# **46. QoS (Voice VLANs) : PART 1**

## **IP Phones / Voice LANs**

* Traditional phones operate over the **Public Switched Telephone Network (PSTN)**.
  + Sometimes, this is called **POTS (Plain Old Telephone System)**.
* **IP Phones** use **VoIP (Voice Over IP)** technologies to enable phone calls over an **IP network**, such as the **Internet**.
* IP Phones are connected to a **switch**, just like any other **end host**.

### **IP Phones**

* Have an **internal 3-port switch**:  
  + **1 port** is the **uplink** to the **external switch**.
  + **1 port** is the **downlink** to the **PC**.
  + **1 port** connects **internally** to the **phone itself**.
* This allows the **PC and IP phone** to share a **single switch port**.
* It is **recommended** to separate **voice traffic (from IP phone)** and **data traffic (from the PC)** by placing them into **separate VLANs**:  
  + This can be accomplished using a **Voice VLAN**.
  + Traffic from the **PC** will be **untagged**, but traffic from the **phone** will be **tagged** with a **VLAN ID**.

## **Power Over Ethernet (PoE)**

* **PoE** allows **Power Sourcing Equipment (PSE)** to provide **power** to **Powered Devices (PDs)** over an **Ethernet cable**.
* Typically, the **PSE** is a **switch**, and the **PDs** are **IP phones, IP cameras, wireless access points, etc.**
* The **PSE** receives **AC power** from the outlet, converts it to **DC power**, and supplies that **DC power** to the **PDs**.

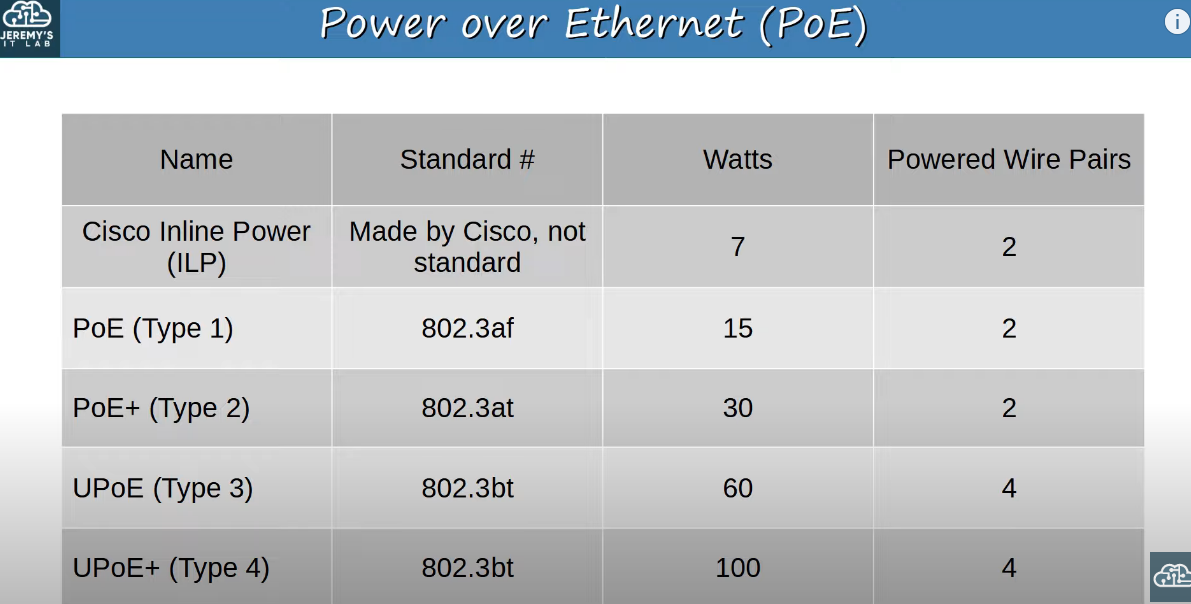
### **PoE Power Management**

* **Too much electrical current** can **damage devices**.
* **PoE** has a process to determine **if a connected device needs power** and **how much** it needs:
  + When a **device** is connected to a **PoE-enabled port**, the **PSE (switch)** sends **low-power signals**, monitors the response, and determines the power requirement.
  + If the **device needs power**, the **PSE supplies power** to allow the **PD to boot**.
  + The **PSE** continues to **monitor the PD** and supply the **required amount of power** (without overloading it).

