What is 802.1X?

IEEE 802.1X specifies a general method for the provision of port-based network access control at Layer 2. This method allows you to authenticate a client when it initially connects to a LAN before it gets an IP address (via DHCP) or other higher layer transport configuration. 802.1X does so by defining a standard for passing an authentication protocol (such as EAP) over a wired or wireless 802 LAN by packaging these messages in Ethernet frames.

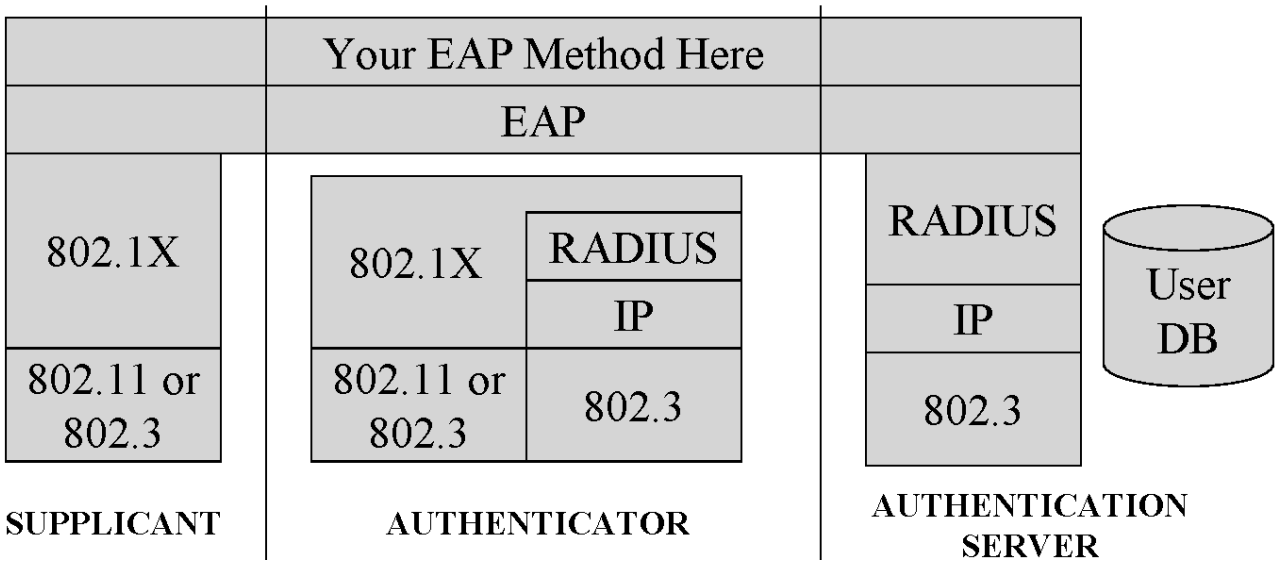
IEEE 802.1X is also called EAPOL (EAP encapsulation over LANs). So to understand what 802.1X does it helps to look at the history of the underlying layer EAP and its precursor PPP. PPP and EAP are Internet standards, defined through the RFC process. IEEE 802.1X is an IEEE standard built on the Internet standard EAP

So how does IEEE 802.1X work ?

IEEE 802.1X uses three terms to describe the three parties involved in an authentication transaction.

- The user or client that wants to be authenticated is a **Supplicant**.
- The actual server doing the authentication is called the **Authentication Server**.
- The device in between these two elements, such as a switch or wireless access point, is called the **Authenticator**.

The Authenticator is necessary because the Supplicant has no knowledge of where to find the Authentication Server (its IP address for example) and no method to communicate with it (since the Supplicant needs to authenticate before it can pass Layer 3 traffic).



