antaira®

making connectivity simple...

Software Manual

Version 1.0

(July 2025)

AIROLINX-6 Series

Software Manual
Version 1.0 (July 2025)

The manual supports the following models:

- AIROLINX-6-IOM-IP67-T
- AIROLINX-6-EOM-IP67-T
- AIROLINX-6-ILD-IP67-T

This manual supports the following firmware version:

- v7.19.3

Please check our website (www.antaira.com) for any updated manual or contact us by e-mail (support@antaira.com).

1 Introduction

The Antaira Airolinx 6 and Airolinx 6 Pro are line of Single and Dual band Wi-Fi 6 (802.11ax) product designed for harsh environments such as Outdoor, Warehouse and Industrial Automation applications. This manual provides a structured guide to configuring and troubleshooting your Airolinx 6 product. Each section below corresponds to a menu item in the device's web interface. The content is organized in the order of the interface's menus, allowing you to navigate easily via the left-hand menu panel.

Note:
The Default IP Address of the Airolinx 6 is **192.168.1.1**

There are three default logins for the Airolinx 6:
Username: **root** PW: **admin**
Used for standard setup of wireless including AP, Point-to-Point, Point-to-Multipoint, and Mesh setup.

Username: **advanced** PW: **advancedpw**
Used for advanced features including advanced routing and terminal access.

Username: **expert** PW: **expertpw**
Includes all Airolinx 6 features including IPv6, Wireguard, and advanced profiles.

Important: Some settings and tabs are restricted by User access. If you need a setting or tab described in this manual and it is not available on the root or advanced login, re-login as a higher level user. The user levels in order are root -> advanced -> expert.

Note: When setting up the Airolinx 6 for the first time, log in with each username (root, advanced, and expert) and change the password. Upon first login, you will be prompted automatically to change the password.

Upon logging in to the device's web interface at its default address (typically 192.168.1.1 for new devices), you'll find an initial **Quick Set** wizard for basic setup. Advanced configuration is available through the full menu. The device supports multiple user experience levels: basic setups can be done via Quick Set, while the full menu exposes comprehensive settings for advanced networking. (Note: Some advanced features and menus appear only for users with higher privilege levels.)

The manual covers all major features relevant to the Airolinx 6 hardware platform. Each major section includes an introduction, configuration steps, and troubleshooting tips (where applicable), followed by detailed descriptions of sub-features in the same order as the interface.

2 Quick Set

2.1 Introduction

Quick Set is a simple configuration wizard that prepares your router with common settings in just a few clicks. It is the first screen presented when you access the device's web interface at its default IP. It provides a streamlined, simple way to configure basic AP operation.

- **Home AP:** Standard access point mode for simple AP setup, providing NAT, DHCP, and wireless access (single-band or *HomeAP dual* for dual-band) (Airolinx 6 Pro).

2.2 Configuration Steps (Simple AP Mode)

1. **Access Quick Set:** Connect your PC to the router (via Ethernet) and browse to http://192.168.1.1. Log in with the default credentials (printed on the device label).
2. **Configure Network:** Fill in the basic settings shown. In HomeAP mode, you will set:
    - Internet (WAN) type – e.g., DHCP client, Static, or PPPoE for your ISP.
    - Local Network – SSID (network name) and password for the Wi-Fi 6 network.
    - Wireless – SSID (network name) and password for the Wi-Fi 6 network.
    - Administrator Password – Set a new password for router login (to replace the default).
3. **Apply Configuration:** Click "Apply" (or "OK/Save") to commit changes. The router will apply the settings; you may temporarily lose connection if its IP or your Wi-Fi settings changed. Reconnect to the new IP/SSID if needed.

2.3 Troubleshooting

Quick Set page not loading: Ensure your computer is in the same subnet (e.g. 192.168.1.x). Check that any firewall on your PC isn't blocking access.

Cannot log in: Double-check the default username (root, advanced or expert)If you changed the password and lost it, contact Antaira Technical Services at support@antaira.com.

Wi-Fi client cannot connect: Ensure you set the Wi-Fi password (WPA2/WPA3 passphrase) correctly and that the client supports the security mode.

Quick Set is meant for convenience. For any settings not present in Quick Set (for example, more advanced firewall rules or detailed wireless tweaks), you should switch to the main configuration interface by clicking "Advanced". Once in the main interface, all features described in the subsequent sections of this manual will be available for configuration.

3 WiFi

3.1 Introduction

The **WiFi** menu is dedicated to managing Wi-Fi interfaces and settings on the Airolinx 6. On Antaira Airolinx 6 devices, this menu configures the modern 802.11ax wireless interface, supporting Wi-Fi 6 features such as WPA3 security, OFDMA, MU-MIMO, and improved throughput.

The WiFi configuration is profile-based and highly flexible. Key components include:

- **Wi-Fi Interfaces:** Physical wireless radios (e.g., wifi1, wifi2) corresponding to 2.4 GHz and 5 GHz radios. Each interface has a Configuration that references various profiles.
- **Security Profiles:** Define authentication and encryption (WPA2, WPA3, OWE, etc.) for use by Wi-Fi interfaces.
- **AAA (Authentication, Authorization, Accounting):** Settings for external authentication (like 802.1X, RADIUS) for enterprise networks.
- **CAPsMAN (Controlled AP) Settings:** The device can act as a CAP (controlled AP) or CAPsMAN manager. CAPsMAN profiles (Datapath, Security, etc.) such as "AAA", "Datapath", "Interworking" appear in this menu for managing multiple APs centrally.
- **Channel and Datapath Profiles:** Channel profiles define radio frequency, channel width, etc., and Datapath profiles define bridging/routing of client traffic (used in CAPsMAN scenarios).
- **Client Steering and Mesh:** Features like "Steering" help with band steering or client load balancing between radios. Mesh setups for Wi-Fi 6 use the Home Mesh mode (via CAPsMAN) rather than legacy 802.11s.

Note:
The Airolinx 6 supports the use of *profiles* to simplify and organize wireless configuration. While profiles are optional—users can configure all settings directly within the WiFi interface—they offer a reusable way to apply settings across multiple interfaces.

When a profile is assigned to a WiFi interface (e.g., wifi1), the values from that profile will apply unless they are specifically overridden within the interface settings. In other words, settings configured directly on the interface take precedence over those inherited from the assigned profile.

Example:
1. A Configuration profile named cfg1 is created with an SSID value of TestSSID.
2. The wifi1 interface is set to use cfg1 under its Configuration field. The SSID will be TestSSID.
3. If the SSID is then manually set directly within the wifi1 interface, that SSID will override the one in cfg1.
4. This override behavior applies to all profile fields.

The supported profile types include:
- **Configuration**
- **Channel**
- **Security**
- **AAA**
- **Datapath**
- **Interworking**

- **Steering**

Important Login Notice:
Profiles are only available when logged in as advanced or expert.
The root login does not provide access to profile management. All configuration must be done directly within each interface when using the root account.

3.2 Configuration Steps

Basic AP Setup (Stand-alone):

1. Go to **Wireless > WiFi Interfaces**. You should see at least one interface (e.g., wifi1). Click on the interface to edit it (or click "Add" if it's not present).
2. In the interface settings, set the Mode to "AP" (Access Point) for creating a Wi-Fi network. Ensure **disabled** is set to "no" (to enable the radio).
3. Choose a country/regulatory domain to enforce proper transmit power and channels (under a "Configuration" or "Channel" section).
4. Set the **SSID** (network name) for your wireless network.
5. Configure security: Under the WiFi Security or Security Profiles section, either edit the default profile or create a new one. For example, set **Authentication Types** to wpa2-psk,wpa3-psk and provide a strong **passphrase** (8-63 characters for WPA2/WPA3). Ensure the WiFi interface is set to use this security profile.
6. Click "OK" or "Apply" to save changes. The Wi-Fi LED should indicate broadcasting, and you can now see the SSID on client devices.

Basic Station (Client) Setup: If you want the device to act as a Wi-Fi client (for example, as a WAN uplink), set the interface **Mode** to "station". Then use the **Scan** feature to find the target SSID and configure the security profile to match the AP's password.

Using CAPsMAN: If you have multiple APs, one device (perhaps this router) can act as a CAPsMAN controller. To do so, configure the CAPsMAN Manager under WiFi settings (enable manager, set IP ranges, etc.), and on each AP (including this one if it also serves clients), enable CAP mode pointing to the manager. Profiles like **Provisioning**, **Datapath**, and **Security** under the WiFi menu will be used to automatically configure the CAPs.

3.3 Troubleshooting

Clients cannot see the SSID: Ensure the wireless interface is not disabled and that the frequency selected is allowed in your country. If hidden SSID is enabled, clients won't see it — disable "hide-ssid" if you want it visible.

Cannot connect with WPA3: Some older clients don't support WPA3. Airolinx 6 by default allows mixed WPA2/WPA3 if configured as above, so those clients should fall back to WPA2. If connectivity issues persist, test with WPA2-only to isolate if the issue is related to WPA3 compatibility.

Weak signal or low throughput: Verify the antenna connections (if external) and placement. Use the **Frequency Scan** tool to choose a less congested channel. Ensure you haven't set the transmit power

too low (by default, auto should use regulatory max). The "Registration" table in Wireless will show signal strengths for connected clients — use it to assess signal quality.

Advanced features not taking effect: Some settings (like "Interworking" for Hotspot 2.0 or advanced 802.11k/v roaming) require support on client devices and a properly configured infrastructure (e.g., RADIUS backend). Ensure all parts of the network are configured if using enterprise features.