### **Power Policing**

* **Power policing** can be configured to prevent a **PD** from drawing **too much power**.
* power inline police command configures power policing with the **default settings**:
  + Disables the **port** and sends a **syslog message** if a **PD** draws **too much power**.
  + Equivalent to: power inline police action err-disable.
  + The **interface** enters an **error-disabled** state and can be re-enabled with shutdown followed by no shutdown.
* power inline police action log does **not shut down the interface** if the **PD draws too much power**.
  + It **restarts the interface** and **logs a syslog message**.

Not Tested Good To Know:



## **Introduction to Quality of Service (QoS)**

* **Voice traffic and data traffic** used to use **separate networks**:
  + **Voice traffic** used the **PSTN**.
  + **Data traffic** used an **IP network (Enterprise WAN, Internet, etc.)**.
* **QoS wasn’t necessary** as the **different types of traffic** did not compete for **bandwidth**.
* **Modern networks** are typically **converged networks**, where **IP phones, video traffic, and regular traffic** share the same **IP network**.
  + Enables **cost savings** and **advanced features** (e.g., **Cisco WebEx, MS Teams, etc.**).
  + However, different types of **traffic** must **compete for bandwidth**.
* **QoS** is a set of **tools** used by **network devices** to apply **different treatments** to different packets.

### **QoS Traffic Management**

QoS is used to **manage** the following characteristics of **network traffic**:

* **Bandwidth**
  + Overall **capacity** of the **link** (measured in **bps**).
  + QoS tools allow **reserving bandwidth** for specific kinds of **traffic**.
* **Delay**
  + **One-Way Delay** = Time it takes for traffic to travel from **source to destination**.
  + **Two-Way Delay** = Round trip time (**source → destination → back to source**).
* **Jitter**
  + Variation in **one-way delay** between **packets sent** by the **same application**.
  + **IP phones** have a **jitter buffer** to provide **fixed delay** for **audio packets**.
* **Loss**
  + The **percentage** of **packets** that **do not reach** their **destination**.
  + Causes:
    - **Faulty cables**.
    - **Full packet queues**, causing **packet drops**.

### **QoS Recommended Standards**

* **Acceptable interactive audio quality**:
  + **One-Way Delay**: **≤150ms**.
  + **Jitter**: **≤30ms**.
  + **Loss**: **≤1%**.
* If these **standards** are **not met**, it can cause a **noticeable drop** in **phone call quality**.

## **QoS Queuing**

* If a **network device** receives **messages faster** than it can **forward them**, the messages are placed in a **queue**.
* By **default**, the **queued messages** are **forwarded in FIFO (First In First Out) order**.
  + Messages are **sent** in the **order received**.
* If the **queue is full**, **new packets are dropped** (known as **tail drop**).

### **Tail Drop and TCP Global Synchronization**

* **Tail drop** is **harmful** because it can lead to **TCP global synchronization**.
* When the **queue fills up** and **tail drop** occurs:
  + **All TCP hosts** slow down their **traffic transmission rate**.
  + They **all increase the rate simultaneously**, leading to **congestion and packet drops**.
  + The process **repeats** in **waves**, causing **network instability**.

### **Solution: Random Early Detection (RED)**

* **RED** helps **prevent tail drop** and **TCP global synchronization**.
* When **traffic in the queue** reaches a **certain threshold**, the **device randomly drops packets**.
  + **Only certain TCP flows** are affected, preventing **all TCP hosts** from slowing down at the same time.
* **Weighted Random Early Detection (WRED)**
  + An **improved version** of RED.
  + Allows **controlling** which **packets** are dropped based on **traffic class**.