Documentation Facts

Good documentation ensures that users can find the information they need when making decisions or troubleshooting problems on your network. Documentation details what must be done and how it is to be accomplished. Good documentation keeps track of what has happened and what actions were taken, and helps ensure that actions are consistent within your organization.

The following table lists various types of documents that you might be responsible for creating or maintaining on your network.

Document Type	Description
Policy	A *policy* is a document that describes the overall goals and requirements for a network. A policy identifies what should be done, but may not necessarily define how the goal is to be reached. Depending on your network, you might define policies for different areas of implementation such as policies for: • Administrative delegation • Network documentation • Security
Regulation	A *regulation* is a requirement published by a government or other licensing body that must be followed. While you are not responsible for writing regulations, you are responsible for knowing which regulations apply to your organization, and making sure that those regulations are understood and adhered to. Policies are often written in response to regulations.
Procedure	A *procedure* is a step-by-step process outlining how to implement a specific action. The design of a procedure is guided by goals defined in a policy, but go beyond the policy by identifying specific steps that are to be implemented. The use of consistent procedures ensures that the goals defined in a policy are met, and provides consistency of action by multiple administrators.
Network diagram	A *network diagram* shows the logical and/or physical layout of your network. The network diagram could be a collection of diagrams showing the following information: • The location and IP addresses of hubs, switches, routers, and firewalls. • The relationship of remote locations and the WAN links that connect remote locations. • Subnets within your network, including the subnet addresses and routers connecting each subnet.
Wiring schematic	A *wiring schematic* is a type of network diagram that focuses on the physical connections between devices. The wiring diagram typically shows: • The location of drop cables and ports within offices or cubicles. • The path that wires take between wiring closets and offices. • A labeling scheme that matches endpoints in offices and cubicles with specific

	switch ports or punchdown block locations.
Configuration	*Configuration* documentation identifies specific configuration information for a device. For example, a configuration document for a firewall might include information about the IP addresses assigned to each interface and opened firewall ports. Configuration documentation has two goals: • Document the configuration so that the device can be restored to the original configuration. • Document the configuration so that the current configuration can be compared to the desired configuration to identify any changes.
Change/job logs	*Change* or *history* documentation keeps track of changes to the configuration of a device or the network. Change documentation is often included as a part of the configuration documentation. For example, you might record a change in a network interface card in a device, or a repair to a WAN link. Change documentation is useful for troubleshooting to identify what has been done to the device, and keeps track of changes in the configuration as well as the rationale behind those changes.
Baseline	A *baseline* is a snapshot of the performance statistics of the network or devices. The baseline is used as a logical basis for future comparison. Baselines enable you to effectively monitor the performance of your system to determine when changes negatively impact performance or when systems need upgrading or replacing. It is important to measure network performance at subsequent intervals to see how your server is performing compared to the baseline.

SNMP Facts

Simple Network Management Protocol (SNMP) is a protocol designed for managing complex networks. SNMP lets network hosts exchange configuration and status information. This information can be gathered by management software and used to monitor and manage the network.

SNMP uses the following components.

Component	Description
Manager	A *manager* is the computer used to perform management tasks. The manager queries agents and gathers responses.
Agent	An *agent* is a software process that runs on managed network devices. The agent communicates information with the manager and can send dynamic messages to the manager.
Management Information Base (MIB)	The MIB is a database of host configuration information. Agents report data to the MIB, and the manager can then view information by requesting data from the MIB.
Trap	A *trap* is an event configured on an agent. When the event occurs, the agent logs

	details regarding the event.

Agents and the manager are configured to communicate with each other using the community name. The community name identifies a group of devices under the same administrative control. The community name is *not* a password, but rather it is simply a value configured on each device. Devices with different community names are unable to send SNMP messages to each other.

Remote Management Facts

There are typically two types of solutions for providing remote management of network devices:

Method	Description
Terminal emulation	A *terminal* is a monitor and keyboard attached to a device (i.e. mainframe, server, or router), typically through a serial or special console port. The terminal displays a text-based interface, and users interact with the device by typing commands. A *terminal emulation* utility is a program that allows a console connection through the network. The terminal emulation software communicates with the device over the network, and displays the text-based console screen. There are two common terminal emulation programs used: • Telnet opens a plain-text, unsecured connection. Telnet uses TCP port 23. • Secure Shell (SSH) provides the same capabilities as Telnet, but encrypts data. SSH uses TCP port 22.