Wi-Fi 6 brings higher performance but can be complex. This Antaira device leverages Airolinx 6's advanced wireless stack, meaning you have fine control over your wireless environment. Features like **Channel widths (20/40/80 MHz)**, **DFS channels**, and **Mesh** networking (via CAPsMAN Home Mesh) can be configured to tailor the Wi-Fi to your needs.

3.4 General

The **General** section of the WiFi interface configuration contains fundamental settings that define the basic identity, interface behavior, and hardware characteristics of the wireless interface. These settings affect how the device is represented on the network, how it handles traffic at Layer 2, and how ARP and interface modes are applied.

While some fields are read-only and reflect hardware-determined values, others are user-configurable and directly impact device functionality. Understanding this section is essential for both basic and advanced wireless configuration.

- **Name** (read/write): Logical name of the interface. Rename for clarity (e.g., wifi1, wifi_guest). Avoid special characters or spaces.
- **Default Name** (read-only): Hardware-assigned name used as the system reference. Not editable.
- **Type** (read-only): Displays interface type (e.g., wifi). Determined by the driver and hardware platform.
- **Actual MTU** (read-only): Actual Maximum Transmission Unit in use. May be reduced by encapsulations or other restrictions.
- **L2 MTU** (read-only): Layer 2 MTU representing the frame size excluding encapsulations. Determined by hardware/driver.
- **MTU** (read/write): Sets the IP-level MTU.
    - **Valid Range:** Typically 576—1500
    - **Default:** 1500 (or driver-determined default)
    - **Usage Notes:** Lower MTU may be needed for VPN or tunneled traffic. Incorrect MTU can cause fragmentation or dropped packets.
- **L2MTU** (read/write): Manually sets Layer 2 MTU.
    - **Usage Notes:** Typically used for advanced scenarios like MPLS or bridging.
- **MAC Address** (read/write): Hardware MAC address. Can be overridden.
    - **Usage Notes:** Must remain unique. Useful for MAC-based access control or hardware replacement scenarios.
- **ARP** (read/write): ARP behavior control.
    - **Dropdown Options:**
        - enabled — Normal ARP operation
        - proxy-arp — Responds on behalf of other devices
        - reply-only — Only replies to known ARP entries
        - disabled — Disables ARP processing

    - Usage Notes: Use proxy-arp in bridging or NAT scenarios; reply-only for static ARP security.
- **ARP Timeout** (read/write): ARP entry timeout duration.
    - **Default:** Typically 30—60 seconds
    - **Units:** Time format such as 00:01:00
    - **Usage Notes:** Short timeouts for dynamic networks; long timeouts reduce CPU use in stable environments.
- **Mode** (read/write): Defines the wireless role of the interface.
    - **Dropdown Options:**
        - ap — Operates as a wireless Access Point for client devices to connect to.
        - station — Connects as a client to another Access Point.
        - station bridge — Connects to an AP and bridges traffic at Layer 2 (similar to Ethernet bridging).
        - station pseudobridge — Connects to an AP and simulates bridging using NAT (use when station bridge is not supported by the AP).

Note:
The **Mode** setting determines the functional behavior of the wireless interface. It is important to choose the correct mode based on the role of the device in the wireless network:

- **ap**: Use this mode when the Airolinx 6 is providing a wireless network for other devices to connect to. This is typical in Access Point deployments.
- **station**: Use when the Airolinx 6 is connecting to an existing wireless network as a client. Supports connections to any standard Wi-Fi AP.
- **station bridge**: Recommended when the Airolinx 6 must be part of the same broadcast domain as the AP (true Layer 2 bridging). Requires support from the AP side such as another Airolinx 6 unit in AP Mode.
- **station pseudobridge**: Useful for basic client access where bridging is needed but full station bridge mode is not supported by the AP. Uses a workaround to simulate Layer 2.

3.5 Configuration

The **Configuration** section defines the wireless operating parameters for a given WiFi interface or configuration profile. These settings include country regulation, SSID, radio chains, power, distance, and quality-of-service parameters. These settings directly affect how the device broadcasts and manages client associations.

- **Configuration** (read/write): Assigns a pre-defined Configuration Profile to the interface. This enables profile-based deployment and simplifies management across multiple interfaces.

Important: This setting is only available when logged in as advanced or expert. It is not visible under the basic root login.

- **SSID** (read/write): The broadcasted name of the wireless network.
- **Country** (read/write): Specifies the regulatory domain.
    - **Dropdown Options:** Varies by device; includes country names like United States, Canada, etc.

    - **Usage Notes:** The selected country limits available channels and transmit power according to local regulations. Available countries may vary depending on regulatory domain restrictions.
- **Chains** (read/write): Determines which antenna chains are active (used for both Tx and Rx). Typically 0, 1, or both.
- **Tx Chains** (read/write): Specifically defines the chains used for transmission. Matches Chains if not specified.
- **Max Tx Power** (read/write): Sets the maximum transmit power in dBm.
    - **Valid Range:** Based on country and chipset (e.g., 0—30 dBm)
    - **Usage Notes:** Higher power may increase range but could cause interference or violate regulations.
- **Antenna Gain** (read/write): Specifies the antenna gain (in dBi). Used in regulatory calculations to maintain legal EIRP.
- **Distance** (read/write): Sets the expected maximum client distance (in kilometers). Helps tune timing parameters for long-distance links.
- **Hide SSID** (read/write): Whether to suppress SSID broadcast.
    - **UI Control:** A graphical on/off switch. When enabled, the switch moves to the right and turns green.
- **Manager** (read/write): Indicates whether the configuration is managed locally or by a controller.
    - **Dropdown Options:**
        - capsman — Configuration is managed remotely via CAPsMAN (Centralized Access Point Manager).
        - capsman or local — Configuration may fall back to local if CAPsMAN is unavailable.
        - local — Configuration is fully managed locally on the device.
- **Beacon Interval** (read/write): Time between beacon transmissions (in ms). Common default is 100.
- **DTIM Period** (read/write): Delivery Traffic Indication Message interval (in beacons). Affects power-saving clients.
- **Multicast Enhance** (read/write): Enables multicast enhancement to optimize streaming performance.
    - **Dropdown Options:** enabled, disabled
- **QoS Classifier** (read/write): Determines how QoS is handled for traffic prioritization.
    - **Dropdown Options:**
        - dscp high 3 bits — Uses the top 3 bits of the DSCP field in IP packets to classify priority.
        - priority — Uses Layer 2 (802.1p) priority tags to classify traffic.
- **Station Roaming** (read/write): Enables or disables roaming support for wireless clients.
    - **UI Control:** A graphical on/off switch. When enabled, the switch moves to the right and turns green.
    - **Usage Notes:** This assists client devices with transitioning between access points in a multi-AP deployment.
- **Max Clients** (read/write): Sets the maximum number of clients that can connect to this SSID.
    - **Usage Notes:** Helps manage load and prevents over-association in dense deployments.

3.6 Channel

The **Channel** section controls wireless frequency, channel width, and regulatory behavior for the radio. These settings are essential for optimizing performance, complying with country restrictions, and reducing interference in the 5GHz band.

- **Channel** (read/write): Assigns a predefined Channel Profile to the interface. Channel Profiles allow consistent RF behavior to be reused across multiple WiFi interfaces.

  Important: This setting is only available when logged in as advanced or expert. It is not visible under the basic root login.

- **Band** (read/write): Specifies the Wi-Fi technology used on the 5GHz interface.
  - **Dropdown Options:**
    - 5GHz A — Uses legacy 802.11a
    - 5GHz A/N — Enables both 802.11a and 802.11n (Wi-Fi 4)
    - 5GHz AC — Enables 802.11ac (Wi-Fi 5)
    - 5GHz AX — Enables 802.11ax (Wi-Fi 6)
  - **Usage Notes:** Wi-Fi 6 (802.11ax) is fully backward-compatible. Devices using older standards (such as 802.11a/n/ac) can still connect to a Wi-Fi 6 access point, but they will operate using their respective standard and will not benefit from Wi-Fi 6 enhancements.
- **Channel Width** (read/write): Controls the width of the wireless channel and potential throughput.
  - **Dropdown Options:**
    - 20MHz — Basic, stable; good for congested areas. Ideal environments up to 287 Mbps
    - 20/40MHz — Auto between 20 and 40 MHz. Ideal environments up to 574 Mbps
    - 20/40MHz Ce — Center extension below control channel. Mostly used for channel interference management, spectrum coordination, and DFS avoidance.
    - 20/40MHz eC — Center extension above control channel. Mostly used for channel interference management, spectrum coordination, and DFS avoidance.
    - 20/40/80MHz — Used by 802.11ac and 802.11ax. Ideal environments up to 1201 Mbps
    - 20/40/80/160MHz — Enables 160 MHz for max throughput. Ideal environments up to 2402 Mbps
    - 20/40/80+80MHz — Two separate 80 MHz blocks for advanced Wi-Fi 6 clients. Ideal environments up to 2402 Mbps.

    Note:
    Wider channels provide higher bandwidth but are more prone to interference. Use narrower widths in dense environments. Ideal throughputs listed are in RAW Wi-Fi data. Real-world throughput is always lower than the raw (PHY) data rate due to protocol overhead, interference, contention, retransmissions, encryption, etc.