Simple 802.1X Authentication

1. The Authenticator sends an "EAP-Request/Identity" packet to the Supplicant as soon as it detects that the link is active (e.g., the supplicant system has associated with the switch).
2. The Supplicant sends an "EAP-Response/Identity" packet with its identity in it to the Authenticator. The Authenticator then re-packages this packet in the RADIUS protocol and passes it to the Authentication (RADIUS) Server.
3. The Authentication Server sends back a challenge to the Authenticator, such as with a token password system. The Authenticator unpacks this from RADIUS and re-packs it into EAPOL and sends it to the Supplicant.
4. The Supplicant responds to the challenge via the Authenticator, which passes the response onto the Authentication Server.
5. If the supplicant provides proper credentials, the Authentication Server responds with a success message, which is then passed on to the Supplicant. The Authenticator now allows access to the LAN, restricted based on attributes that came back from the Authentication Server.

EAPOL (EAP over LAN) has other message types as well. For example, when the Supplicant is finished, it can send an explicit "logoff" notification to the Authenticator. 802.1X also defines a re-authentication timer, which can be used to require the Supplicant to re-authenticate periodically.

802.1X and wireless

Because 802.1X is actually a LAN-based protocol, the introduction of a wireless access point into the mix doesn't affect much of the behavior. The access point plays the role of the Authenticator.

However, exchanging user authentication credentials over a wireless network must be done with great care because traffic interception is much easier. Without the security of a direct physical connection, safeguards must be built into the protocols to prevent attackers from recovering user credentials as they travel over the radio link.

Since 802.1X is based on the Extensible Authentication Protocol (EAP), it offers the choice of several methods to protect authentication exchanges. The two most common are based on the IETF's Transport Layer Security (TLS) standard: Tunneled Transport Layer Security (TTLS), and protected EAP (PEAP).

802.1X on wireless networks makes use of one additional message type. EAPOL-Key messages are used to transport keys from the authenticator to the supplicant. In dynamic WEP, the keys are not acknowledged and sent in only a single message. WPA enhances the keying by using multi-message "handshakes" to ensure that both the supplicant and authenticator are ready to use new keys.

The authenticator does not participate in the authentication transaction, and must learn keys from some other source. In practice, the keys are derived from a source of key material. Typically, the supplicant and authentication server derive key material from the authentication transaction, and the authentication server sends the key material to the authenticator through RADIUS attributes.

Although the details of wireless authentication protocols are different, they share the same basic design in that now the 802.1X authentication becomes a 2 stage process:

In Stage 1, the Supplicant and the Authentication Server build a TLS tunnel.

1. The Authenticator sends an "EAP-Request/Identity" packet to the Supplicant as soon as it detects that the link is active (e.g., the Supplicant has associated with the access point).
2. The Supplicant sends an "EAP-Response/Identity" packet with its identity in it to the Authenticator. The Authenticator then re-packages this packet inside the RADIUS protocol and passes it on to the Authentication (RADIUS) Server.
3. The Authenticator sends an "EAP-Request/EAP-Packet" packet of the desired authentication type. It is typically written as an EAP-Request/Method, such as "EAP-Request/PEAP" or "EAP-Request/TTLS."
4. The Authentication Server and Supplicant then exchange keying material and build a TLS record layer cipher suite with which to secure subsequent EAP communication.

In Stage 2, the Supplicant and Authentication Server use the TLS tunnel to exchange information

1. The Authentication Server sends back an encrypted challenge to the Authenticator, such as with a token password system. The Authenticator unpacks this from RADIUS and re-packs it into EAPOL and sends it to the Supplicant.
2. The Supplicant responds to the challenge via the Authenticator, which passes the response onto the Authentication Server.
3. If the Supplicant provides proper credentials, the Authentication Server responds with a success message, which is then passed on to the Supplicant. The Authenticator now allows access to the LAN, restricted based on attributes that came back from the Authentication Server. LAN access may be restricted to a particular VLAN, time of day, or even with a specified ACL.

All valuable information (username, password) remains encrypted between the Supplicant and Authentication Server protecting both ends from attack and interception. The tunnel, however, still needs a carrier protocol that each end can understand. The Authenticator supports this by passing the encrypted data between the Supplicant and Authentication Server via the appropriate protocol (EAPOL or RADIUS)