Remote desktop	Instead of showing a simple command-line interface, a remote desktop utility displays the graphical user interface of a remote device. Remote desktop solutions are used to remotely manage a computer, or to allow support personnel to view and troubleshoot a remote user's system. Remote desktop software typically has the following three components:
	
	• The server software runs on the target desktop.
	• The client (or viewer) software runs on a remote system. When you run the client software, you see the desktop of the server system.
	• The remote desktop protocol is responsible for the communication between the server and the client.
	  ○ The graphical desktop on the server is sent to the client.
	  ○ Keystrokes and mouse movements on the client are sent to the server.
	  ○ The server executes the actions performed on the client, which modifies data on the server and results in changes to the desktop.
	  ○ The desktop changes are transferred and displayed on the client.
	
	The protocol is optimized to minimize the amount of traffic generated by this exchange.
	
	There are multiple protocols that can be used for remote desktop connections.
	
	• Virtual Network Computing (VNC) was originally developed for UNIX. Applications using VNC include RealVNC, TightVNC, UltraVNC, and Vine Server.
	• Independent Computing Architecture (ICA) is the protocol used by Citrix products (WinFrame and MetaFrame/XenApp).
	• The Remote Desktop Protocol (RDP) is the protocol developed by Microsoft and used in Microsoft's Terminal Services, Remote Desktop, and Remote Assistance

	solutions. Aqua Connect has licensed RDP and created a version for Mac OS X as a server. Most remote desktop protocols support the following features: • Client software for a variety of operating systems. • Server software for a limited number of operating systems. • The ability to show a remote desktop in a browser without installing client software. • Redirecting printing, sound, or storage from the server to devices connected to the client.

In addition to these solutions, most operating systems or network services provide management tools that are capable of contacting a system remotely.

Network Monitoring Facts

The goal of monitoring is to keep track of conditions on the network, identify situations that might signal potential problems, identify the source of problems, and identify areas of your network that might need upgraded or changed. The following table lists some tools you can use to check the health of your network.

Tool	Description
Logs	*Logs* contain a record of events that have happened on a system. Logging capabilities are built into operating systems, services, and applications. Log entries are generated in response to configuration changes, changes in system state, or in response to network conditions. • By default, some logging is enabled and performed automatically. To gather additional information, you can usually enable more extensive logging. • Many systems have logs for different purposes: a system log for operating system entries, a security log for security-related entries, or an application log for events related to services and processes. • Logging requires system resources (processor, memory, and disk). You should only enable additional logging based on information you want to gather, and should disable logging after you obtain the information you need. • Logs must be analyzed to be useful; only by looking at the logs will you be able to discover problems. Depending on the log type, additional tools might be available to analyze logs for patterns.
Load tester	A *load tester* simulates a load on a server or service. For example, the load tester might simulate a large number of client connections to a Web site, test file downloads for an FTP site, or simulate large volumes of e-mail. Use a load tester to make sure that a system has sufficient capacity for expected loads, and even to estimate a failure point where the load is more than the system can handle.
Throughput tester	A *throughput tester* measures the amount of data that can be transferred through a network or processed by a device (such as the amount of data that can be retrieved from disk in a specific period of time). On a network, a throughput tester sends a specific amount of data

	through the network and measures the time it takes to transfer that data, arriving at a measure of the actual bandwidth. Use a throughput tester to validate the bandwidth on your network, and to identify when the bandwidth is significantly below what it should be. **Note**: A throughput tester can help you identify when a network is slow, but does not give you sufficient information to identify why it is slow.
Packet sniffer	A *packet sniffer* is special software that captures (records) frames that are transmitted on the network. Use a packet sniffer to: • Identify the types of traffic on a network. • View the exchange of packets between communicating devices. For example, you can capture frames related to DNS and view the exact exchange of packets for a specific name resolution request. • Analyze packets sent to and from a specific device. • View packet contents. You typically run a packet sniffer on one device with the intent of capturing frames for all other devices on a subnet. Using a packet sniffer in this way requires the following configuration changes: • By default, a NIC will only accept frames addressed to that NIC. To enable the packet sniffer to capture frames sent to other devices, configure the NIC in *promiscuous mode* (sometimes called *p-mode*). In p-mode, the NIC will process every frame it sees. • When using a switch, the switch will only forward packets to the switch port that holds a destination device. This means that when your packet sniffer is connected to a switch port, it will not see traffic sent to other switch ports. To configure the switch to send all frames to the packet sniffing device, configure *port mirroring* on the switch. With port mirroring, all frames sent to all other switch ports will be forwarded on the mirrored port. **Note**: If the packet sniffer is connected to a hub, it will already see all frames sent to any device on the hub.