- **Frequency** (read/write): Specifies the center frequency in MHz. Must be within legal and hardware-supported range.
- **Skip DFS Channels** (read/write): Controls whether Dynamic Frequency Selection (DFS) channels are used.
  - **Dropdown Options:**

    - disable — All DFS channels allowed; device performs radar check before use
    - 10min CAC — Device waits 10 minutes for Channel Availability Check before using a DFS channel
    - all — Skips DFS channels completely
- **Reselect Interval** (read/write): Defines how often (in seconds) the system reevaluates the current channel for changes or interference. Mostly used in dynamic or controller-managed environments.

> **Note:**
> DFS channels can delay availability due to required radar checks. Use all to avoid delays. Use 10min CAC only when DFS channels must be used.

3.7 Security

The **Security** section defines the authentication and encryption parameters for securing wireless networks. These settings control how clients authenticate and what type of encryption is applied to their traffic. Security Profiles can be created and reused across multiple WiFi interfaces.

- **Security** (read/write): Assigns a predefined **Security Profile** to the interface.

> **Important:** This setting is only available when logged in as advanced or expert. It is not visible under the basic root login.

- **Authentication Types** (read/write): Specifies the authentication methods accepted by the access point. Multiple options can be selected simultaneously (checkbox format).
    - WPA PSK — WPA with Pre-Shared Key (legacy; not recommended on its own)
    - WPA2 PSK — WPA2 with Pre-Shared Key (most widely supported)
    - WPA3 PSK — WPA3 with Pre-Shared Key (modern and secure, but not supported on older clients)
    - WPA EAP — WPA with Enterprise authentication (RADIUS or external AAA server)
    - WPA2 EAP — WPA2 with Enterprise authentication
    - WPA3 EAP — WPA3 with Enterprise authentication
    - WPA3 EAP 192 — WPA3-Enterprise 192-bit mode (uses Suite-B ciphers; highly secure)
    - OWE — Opportunistic Wireless Encryption (provides encryption without authentication)
- **Encryption** (read/write): Specifies allowed data encryption methods. These are used for encrypting individual wireless data frames (checkbox format).
    - TKIP — Temporal Key Integrity Protocol (legacy; avoid for new deployments)
    - CCMP — AES-based encryption (standard in WPA2)
    - CCMP 256 — Stronger version of CCMP with 256-bit keys
    - GCMP — Galois Counter Mode Protocol (used in WPA3)
    - GCMP 256 — GCMP with 256-bit keys
- **Group Encryption** (read/write): Controls encryption for broadcast/multicast traffic (drop-down).
    - TKIP
    - CCMP
    - CCMP 256
    - GCMP
    - GCMP 256

- **Group Key Update** (read/write): Specifies how often (in seconds) the group key is rotated.
- **Passphrase** (read/write): Defines the password used for PSK-based authentication. Must be between 8 and 63 characters.
- **Disable PMKID** (read/write): Disables the use of Pairwise Master Key Identifier caching.
    - **UI Control:** Graphical on/off switch. When enabled, the switch moves to the right and turns green.
- **Management Protection** (read/write): Enables or requires 802.11w Management Frame Protection (drop-down).
    - disabled
    - allowed
    - required
- **Management Encryption** (read/write): Specifies encryption method for management frames (drop-down).
    - CMAC
    - CMAC 256
    - GMAC
    - GMAC 256
- **Wi-Fi Protected Setup (WPS)** is a simplified method for allowing wireless clients to join a network securely. When enabled, WPS allows a device to connect by using a push-button method rather than manually entering the passphrase.
    - **Enabled** — Allows WPS functionality using the push-button method (PBC).
    - **Disabled** — Disables WPS entirely. Recommended for networks prioritizing security.

  WPS is supported only with WPA2-PSK or WPA3-SAE authentication types. When using enterprise (RADIUS) authentication, WPS should be disabled.
- **DH Groups** (read/write): Specifies which Diffie-Hellman groups are allowed for SAE (checkbox format).
    - 19 — 256-bit curve (NIST P-256): Strong balance of security and performance; widely supported.
    - 20 — 384-bit curve (NIST P-384): Stronger encryption; suitable for high-security environments.
    - 21 — 521-bit curve (NIST P-521): Highest strength; most secure, but adds processing overhead.
- **SAE Anti-Clogging Threshold** (read/write): Defines how many SAE requests per second are allowed before the AP enables anti-clogging tokens.
- **SAE Max Failure Rate** (read/write): Sets the number of failed authentication attempts before triggering rate limiting or blacklisting.
- **SAE PWE** (read/write): Specifies how the SAE Password Element is computed (drop-down).
    - hunting and pecking
    - hash to element
    - both
- **OWE Transition Interface** (read/write): Specifies an interface that broadcasts the OWE open SSID to support legacy clients (drop-down).
    - auto
    - wifi1, wifi2, or any created virtual interface
- **Connect Group** (read/write): Assigns a connection group ID for grouping clients.
- **Connect Priority** (read/write): Sets a numeric value to prioritize this profile when multiple security configurations are available.

3.8 EAP

The **EAP (Extensible Authentication Protocol)** section allows for configuration of advanced, enterprise-grade wireless authentication settings. These settings are commonly used in WPA2/WPA3-Enterprise deployments, often in conjunction with a RADIUS server. The EAP menu appears when WPA EAP methods are enabled in the associated Security Profile.

- **EAP Methods** (read/write): Specifies the EAP method(s) used for authenticating clients. This is a drop-down list, and multiple methods can be added or removed using the + / — buttons.
    - PEAP — Protected EAP encapsulates EAP within a secure TLS tunnel. Commonly used with username/password credentials.
    - TLS — Requires both client and server certificates. Offers strong, mutual authentication and encryption.
    - TTLS — Tunneled TLS, allows for username/password authentication over a TLS tunnel. Often used where only server-side certificates are required.
- **EAP Certificate Mode** (read/write): Controls how the system handles server certificate validation during EAP authentication (drop-down).
    - don't verify certificate — Server certificates are not validated. Least secure; may expose to man-in-the-middle attacks.
    - no certificates — EAP is used without any certificate requirement. Rare; typically for tunneled methods without server certs.
    - verify certificate — Requires the server certificate to be valid and signed by a trusted CA.
    - verify certificate with CRL — Adds revocation list (CRL) checking to ensure certificates are not only valid but not revoked.
- **EAP TLS Certificate** (read/write): Specifies the client certificate used for EAP-TLS authentication (drop-down).
    - Populated after valid TLS certificates are uploaded to the device.
    - Required only for EAP-TLS authentication.
- **EAP Username** (read/write): Username used for EAP-based authentication (e.g., RADIUS or enterprise server logins). Required for PEAP, TTLS, and some TLS deployments.
- **EAP Anonymous Identity** (read/write): Used to mask the real identity during the initial negotiation phase. The actual username is sent encrypted later in the EAP exchange. Useful for privacy in roaming scenarios (e.g., Hotspot 2.0 or Eduroam).
- **EAP Password** (read/write): Password used in conjunction with **EAP Username** for authenticating against a RADIUS or enterprise authentication server.
- **EAP Accounting** (read/write): Enables RADIUS-based accounting of EAP sessions.
    - **UI Control:** Graphical on/off switch. When enabled, the switch moves to the right and turns green.
    - When enabled, the access point will send accounting start/stop messages to the configured RADIUS server, allowing session tracking.

3.9 FT (Fast Transition)

Fast Transition (FT), also known as **802.11r**, is a feature designed to improve the handoff speed between access points (APs) in a wireless network. Without FT, when a client moves between APs, it must reauthenticate from scratch, leading to noticeable delays or connection drops — which is especially problematic for time-sensitive applications such as **video surveillance, remote industrial control, or automation protocols**.

FT allows seamless roaming by securely caching key information ahead of time, enabling faster reassociation.

This section configures how FT operates on the Airolinx 6 device. It is useful in environments with multiple APs where clients may roam — such as **warehouses, transportation hubs, or large outdoor deployments**.

Note:
For FT to be effective, it must also be supported and properly configured on the client devices.

- **FT Enabled** (read/write): Enables or disables Fast Transition support for the interface.
  - **UI Control:** Graphical on/off switch. When enabled, the switch moves to the right and turns green.
  - **Usage:** Enable this to allow seamless client roaming between APs using FT.
- **FT Mobility Domain** (read/write): A 4-character hexadecimal identifier used to group APs that share FT capability.3.9
  - **Format:** Hexadecimal (e.g., 4a3f)
  - **Usage:** All APs that participate in FT roaming must use the same Mobility Domain value.
  - **Tip:** Choose a unique value per roaming group/network deployment.
- **FT Over DS** (read/write): Determines whether FT messages are tunneled through the distribution system (wired network) or sent over the air.
  - **UI Control:** Graphical on/off switch.
  - **When enabled:** FT messages go through the wired network (over DS).
  - **When disabled:** Messages go directly between AP and client over-the-air.
  - **Usage Notes:** Enabling FT over DS can reduce airtime usage and enhance performance when APs are wired together.
- **FT Reassoc. Deadline** (read/write): Specifies the maximum time (in milliseconds) that a client is allowed to complete the reassociation after initiating an FT handoff.
  - **Usage:** Helps control FT timing to avoid stale handoffs or authentication timeouts.
  - **Typical Range:** 100—1000 ms depending on client responsiveness and network conditions.
- **FT NAS Identifier** (read/write): A unique identifier for the Network Access Server (NAS) used in RADIUS-based authentication.
  - **Usage:** Needed when using FT in WPA-Enterprise (EAP) environments with RADIUS. Helps distinguish between APs in RADIUS logs.
  - **Format:** Typically a hostname or alphanumeric string.
- **FT R0 Key Lifetime** (read/write): Defines how long the R0 Key (used during FT handoff) remains valid before it must be regenerated.
  - **Units:** Seconds (e.g., 86400 = 24 hours)
  - **Usage:** Longer lifetimes reduce key regeneration frequency. Shorter durations may be better for high-security environments.
- **FT Preserve VLAN ID** (read/write): Ensures that VLAN tags assigned to clients are retained during the FT roaming process.
  - **UI Control:** Graphical on/off switch.
  - **Usage:** Enable this when using VLAN tagging in enterprise networks to maintain consistent network segmentation as clients roam.

