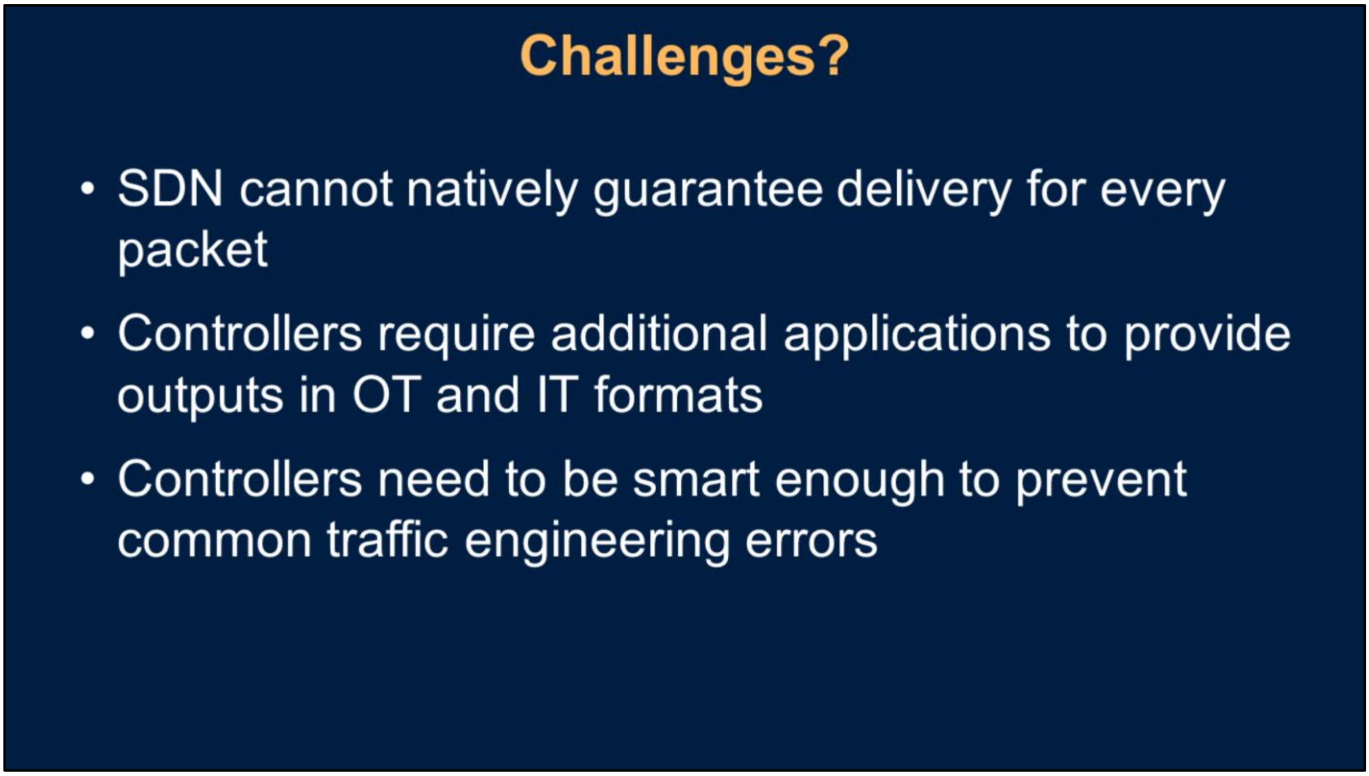
- Traffic engineering
- Single-asset management
- System-wide visualization
- Network flows that are designed and tested like power flows

SDN applies traffic engineering to Ethernet networking with centralized change management and system-wide visualization, while allowing configuration, testing, and maintenance of the network to be done with a service-oriented (instead of packet-oriented) mentality. SDN also blends the worlds of engineering power lines or pipelines with networks. Engineers can apply the same principles of design and validation to the flow of electric power, oil, or communications. Moving electrons or packets from Point A to Point B becomes an engineering solution that can be designed and tested to N-1 or N-2 conditions and can have performance metrics measured before being applied to the live system. Note that N-1 and N-2 redundancies are forms of resilience that ensure system availability in the event of component failure. All components (N) have at least one or two independent backup components or failover paths.

With traditional Ethernet, network operators typically get only abstract information on network performance and events, such as RSTP convergence events or link bounces. With SDN, it is easier for network managers and system operators to monitor and troubleshoot events because they receive information on flow statistics that directly relate to the application and service, enabling root cause to be determined more intuitively.

SDN also offers the following benefits:

- Topology-independent performance.
- Network simplicity.
- Faster failover.
- Application-focused configuration.
- Abstraction of control and security.
- Maximized efficiency and throughput.
- Centralized management and monitoring.



Challenges?

- SDN cannot natively guarantee delivery for every packet
- Controllers require additional applications to provide outputs in OT and IT formats
- Controllers need to be smart enough to prevent common traffic engineering errors

OpenFlow-enabled failover logic cannot guarantee the delivery of every packet in an Ethernet network because packets traversing physical links or egressing physical ports could still be lost. Furthermore, controllers cannot output information to other network management systems (NMSs) naturally. Instead, applications (or plugins) for OpenFlow controllers need to be built that allow tie-ins with existing operational and informational visualization and alerting systems. Controllers also need to be smart enough to prevent scenarios (such as network loops) that specialized protocols guard against today.

Questions?