Optimization Facts

Network optimization has two main goals:

- Provide redundancy of services or devices so that network access can continue in the event of a failure of one or more components. Redundancy to provide access is often called *fault tolerance*. Ensuring that a network or a service is accessible most of the time is called *high availability*.
- Improve the response and performance of network services or devices.

The following table lists various solutions for providing accessibility and improving performance.

Solution	Description
Ethernet bonding	With Ethernet *bonding* (also called NIC *teaming*), two or more physical connections to the same network are logically grouped (or bonded). Data is divided and sent on multiple interfaces, effectively increasing the speed at which the device can send and receive on the network.

	• On an Ethernet network, a device must have multiple network interface cards connected to different switch ports. • The host operating system must be configured to bond the network adapters into a single entity. • The switch ports must be bonded together to recognize both ports as a valid destination for the same device. Bonding primarily provides increased performance, although some fault tolerance is provided if one NIC goes down. Similar solutions allow you to bond multiple dial-up connections or ISDN channels together.
Spanning tree	*Spanning tree* is a protocol on a switch that allows the switch to maintain multiple paths between switches within a subnet. The spanning tree protocol runs on each switch and is used to select a single path between any two switches. • Without the spanning tree protocol, switches that are connected together with multiple links would form a switching loop, where frames are passed back and forth continuously. • Spanning tree provides only a single active path between switches. Switch ports that are part of that path are placed in a forwarding state. • Switch ports that are part of redundant but unused paths are placed in a blocking (non-forwarding) state. • When an active path goes down, the spanning tree protocol automatically recovers and activates the backup ports necessary to provide continued connection between devices. Spanning tree provides fault tolerance in case a switch port or network segment is broken, but does not provide increased performance (because only one path is active at a time).
Load balancing	*Load balancing* configures a group of servers in a logical group (called a *server farm*). Incoming requests to the group are distributed to individual members within the group. Incoming requests can be distributed evenly between group members, or unevenly based on additional criteria such as server capacity. The primary goal of load balancing is to improve performance by configuring multiple devices to respond as one. Load balancing also provides fault tolerance if the load balancing mechanism is able to detect when a specific farm member is unavailable, automatically distributing new requests to the available members.
Caching engine	*Caching* is the process of saving previously-acquired data for quick retrieval at a later time. With caching, data is stored in memory or on disk within a network device, where it can quickly be retrieved when needed. Recalling the data from the cache is faster than requesting the data from the original location. A common application of a caching engine on a network is a proxy server configured to cache Web content. The proxy server is placed close to the users, typically within the same local area network. As users visit Web sites, Web content is retrieved from the Web servers on the Internet and cached on the proxy server. Subsequent requests for the same Web site are sent by the proxy server from cache rather than retrieved from the Internet. Caching engines are implemented primary to improve performance. They offer some degree of fault tolerance, allowing access to cached content even if the source device is

	offline. Caching can lead to some content being out of date if it has changed on the source but has not yet been refreshed in cache.

Quality of Service (QoS)	Quality of Service (QoS) refers to a set of mechanisms that tries to guarantee timely delivery or minimal delay of important or time-sensitive communications. QoS is particular important when implementing Voice over IP (VoIP), Video over IP, or online gaming where delay or data loss make the overall experience unacceptable.
	
	• QoS requires that traffic from different data streams is labeled. Labels are used to prioritize traffic when bandwidth is limited. Multiprotocol Label Switching (MPLS) is one mechanism that can add labels to network traffic.
	• In addition to delay, QoS mechanisms seek to limit the effects of packets arriving out of order, corrupt packets, or lost or dropped packets.
	• Giving higher priority to some traffic implies that other less-important traffic could be delayed. It is assumed that while the delay might make the end user wait, the delay would not make the resulting data unusable.
	• QoS might include a guaranteed level of service. Common service levels include:
	  ○ *Constant* or reserved means that a certain level of service is guaranteed always available. This level is typically only possible by reserving service, even when no data is being sent.
	  ○ *Variable* service guarantees a certain capacity, but service might vary depending on conditions. This level of service is typically sufficient for voice or video.
	  ○ *Available* guarantees a minimum level of service; additional capacity could be used if it is available, but only the minimum is guaranteed.
	  ○ *Unspecified* service provides whatever service is available with little or no guarantees. This level of service should only be used for data that can tolerate long delays.