3.10 AAA

The **AAA (Authentication, Authorization, and Accounting)** section allows the configuration of how wireless clients are authenticated and tracked using external servers such as RADIUS. This is a key component in enterprise environments or centralized authentication systems.

AAA profiles are reusable and can be assigned to multiple interfaces.

- **AAA** (read/write): Assigns a predefined AAA profile to the interface.

> **Important:** This setting is only available when logged in as advanced or expert. It is not visible under the basic root login.

- **Username Format** (read/write): Defines the structure of the username sent to the RADIUS server.
  - **Examples:**
    - user@domain.com
    - DOMAIN\user
    - user
    - Useful when integrating with systems requiring specific domain-qualified usernames.
- **Password Format** (read/write): Specifies how the password is formatted or encoded before being sent to the authentication server.
  - **Usage:** Helps align password formatting with what the RADIUS server expects. Often used in legacy or custom environments.
  - **Examples:**
    - plain-text — Default and most common (password is encrypted by EAP tunnel)
    - uppercase — Transforms password to all uppercase before sending
    - lowercase — Transforms password to all lowercase
    - hex — Treats password string as hexadecimal data
  - **Note:** Always confirm required format with the RADIUS administrator to avoid login issues.
- **Called Format** (read/write): Controls how the Called-Station-Id is sent to the RADIUS server.
  - Typically identifies the AP the client is connecting to.
  - **Formats may include:** AP MAC address, SSID, or a combination (e.g., MAC:SSID).
- **Calling Format** (read/write): Controls how the Calling-Station-Id (client identity) is formatted.
  - Usually the MAC address of the client, but can be adapted to match server requirements.
- **MAC Caching** (read/write): Enables caching of authenticated MAC addresses.
  - Reduces RADIUS load and speeds up reconnections for known clients.
- **Interim Update** (read/write): Sends periodic accounting updates to the RADIUS server for active sessions.
  - Used to track usage or enforce time/session policies.
- **NAS Identifier** (read/write): Identifier sent to the RADIUS server to uniquely represent the AP.
  - May be a hostname, label, or custom string.
  - Important in environments with multiple APs sharing a central authentication system.

3.11 Datapath

Important: The Datapath section is only available when logged in as advanced or expert. It is not visible under the basic root login.

The **Datapath** section controls how client traffic is bridged or routed after connecting to a wireless interface. These settings are critical when integrating with VLANs, Layer 2 bridging, or centralized management systems like CAPsMAN. A proper datapath configuration ensures traffic is handled securely, efficiently, and in accordance with your network design.

Datapath profiles are reusable across multiple interfaces, particularly in controller-managed networks.

- **Datapath** (read/write): Assigns a predefined **Datapath Profile** to the interface.
- **Bridge** (read/write): Specifies which Layer 2 bridge interface the wireless traffic will be associated with.
    - **Dropdown Options:** By default, the bridge named bridge is available. Any user-created bridges will also appear in this list.
    - **Usage:** When set, all wireless clients associated with this interface will be considered members of the selected bridge, allowing for seamless Layer 2 communication with other bridged ports or interfaces.
- **Bridge Cost** (read/write): Sets the STP (Spanning Tree Protocol) cost for the interface when it participates in a bridge.
    - **Usage:** Helps influence STP topology decisions. Lower values increase path preference.
- **Bridge Horizon** (read/write): Prevents forwarding between interfaces within the same bridge horizon group.
    - **Usage:** Useful for isolating traffic between virtual APs or VLANs even if they belong to the same bridge.
- **Client Isolation** (read/write): Enables client isolation at Layer 2, preventing wireless clients from communicating directly with each other.
    - **UI Control:** Graphical on/off switch. When enabled, the switch moves to the right and turns green.
    - **Usage:** Recommended for public or semi-public Wi-Fi environments where client-to-client communication should be blocked for security reasons.
- **Traffic Processing**
  The **Traffic Processing** setting determines how wireless client traffic is handled before it reaches the bridge or routed interface. It controls whether the system processes wireless frames in hardware or software.
    - **default** - Uses the default processing behavior based on the platform and wireless configuration.
    - **cpu** - Forces all wireless traffic to be processed by the CPU. Useful for debugging or applying advanced software-based features.
    - **driver** - Offloads frame processing to the wireless driver, which may provide better performance in certain configurations.

  This setting is relevant for performance tuning and may affect compatibility with bridging, VLANs, or advanced firewall configurations.
- **VLAN ID** (read/write): Assigns a VLAN ID to the wireless interface. All traffic will be tagged with this VLAN ID when exiting the bridge.

    - **Usage:** Essential for segmenting wireless traffic in multi-VLAN networks. Allows integration into VLAN-aware switches and routers.
- **Interface List** (read/write): Determines which logical interface list this interface is included in.
    - **Dropdown Options:**
        - all — Interface is included in all relevant firewall/routing groups
        - dynamic — Automatically included by the system when active
        - none — Excluded from all lists
        - static — Manually managed inclusion in interface lists
    - **Usage:** Interface lists can be used in firewall filters, NAT rules, and routing decisions to group interfaces logically.

3.12 Interworking

Note:
The settings described in this section are identical to those found in the corresponding profile configuration. Profiles allow you to predefine these settings and apply them to multiple interfaces for consistency. The only additional setting in a profile is the ability to name it, so it can be selected from the wireless interface configuration.

- **Interworking** (read/write): Assigns an Interworking Profile to the interface.

  Important: This setting is only available when logged in as advanced or expert. It is not visible under the basic root login.

- **Network Type** (read/write):
    - **Dropdown Options:**
        - emergency only
        - personal device
        - private
        - private with guest
        - public chargeable
        - public free
        - test
        - wildcard **(Default)**
    - **Usage:** Indicates the intended use of the network for interworking/hotspot compatibility.
- **Internet** (read/write): Indicates whether the Internet is available through this interface.
    - **UI Control:** Graphical on/off switch.
- **ESR** (read/write): Indicates emergency services reachable through the access point.
    - **UI Control:** Graphical on/off switch.
- **UESA** (read/write): Unauthenticated Emergency Service Available.
    - **UI Control:** Graphical on/off switch.
    - no — No unauthenticated emergency services are reachable.
    - yes — Unauthenticated emergency services are reachable.
- **Venue** (read/write): Describes the physical location.
    - **Example Options:**
        - outdoor traffic control

        - assembly stadium
        - industrial factory
        - utility unspecified
    - **Note:** The full list is available in the device UI. Choose the most appropriate venue type.
- **HESSID** Homogenous extended service set identifier (HESSID). Devices that provide access to same external networks are in one homogenous extended service set. This service set can be identified by HESSID that is the same on all access points in this set. 6-byte value of HESSID is represented as MAC address. It should be globally unique, therefore it is advised to use one of the MAC address of access point in the service set.
- **Hotspot 2.0** (read/write): Enables Hotspot 2.0 support.
    - **UI Control:** Graphical on/off switch.
- **DGAF** (read/write): Disable Group Addressed Forwarding.
    - **UI Control:** Graphical on/off switch.
- **Roaming OIS** (read/write): Supports multiple entries using +/- controls.
- **Venue Names** (read/write): Supports multiple entries using +/- controls.
- **Authentication Types** (read/write): Supports multiple entries.
    - This setting is only used when asra is set to "yes".
    - Each value supports a URL or can be an empty string using "".
    - **Options:**
        - dns-redirection:url
        - https-redirection:url
        - online-enrollment:url
        - terms-and-conditions:url (Default)
- **IPv4 Availability** (read/write):
    - **Dropdown Options:** double nated, not available, port restricted, port restricted double nated, port restricted single nated, public, single nated, unknown
- **IPv6 Availability** (read/write):
    - **Dropdown Options:** available, not available, unknown
- **Realms** (read/write): Enter a realm name and select an EAP method.
    - **Dropdown Options:** eap aka, eap sim, eap tls, not specified
- **3GPP Info** (read/write): Supports multiple entries using +/- buttons.
- **3GPP Info Raw** (read/write): Raw hex-encoded version of 3GPP information.
- **Domain Names** (read/write): Supports multiple entries using +/- buttons.
- **Operator Names** (read/write): Supports multiple entries using +/- buttons.
- **WAN Status** (read/write):
    - **Dropdown Options:** down, reserved, test, up
- **WAN Symmetric** (read/write): Indicates whether WAN uplink/downlink are symmetric.
    - **UI Control:** Graphical on/off switch.
- **WAN At Capacity** (read/write): Indicates whether WAN link is fully utilized.
    - **UI Control:** Graphical on/off switch.
- **WAN Downlink** (read/write): Maximum expected downstream capacity in kbps. If the downlink speed is not known, set to 0.
- **WAN Downlink Load** (read/write): The downlink load of the WAN connection measured over wan-measurement-duration. Values from 0 to 255.
    - 0 - unknown;
    - 255 - 100%.

- **WAN Uplink** (read/write): Maximum expected upstream capacity in kbps. If the uplink speed is not known set to 0.
- **WAN Uplink Load** (read/write): The uplink load of the WAN connection measured over wan-measurement-duration. Values from 0 to 255.
  - 0 - unknown;
  - 255 - 100%.
- **WAN Measurement Duration** (read/write): Duration during which wan-downlink-load and wan-uplink-load are measured. Value is a numeric value from 0 to 65535 representing tenths of seconds.
  - 0 - not measured;
  - 10 - 1 second;
  - 65535 - 1 hour 49 minutes or more.
- **Connection Capabilities** (read/write): Supports multiple entries with +/- buttons. This option allows to provide information about the allowed IP protocols and ports. This information can be provided in ANQP response. The first number represents the IP protocol number, the second number represents a port number.
  - closed - set if protocol and port combination is not allowed;
  - open - set if protocol and port combination is allowed;
  - unknown - set if protocol and port combination is either open or closed.

  Example: connection-capabilities=6:80:open,17:5060:closed Setting such a value on an Access Point informs the Wireless client, which is connecting to the Access Point, that HTTP (6 - TCP, 80 - HTTP) is allowed and VoIP (17 - UDP; 5060 - VoIP) is not allowed. This property does not restrict or allow usage of these protocols and ports, it only gives information to station device which is connecting to Access Point.
- **Operational Classes** (read/write): Supports multiple entries with +/- buttons.

3.13 Steering

Note:
The settings described in this section are identical to those found in the corresponding profile configuration. Profiles allow you to predefine these settings and apply them to multiple interfaces for consistency. The only additional setting in a profile is the ability to name it, so it can be selected from the wireless interface configuration.

- **Steering** (read/write): Assigns a Steering Profile to the interface.

  Important: This setting is only available when logged in as advanced or expert. It is not visible under the basic root login.

- **Neighbor Group** (read/write): Defines which Access Points are included in steering decisions.
  - **Control**: Drop-down box with +/- buttons to add or remove groups
  - **Usage**: When sending neighbor reports and BSS transition management requests, an Access Point (AP) will list all other APs within its neighbor group as potential roaming candidates.
  - **Default**: A dynamic neighbor group is created for each set of APs with the identical SSID and authentication settings.

Important: A dynamic neighbor group will not be created if EAP is used; it must be defined manually.

- **RRM** (read/write): Enables 802.11k neighbor reports.
  - Control: Graphical on/off switch
  - Default: yes

Note: 802.11k is a standard that improves Wi-Fi roaming by providing information about APs and their signal strength to help devices choose the best one.

- **WNM** (read/write): Enables sending of solicited 802.11v BSS transition management requests.
  - **Control**: Graphical on/off switch
  - **Default**: yes

Note: 802.11v helps manage client roaming by allowing APs to suggest better APs for clients to connect to, optimizing network usage.

3.14 Status

This read-only Status section displays the current state of the Airolinx 6 device's Wi-Fi interface. Each field below is for information only and cannot be modified:

- **Last Link Down Time**: Shows the date and time when the wireless link was last lost (disconnected).
- **Last Link Up Time**: Shows the date and time when the wireless link was last successfully established (connected).
- **Link Down**: Displays the total number of times the wireless link has gone down (dropped).
- **State**: Indicates the interface's current operating state. Possible values include:
  - running — The interface is active and operating normally.
  - disabled — The interface is currently turned off or inactive.
  - scanning — The interface is scanning for a connection (applies to station, station bridge, or station pseudobridge modes).
  - selecting channel — In AP mode, the device is selecting a channel to operate on.
  - cac — The interface is performing a DFS Channel Availability Check (CAC) before using a radar-restricted channel.
  - authorized — The interface has successfully completed authentication and authorization with the connected peer or access point. This state indicates that the client is fully connected and permitted to send and receive network traffic according to the configured security and access control policies.
- **Current Channel**: Displays the current wireless channel frequency and configuration. For example, a value like 5500/ax/Ceee/D indicates the device is on channel 5500 MHz using 802.11ax with a specific channel width, and that the channel is a DFS frequency.
- **Registered Peers**: Shows the number of devices currently registered (associated) with this access point.

- **Authorized Peers**: Shows how many of the registered peer devices are authorized to communicate (i.e. allowed to pass traffic on the network).
- **Tx Power**: Indicates the transmit power setting of the Wi-Fi interface (typically measured in dBm).
- **Channel Priorities**: Lists the preferred channels (and their wireless modes) in order of priority for the interface. Each entry is prefixed by a priority index, followed by the channel details. For example:
    - 0:5180/ax/Ceee — Priority 0 channel at 5180 MHz using 802.11ax.
    - 0:5260/ax/Ceee/D — Priority 0 channel at 5260 MHz using 802.11ax (with "D" indicating a DFS channel).

3.15 Traffic

The **Traffic** section provides detailed statistics about the current performance and utilization of the wireless interface. These metrics are all **read-only** and are useful for monitoring bandwidth usage, packet flow, and troubleshooting issues related to performance or connectivity. The statistics are typically updated in real-time.

- **Tx/Rx Rate** : Displays the current transmit and receive rates in kilobits per second (kbps).
- **Tx/Rx Packet Rate** : Shows how many packets per second (p/s) are being transmitted and received.
- **FP Tx/Rx Rate** : Indicates the transmit and receive rates processed by the fast path (FP), measured in bits per second (bps).
- **FP Tx/Rx Packet Rate** : Packets per second handled by the fast path system.
- **Tx/Rx Bytes** : Total amount of data transmitted and received by the interface, measured in kibibytes (KiB). 1 KiB = 1024 bytes.
- **Tx/Rx Packets** : Total number of packets transmitted and received by the interface.
- **Tx/Rx Drops** : Number of packets that were dropped (not transmitted or received properly).
- **Tx Queue Drops** : Packets dropped from the transmission queue due to overflow or excessive queue length.
- **Tx/Rx Errors** : Packets with transmission or reception errors.
- **Byte Graph** : A real-time graphical representation of the transmit and receive byte rates over time.
- **Packet Graph** : A real-time graphical chart showing packet rate statistics (Tx and Rx) per second.

4 Interfaces

4.1 Introduction

The **Interfaces** menu lists all network interfaces on the router. This includes physical ports (Ethernet LAN/WAN ports, Wi-Fi radios listed here as wireless interfaces) and virtual interfaces. Interfaces are the building blocks for networking — you assign IP addresses to them, bridge them, or use them for various tunneling protocols. Airolinx 6 treats many things as interfaces, providing a unified way to manage different link types.

By default, all Ethernet ports are listed (e.g., `ether1`, `ether2`, etc.), and Wi-Fi appears as `wlan1` or `wifi1` depending on the driver. A new Wi-Fi6 radio may show as wifi1 in the WiFi menu and as an interface here as well. Some ports might be grouped: for instance, your device might ship with a default bridge combining all LAN ports (then those ports show "slave" to a bridge). We cover Bridges separately in the Bridge section.

Interfaces can also be created for virtual networking functions:

- **VLAN Interfaces:** If you need to tag VLANs on a port, you can add a VLAN interface on top of a physical port or bridge.
- **Tunnel Interfaces:** EoIP, GRE, IPIP, WireGuard, etc., will appear here once created (each is a virtual interface carrying traffic).
- **Bonding Interfaces:** For link aggregation of physical ports.
- **MACsec and MAC VLAN:** Advanced Layer2 interfaces for secure Ethernet or MAC filtering (available on this platform as an option for specialized setups).

4.2 Configuration Steps

1. **Viewing Interfaces:** Click on **Interfaces** in the menu. You will see a table of interfaces, their status (R for running), and some statistics.
2. **Enable/Disable:** To administratively shut down an interface, select it and click "Disable". To bring it back up, "Enable" it. For example, you might disable `ether2` if it's unused to prevent unwanted access.
3. **Rename (optional):** You can rename interfaces (especially bridges or VLANs) for clarity by double-clicking and changing the **Name** field.
4. **Add VLAN:** To create a VLAN interface, click "+", choose **VLAN**. Set VLAN ID and select the **Interface** on which this VLAN will be present (e.g., `ether1` for trunk port). This will create `vlanX` interface that can be used like a normal interface (assign IP, etc.).
5. **Add Bonding:** If you need link aggregation, click "+", choose **Bonding**. Select member ports and bonding mode (e.g., IEEE 802.3ad for LACP). A bonding interface will then appear.
6. **Interface Lists:** Under **Interface List** (in some Airolinx 6 versions, separate menu), you can group interfaces (e.g., for use in firewall or bridge settings). If needed, add interfaces to lists like "LAN" or "WAN" for easier management (for instance, firewall rules referencing "LAN" list).

4.3 Troubleshooting

Link Down issues: If an Ethernet interface shows "link down", check the cable and the remote device. The **Health** or **Ethernet** tab on that interface might show if it's not negotiating. Ensure speed/duplex aren't manually mismatched (by default they auto-negotiate).

VLAN not passing traffic: Make sure the VLAN ID matches on both ends and that the physical port is either in a bridge with VLAN filtering configured or, if using switch chip features, that the VLAN is allowed. On Airolinx 6, if using the software bridge approach, configure VLANs under Bridge settings (Bridge VLANs) for proper tagging on ports.