Traffic shaper	A *traffic shaper* (also called a *bandwidth shaper*) is a device that is capable of modifying the flow of data through a network in response to network traffic conditions. Specific applications for a traffic shaper include: • A device used with QoS guarantees to ensure timely delivery of time-sensitive data streams. • *Bandwidth throttling* to restrict the amount of data sent within a specific time period, such as to limit the amount of data that can be downloaded from a Web site in an hour. • *Rate limiting* to restrict the maximum bandwidth available to a customer (used by an ISP or a WAN provider).
Multilayer switch/content switch	Normal switching occurs at the OSI model layer 2 using the MAC address to perform frame forwarding. Switches use specialized hardware called an application-specific integrated circuit (ASIC) which performs switching functions in hardware rather than using the CPU and software. Because of this specialized hardware, switches can perform the switching function at "wire speed," meaning that frames are switched without a delay that would be introduced if the CPU and software were required to process the frame. A *multilayer switch* operates at other OSI model layers and can use other information within a packet for making forwarding decisions. For example, a layer 3 switch uses the IP address for making forwarding decisions. Because it is a switch, the ASIC hardware performs the switching functions faster than a comparable router which uses the CPU

	and software. Layer 4-7 switches, also called *content switches*, *web switches*, or *application switches*, are typically used for load balancing. • The switch distributes packets between multiple servers. • Some switches can transform packets at wire speed, for example by performing NAT or adding/removing encryption with SSL or digital certificates.

Network Segmentation Facts

Network segmentation is the process of dividing the network to overcome problems and increase network performance, maximize bandwidth, and reduce congestion. As you segment the network, you will need to consider the collision and broadcast domains on the network.

- A *collision domain* is any network or subnetwork where devices share the same transmission medium and where packets can collide. Collisions naturally increase as the number of devices in a collision domain increase.
- A *broadcast domain* is any network or subnetwork where computers can receive frame-level broadcasts from their neighbors. As you add devices to a network segment, the amount of broadcast traffic on a segment also increases. **Note:** A special condition called a *broadcast storm* happens when broadcast traffic is sent, regenerated, and responded to. In this condition, the amount of broadcast traffic consumes network bandwidth and prevents normal communications. Faulty devices or improper configuration conditions can lead to a broadcast storm.

Segmentation may increase the number of both the collision and broadcast domains. Membership within collision or broadcast domains differs depending on the connection device used.

Device	Collision Domain	Broadcast Domain
Hub	All devices connected to the hub are in the same collision domain.	All devices are in the same broadcast domain.
Bridge or Switch	All devices connected to a single port are in the same collision domain (each port is its own collision domain).	All devices connected to the bridge or the switch are in the same broadcast domain.
Router	All devices connected to a single interface are in the same collision domain.	All devices accessible through an interface (network) are in the same broadcast domain. Each interface represents its own broadcast domain if the router is configured to not forward broadcast packets.

In considering a network expansion solution, it is important to identify the connectivity problems you need to resolve, and then identify the device that is best suited for that situation. The main differences between routers, switches, and bridges are the range of services each performs and the OSI layer at which they operate.

Device	Characteristics
Router	Choose a router if you need to:

	• Filter broadcast traffic to prevent broadcast storms • Reduce the number of devices within a broadcast domain (effectively increasing the number of broadcast domains) • Enforce network security
Switch	Choose a switch if you need to: • Provide guaranteed bandwidth between devices • Reduce collisions by decreasing the number of devices in a collision domain (effectively creating multiple collision domains) • Reduce the number of devices within a broadcast domain (creating multiple broadcast domains on a switch is done by using virtual LANs (VLANs)) • Implement full-duplex communication
Bridge	Choose a bridge if you need to: • Isolate data traffic to one network segment • Link unlike physical media (e.g. twisted pair and coaxial Ethernet) of the same architecture type **Note:** In most cases where you might use a bridge, choose a switch instead.

In general, follow these guidelines to make decisions about the appropriate connectivity device.

- Use a bridge to segment the network (divide network traffic) and to provide fault tolerance.
- Use a switch to reduce collisions and offer guaranteed bandwidth between devices.
- Use a router or a switch with virtual LANs (VLANs) to filter broadcast messages, implement security, or connect different networks.