Cannot add IP to Ether port because "slave": If an interface is part of a bridge (slave), you cannot assign IP directly to it (it's grayed out). Instead, assign to the bridge. Check Bridge section; you might need to put IP on Bridge if bridging LAN ports.

Most of the time, you won't need to change much in the Interfaces menu except when adding VLANs or tunnels. But it's useful to monitor – e.g., the "**RX/TX**" **counters** help verify if data is flowing. You can also see wireless registration info by selecting a Wi-Fi interface and clicking "**Wireless**" tab (shortcut to wireless settings).

4.4 Interface Types

All interfaces—both physical and virtual—are listed under the **Interfaces** tab in the Airolinx 6 web interface. This section provides a summary of available interface types and their use cases.

4.4.1 Default Interfaces

- **bridge** — Type: Bridge
  Used to combine multiple interfaces into a single Layer 2 broadcast domain.
- **ether1** — Type: Ethernet
  The physical Ethernet port of the device. Click here for Ethernet config settings
- **lo** — Type: Loopback
  A virtual interface used for internal routing, testing, or diagnostics.
- **wifi1** — Type: WiFi
  The built-in wireless interface. Click here for Wi-Fi config settings

4.4.2 Adding New Interfaces

To add a new interface, click the **New** button on the Interfaces page. You will be prompted to select an interface type from the available list.

Available Interface Types

- VRRP
- Bridge
- EoIP Tunnel
- IP Tunnel
- VLAN
- Bonding
- 6to4 Tunnel
- Mesh
- VPLS
- GRE Tunnel
- IPIPv6 Tunnel

- EoIPv6 Tunnel
- GRE6 Tunnel
- MACsec
- VXLAN
- WiFi
- WireGuard
- VETH
- L2TP Ethernet
- MACVLAN
- PPTP Server Binding
- PPTP Client
- PPPoE Server Binding
- PPPoE Client
- PPP Server
- PPP Client
- L2TP Server Binding
- L2TP Client
- OVPN Client
- OVPN Server Binding
- SSTP Server Binding
- SSTP Client

4.4.3 Adding a New VRRP Interface

- **Enabled** (read/write): Enables or disables the VLAN interface. Toggle switch.
- **Comment** (read/write): Optional text field for labeling the interface.

General

- **Name** (string): User-defined name for the VRRP interface.
- **Type** (read-only): Displays "VRRP".
- **MTU** : Maximum Transmission Unit size. Default: 1500.
- **Actual MTU** (read-only): Shows effective MTU after creation.
- **L2 MTU** (read-only): Layer 2 MTU, shown post-creation.
- **MAC Address** : MAC address of the interface.
- **ARP** : ARP mode (enabled, disabled, etc.).
- **ARP Timeout** : Timeout for ARP entries.

VRRP

- **Interface** : Select base interface for VRRP instance.
- **VRID** : Virtual Router ID. Used to identify the VRRP group.
- **Priority** : Determines which router becomes master (higher wins).
- **Group Authority** : Designates if router can control group mastership.
- **Interval** : Advertisement interval in milliseconds.
- **Preemption Mode** : Whether higher-priority routers can take over.
- **Authentication** : Method (none, simple, ah).
- **Password** : Password for authentication (if enabled).
- **Version** : VRRP version (2 or 3).

- **V3 Protocol** : Protocol for version 3 (IPv4/IPv6).

Conn. Tracking

- **Sync. Connection Tracking** : Enables session sync between nodes.
- **Remote Address** : Remote peer IP for session sync.

Scripts

- **On Master** : Script executed when becoming master.
- **On Backup** : Script executed when becoming backup.

Status

- Read-only information about the interface state.

Traffic

- Displays traffic statistics.

4.4.4 Adding a New Bridge Interface

- **Enabled**: Toggles the bridge interface on or off.
- **Comment**: Optional field for adding a description or note to the interface.

General

- **Name**: Sets the identifier of the bridge interface (e.g., `bridge1`).
- **Type**: Indicates the interface type (Bridge).
- **MTU**: Maximum Transmission Unit for Layer 3.
- **Actual MTU**: Shows the effective MTU based on interface and configuration.
- **L2 MTU**: Layer 2 MTU size.
- **MAC Address**: MAC address of the bridge interface.
- **ARP**: Controls how the interface handles ARP requests. Options:
  - `enabled`: Accept and respond to ARP.
  - `disabled`: Ignore ARP requests.
  - `local-proxy-arp`: Respond on behalf of local addresses on other interfaces.
  - `proxy-arp`: Respond on behalf of remote addresses.
  - `reply-only`: Respond only to known static ARP entries.
- **ARP Timeout**: Sets the expiration for dynamic ARP entries.
- **Admin. MAC Address**: Optionally forces a specific MAC address.
- **Ageing Time**: Time before a learned MAC address is purged if unused.
- **Max Learned Entries**: Maximum number of MAC entries the bridge can store (`auto` for unlimited).
- **IGMP Snooping**: Enables IGMP snooping to optimize multicast traffic forwarding.
- **DHCP Snooping**: Enables validation of DHCP packets to prevent rogue DHCP servers.
- **Fast Forward**: Allows hardware or accelerated switching of packets through the bridge.

STP (Spanning Tree Protocol)

- **Protocol Mode**: Selects the STP variant: `none`, `STP`, `RSTP`, or `MSTP`.
- **Priority**: Bridge priority value used for STP elections.
- **Port Cost Mode**: Selects calculation mode for port costs: `short` or `long`.
- **Region Name**: MSTP region name.

- **Region Revision**: MSTP configuration revision.
- **Max Message Age**: Maximum time before an STP BPDU is considered stale.
- **Forward Delay**: Time spent in listening and learning states during STP changes.
- **Transmit Hold Count**: Number of BPDUs that can be sent per second.
- **Max Hops**: Max hops allowed in an MSTP region.

VLAN

- **VLAN Filtering**: Enables filtering of VLAN traffic on the bridge.

IGMP Snooping

- **IGMP Version**: Version of IGMP supported (typically 2).
- **MLD Version**: Version of MLD (Multicast Listener Discovery) used for IPv6.
- **Multicast Router**: Determines how the bridge treats multicast routers. Options:
  - Disabled
  - Permanent
  - Temporary Query
- **Multicast Querier**: Enables bridge to act as an IGMP querier if no other is present.
- **Startup Query Count**: Number of initial queries sent at startup.
- **Last Member Query Count**: Number of queries sent before pruning a group.
- **Last Member Interval**: Time to wait for a membership report after sending a last-member query.
- **Membership Interval**: How long to keep group membership if no report is heard.
- **Querier Interval**: Time between general queries.
- **Query Interval**: Time between sending standard IGMP queries.
- **Query Response Interval**: Time to wait for responses to queries.
- **Startup Query Interval**: Interval between initial queries during startup.

Status

- **Last Link Down Time**: Timestamp of the last link down event.
- **Last Link Up Time**: Timestamp of the last link up event.
- **Link Downs**: Number of times the link has gone down.

Traffic

- **Tx/Rx Rate**: Current transmit and receive rate in bits per second.
- **Tx/Rx Packet Rate**: Current packet rate (packets per second).
- **FP Tx/Rx Rate**: Fast path transmit and receive rate.
- **FP Tx/Rx Packet Rate**: Fast path packet rate.
- **Tx/Rx Bytes**: Total bytes sent and received.
- **Tx/Rx Packets**: Total packets sent and received.
- **Tx/Rx Drops**: Number of packets dropped on transmit and receive.
- **Tx Queue Drops**: Packets dropped due to queue overflow.
- **Tx/Rx Errors**: Packet errors on transmit and receive.
- **Byte Graph**: Visual graph of Tx/Rx bytes over time.
- **Packet Graph**: Visual graph of Tx/Rx packets over time.

4.4.5 Adding a New EoIP Tunnel Interface

The EoIP (Ethernet over IP) tunnel allows you to create Layer 2 tunnels between remote routers over an IP connection. It is commonly used for bridging Ethernet networks across routed networks.

- **Enabled** (read/write): Enables or disables the VLAN interface. Toggle switch.
- **Comment** (read/write): Optional text field for labeling the interface.

General

- **Name** : The name of the interface (e.g., eoip-tunnel1).
- **Type** : (Read-only) Displays the interface type as EoIP Tunnel.
- **MTU / Actual MTU / L2 MTU** : Maximum Transmission Unit settings for controlling frame sizes.
- **MAC Address** : The MAC address assigned to this interface.
- **ARP**
- **ARP**: Controls how the interface handles ARP requests. Options:
    - enabled: Accept and respond to ARP.
    - disabled: Ignore ARP requests.
    - local-proxy-arp: Respond on behalf of local addresses on other interfaces.
    - proxy-arp: Respond on behalf of remote addresses.
    - reply-only: Respond only to known static ARP entries.
- **ARP Timeout** : Optionally set a timeout for ARP entries.

Tunnel Settings

- **Local Address** : IP address of the local device (optional).
- **Remote Address** : IP address of the remote EoIP peer.
- **Tunnel ID** : Must match between both tunnel endpoints.
- **IPSec Secret** : Optional key for encrypting the tunnel traffic using IPsec.

Keepalive

- **Keepalive** : Enables failure detection by sending keepalive packets. Format: interval,retry count.

Traffic Control

- **DSCP** : Differentiated Services Code Point setting for packet marking. Default is inherit.
- **Don't Fragment** : Prevent fragmentation of EoIP packets.
- **Clamp TCP MSS** : Automatically adjusts TCP Maximum Segment Size to avoid fragmentation.
- **Allow Fast Path** : Enables accelerated packet processing when supported.

Loop Protect

- **Loop Protect** : Enables protection against Layer 2 loops (default, on, or off).
- **Send Interval** : Frequency for sending loop detection packets.
- **Disable Time** : Time to disable the interface after loop detection.

Status

- **Last Link Down Time / Last Link Up Time** : Most recent timestamp of interface status change.
- **Link Downs** : Number of times the link has transitioned down.

Traffic

- **Tx/Rx Rate** : Current, average, and max transmit/receive throughput (in bps).

- **Tx/Rx Packet Rate** : Packet rates per second.
- **FP Tx/Rx Rate** : Fast path traffic rate.
- **Tx/Rx Bytes / Packets / Drops / Errors** : Traffic counters.
- **Tx Queue Drops** : Number of dropped packets in queue.
- **Byte Graph / Packet Graph** : Visual representation of interface traffic over time.

4.4.6 Adding a New IP Tunnel Interface

IP Tunnel creates a simple IP-over-IP tunnel between two endpoints, useful for routing over non-routing networks.

- **Enabled** (read/write): Enables or disables the VLAN interface. Toggle switch.
- **Comment** (read/write): Optional text field for labeling the interface.

General Settings:

- **Name**: Name of the interface (e.g., ipip-tunnel1).
- **Type**: Displays IP Tunnel (read-only).
- **MTU**: Maximum packet size (default set by system).
- **Actual MTU / L2 MTU**: Displayed after interface is created (read-only).
- **MAC Address**: Auto-generated or manually assigned.
- **ARP / ARP Timeout**: Controls ARP behavior and expiration.

Tunnel Configuration:

- **Local Address**: Local tunnel endpoint IP.
- **Remote Address**: Remote peer IP address.
- **IPSec Secret**: Optional shared secret for encryption.

Tunnel Behavior:

- **Keepalive**: Time interval and retry count for link checking.
- **DSCP**: Traffic marking for QoS (e.g., inherit).
- **Don't Fragment**: Enable/disable IP fragmentation flag.
- **Clamp TCP MSS**: Adjusts TCP MSS to prevent fragmentation.
- **Allow Fast Path**: Enables hardware acceleration for improved performance.

Status:

- **Last Link Down Time / Up Time**: Timestamp of link status changes.
- **Link Downs**: Number of disconnections recorded.

Traffic:

- **Tx/Rx Rate & Packet Rate**: Real-time transmit/receive stats.
- **FP Tx/Rx Rate**: Hardware-accelerated throughput.
- **Tx/Rx Bytes & Packets**: Cumulative traffic counters.
- **Drops / Queue Drops / Errors**: Indicators of transmission issues.
- **Byte Graph / Packet Graph**: Visual performance history.

4.4.7 Adding a New VLAN Interface

- **Enabled** (read/write): Enables or disables the VLAN interface. Toggle switch.

- **Comment** (read/write): Optional text field for labeling the interface.

General

- **Name** (read/write): The name of the VLAN interface (e.g., vlan1).
- **Type** (read-only): Displays the interface type, in this case, "VLAN".
- **MTU** (read/write): Sets the maximum transmission unit size in bytes. Default is 1500.
- **Actual MTU** (read-only): Displays the effective MTU after configuration.
- **L2 MTU** (read-only): Displays Layer 2 MTU size.
- **MAC Address** (read-only): Displays the MAC address of the VLAN interface.
- **ARP** (read/write): Sets the Address Resolution Protocol mode.
  - enabled: Accept and respond to ARP.
  - disabled: Ignore ARP requests.
  - local-proxy-arp: Respond on behalf of local addresses on other interfaces.
  - proxy-arp: Respond on behalf of remote addresses.
  - reply-only: Respond only to known static ARP entries.
- **ARP Timeout** (read/write): Custom timeout value for ARP entries.
- **VLAN ID** (read/write): Numerical VLAN ID associated with this interface.
- **Interface** (read/write): Select the physical interface this VLAN will be assigned to (e.g., bridge).
- **Use Service Tag** (read/write): Enables tagging using service VLAN (S-VLAN).
- **MVRP** (read/write): Enables Multiple VLAN Registration Protocol for dynamic VLAN learning.

Loop Protect

- **Loop Protect** : Enables loop protection on the interface (default, on, off).
- **Send Interval** : Time interval between loop detection packets.
- **Disable Time** : How long the interface is disabled after loop detection.

Status

- Last Link Down Time
- Last Link Up Time
- Link Downs (count)

Traffic

- Tx/Rx Rate (bps)
- Tx/Rx Packet Rate (p/s)
- FP Tx/Rx Rate and Packet Rate
- Tx/Rx Bytes, Packets, Drops, Queue Drops, Errors
- Byte Graph / Packet Graph (real-time visualization)

4.4.8 Adding a New Bonding Interface

- **Enabled** (read/write): Toggles the bonding interface on or off.
- **Comment** (read/write): Optional text field for naming or describing the interface.

General

- **Name** (read/write): The name of the bonding interface (e.g., bonding1).
- **Type** (read-only): Displays “Bonding”.
- **MTU** (read/write): Maximum Transmission Unit size. Default: 1500.

- **Actual MTU** (read-only): Calculated MTU based on configuration.
- **L2 MTU** (read-only): Layer 2 MTU size.
- **MAC Address** (read-only): Hardware MAC address.
- **ARP** (read/write): Address Resolution Protocol mode (enabled/disabled/other options).
    - enabled: Accept and respond to ARP.
    - disabled: Ignore ARP requests.
    - local-proxy-arp: Respond on behalf of local addresses on other interfaces.
    - proxy-arp: Respond on behalf of remote addresses.
    - reply-only: Respond only to known static ARP entries.
- **ARP Timeout** (read/write): Custom timeout for ARP entries.
- **Forced MAC Address** (read/write): Sets a custom MAC address to override the default.

Bonding

- **Slaves** (read/write): Add physical interfaces to participate in the bond.
- **Mode** (read/write): Bonding strategy. Options include balance-rr, active-backup, etc.
- **Primary** (read/write): Primary slave used when in active-backup mode.
- **Link Monitoring** (read/write): Type of monitoring (e.g., mii, arp).
- **Transmit Hash Policy** (read/write): Defines how traffic is distributed (e.g., layer 2).
- **Min. Links** (read/write): Minimum number of links required to activate the bond.
- **Down Delay** (read/write): Time to wait before disabling a link, in ms.
- **Up Delay** (read/write): Time to wait before enabling a recovered link, in ms.
- **LACP Rate** (read/write): Frequency of LACPDU transmission (e.g., 30s).
- **LACP User Key** (read/write): Unique key identifier for LACP sessions.
- **MLAG ID** (read/write): Multi-chassis Link Aggregation Group ID.
- **LACP Mode** (read/write): Mode for LACP operation (e.g., active, passive).
- **MII Interval** (read/write): Interval for MII monitoring.

Status

- **Last Link Down Time** (read-only): Timestamp of last link-down event.
- **Last Link Up Time** (read-only): Timestamp of last link-up event.
- **Link Downs** (read-only): Number of times the link has gone down.
- **Mode** (read-only): Current operational mode of the bonding interface.
- **Active Ports** (read-only): Currently active bonded ports.
- **Inactive Ports** (read-only): Currently inactive bonded ports.
- **LACP System ID** (read-only): System ID used for LACP negotiation.
- **LACP System Priority** (read-only): System priority in LACP operations.
- **LACP Partner System ID** (read-only): System ID of connected LACP peer.

Traffic

- **Tx/Rx Rate** (read-only): Transmit/receive bandwidth in bits per second.
- **Tx/Rx Packet Rate** (read-only): Number of packets sent/received per second.
- **FP Tx/Rx Rate** (read-only): Fast path Tx/Rx bandwidth (if applicable).
- **FP Tx/Rx Packet Rate** (read-only): Fast path packet rate.
- **Tx/Rx Bytes** (read-only): Total data sent and received in bytes.
- **Tx/Rx Packets** (read-only): Total packets sent and received.
- **Tx/Rx Drops** (read-only): Dropped packets count.
- **Tx Queue Drops** (read-only): Packets dropped due to queue overflow.

- **Tx/Rx Errors** (read-only): Transmission/reception error count.
- **Byte Graph** : Graphical representation of throughput over time.
- **Packet Graph** : Graphical representation of packet rate over time.

4.4.9 Adding a New 6to4 Tunnel Interface

- Enabled (read/write): Toggles the bonding interface on or off.
- Comment (read/write): Optional text field for naming or describing the interface.

General

- **Name** — (string) The name assigned to the 6to4 tunnel interface (e.g., 6to4-tunnel1).
- **Type** — (read-only) Automatically set to 6to4 Tunnel.
- **MTU** — (integer) Maximum Transmission Unit for the interface. Controls the largest packet size allowed.
- **Actual MTU** — (read-only) Displays the current calculated MTU in use.
- **L2 MTU** — (read-only) The Layer 2 MTU value of the interface.
- **Local Address** — (IP) Source address used for the tunnel. This must be a local IPv6 address on the router.
- **Remote Address** — (IP) Destination address of the tunnel. If not specified, the IPv4 address is derived from the 2002::/16 6to4 prefix.
- **IPSec Secret** — (string) Optional pre-shared key for encrypting the tunnel using dynamic IPsec (uses SHA1/AES-128-CBC).
- **Keepalive** — (time + retries) Interval and retry attempts for keeping the tunnel up. Format: KeepaliveInterval,KeepaliveRetries. Default is 10s,10 retries.
- **DSCP** — (0—63 or inherit) Defines the Differentiated Services Code Point used for QoS. inherit uses the value from the encapsulated packet.
- **Don't Fragment** — (yes | no | inherit) Controls the use of the DF (Don't Fragment) bit:
  - no : fragment if needed.
  - inherit : follow the DF setting of the original packet.
- **Clamp TCP MSS** — (yes | no; default: yes) When enabled, adjusts the TCP Maximum Segment Size for incoming SYN packets to prevent fragmentation across the tunnel.

Status

- **Last Link Down Time** — Timestamp of the last event when the tunnel interface went down.
- **Last Link Up Time** — Timestamp of the most recent time the tunnel came up.
- **Link Downs** — Number of times the tunnel interface has gone down.

Traffic

- **Tx/Rx Rate** — Current transmit/receive rates in bits per second.
- **Tx/Rx Packet Rate** — Packets per second transmitted and received.
- **FP Tx/Rx Rate** — FastPath Tx/Rx rates in bits per second.
- **FP Tx/Rx Packet Rate** — FastPath packet rates (pps).
- **Tx/Rx Bytes** — Total transmitted and received bytes.
- **Tx/Rx Packets** — Total number of packets transmitted and received.
- **Tx/Rx Drops** — Dropped packets count for transmit and receive.
- **Tx Queue Drops** — Packets dropped from the transmit queue.
- **Tx/Rx Errors** — Transmit and receive error counts.

- **Byte Graph** — A graphical representation of transmitted and received bytes over time.
- **Packet Graph** — Graph of packets sent and received per second over time.

4.4.10 Adding a New Mesh Interface

- **Enabled** — Toggle to enable or disable the Mesh interface.
- **Comment** — Optional field to describe the interface.

General

- **Name** — Unique name for the interface (e.g., mesh1).
- **Type** — Displays "Mesh" as the interface type (read-only).
- **MTU** — Maximum Transmission Unit (default: 1500).
- **Actual MTU** — Displays current MTU after configuration (read-only).
- **L2 MTU** — Layer 2 MTU (read-only).
- **MAC Address** — Hardware address of the interface.
- **ARP** — Sets ARP behavior (e.g., enabled, disabled, proxy-arp).
    - enabled: Accept and respond to ARP.
    - disabled: Ignore ARP requests.
    - local-proxy-arp: Respond on behalf of local addresses on other interfaces.
    - proxy-arp: Respond on behalf of remote addresses.
    - reply-only: Respond only to known static ARP entries.
- **ARP Timeout** — Optional timeout value for ARP entries.
- **Admin. MAC Address** — Optional manually configured MAC address.

HWMP

- **Mesh Portal** — Enables mesh portal mode (acts as entry/exit point for mesh network).
- **Default Hoplimit** — Maximum number of hops allowed (default: 32).
- **PREQ Waiting Time** — Time to wait before sending a Path Request (default: 4 seconds).
- **PREQ Retries** — Number of retry attempts for sending PREQ (default: 2).
- **PREQ Destination Only** — Sends PREQ only to destination (toggle).
- **PREQ Reply and Forward** — Enables both forwarding and replying to PREQs (toggle).
- **PREP Lifetime** — Validity duration of Path Reply (default: 5 minutes).
- **RANN Interval** — Time between Root Announcements (default: 10 seconds).
- **RANN Propagation Delay** — Propagation delay value in milliseconds (default: 500).
- **RANN Lifetime** — Duration the Root Announcement is valid (default: 22 seconds).
- **Reoptimize Paths** — Toggle to allow dynamic re-optimization of mesh routes.

Status

- **Last Link Down Time** — Timestamp of the last time the link went down.
- **Last Link Up Time** — Timestamp of the most recent successful link-up.
- **Link Downs** — Number of link-down events recorded.

Traffic

- **Tx/Rx Rate** — Current transmit and receive data rate in bits per second.
- **Tx/Rx Packet Rate** — Packet rate for transmit and receive (packets per second).
- **FP Tx/Rx Rate** — Fast Path transmit and receive rate.
- **FP Tx/Rx Packet Rate** — Fast Path packet rate per second.

- **Tx/Rx Bytes** — Total data sent and received (bytes).
- **Tx/Rx Packets** — Total packets sent and received.
- **Tx/Rx Drops** — Number of dropped packets in each direction.
- **Tx Queue Drops** — Number of packets dropped from the transmission queue.
- **Tx/Rx Errors** — Errors encountered during transmission or reception.
- **Byte Graph** — Graphical history of bandwidth usage.
- **Packet Graph** — Graphical history of packet rate.

4.4.11 Adding a New VPLS Interface

- **Enabled** — Toggle to enable or disable the Mesh interface.
- **Comment** — Optional field to describe the interface.

General

- **Name** — Name of the VPLS interface.
- **Type** — Read-only; displays the type as "VPLS".
- **MTU** — Sets the Maximum Transmission Unit size (default is typically 1500).
- **Actual MTU** — Read-only; displays the active MTU size.
- **L2 MTU** — Read-only; shows the Layer 2 MTU.
- **MAC Address** — Hardware address of the interface.
- **ARP** — Select ARP mode (enabled, disabled, reply-only, etc.).
    - enabled: Accept and respond to ARP.
    - disabled: Ignore ARP requests.
    - local-proxy-arp: Respond on behalf of local addresses on other interfaces.
    - proxy-arp: Respond on behalf of remote addresses.
    - reply-only: Respond only to known static ARP entries.
- **ARP Timeout** — Sets the timeout value for ARP entries.
- **Remote Peer** — IP address of the remote VPLS peer.
- **VPLS ID** — Unique identifier for the VPLS tunnel.
- **Cisco Static ID** — Toggle to enable Cisco-compatible static ID assignment.

VPLS

- **Bridge** — Assign the interface to an existing bridge.
- **Bridge Cost** — Sets the cost of this port in the bridge topology.
- **Bridge Horizon** — Defines split-horizon group for loop prevention.
- **Bridge PVID** — Port VLAN ID when bridging VLAN-tagged traffic.
- **PW Type** — Specifies the pseudowire type (e.g., Ethernet, VLAN).
- **PW L2MTU** — L2 MTU setting for the pseudowire tunnel.
- **PW Control Word** — Toggle to enable/disable MPLS control word.
- **BGP VPLS** — Read-only status of BGP VPLS integration.
- **BGP VPLS Prefix** — Displays BGP VPLS route target prefix (if set).

Status

- **Last Link Down Time** — Timestamp of the last link-down event.
- **Last Link Up Time** — Timestamp of the last link-up event.
- **Link Downs** — Count of how many times the interface has gone down.
- **Remote Label / Local Label** — Displays MPLS label info.

- **Remote Status** — Shows tunnel health:
  - Not forwarding
  - Attachment circuit TX fault
  - Attachment circuit RX fault
  - PW TX fault
  - PW RX fault
- **Remote Group / TE Tunnel** — Displays remote group or tunnel info.

Traffic

- **Tx/Rx Rate** — Real-time transmit/receive throughput.
- **Tx/Rx Packet Rate** — Packets per second (Tx/Rx).
- **FP Tx/Rx Rate** — FastPath throughput.
- **FP Tx/Rx Packet Rate** — FastPath packets per second.
- **Tx/Rx Bytes / Packets / Drops / Errors** — Interface-level traffic counters.
- **Tx Queue Drops** — Number of packets dropped from queue.
- **Graphs** — Live charts of:
  - **Byte Graph** — Displays Tx/Rx bytes over time.
  - **Packet Graph** — Shows packet rate over time.

Troubleshooting VPLS Interfaces

- **MTU Mismatch**: Ensure the MTU of all interfaces is large enough (≥ 1508).
- **Missing BGP or LDP Signaling**: Verify BGP/LDP sessions are established and routes exchanged.
- **Incorrect Bridge Configuration**: Add the VPLS interface to the correct bridge with all ports.
- **Incorrect VPLS ID or Peer Address**: Check VPLS IDs and peer IP addresses match on both sides.
- **Check Interface Status**: Review the status tab for errors or inactive states.
- **No Traffic Passing**: Inspect bridge split-horizon settings or firewall rules blocking traffic.

4.4.12 Adding a New GRE Tunnel Interface

- **Enabled** — Toggle to enable or disable the Mesh interface.
- **Comment** — Optional field to describe the interface.

General

- **Name** — A user-defined name for the GRE tunnel interface (e.g., gre-tunnel1).
- **Type** — This field is read-only and identifies the interface type as "GRE Tunnel".
- **MTU** — Maximum Transmission Unit; defines the largest size of packets that can be transmitted (default is typically 1476).
- **Actual MTU** — Read-only field showing the current MTU used after system adjustments.
- **L2 MTU** — Read-only field showing Layer 2 MTU; may be used for bridging or tunneling applications.
- **Local Address** — IP address of the local GRE tunnel endpoint.
- **Remote Address** — IP address of the remote GRE tunnel endpoint. Required to establish the tunnel.
- **IPsec Secret** — Optional pre-shared key to secure GRE traffic using IPsec encryption. When set, traffic is encapsulated in IPsec ESP using AES-128 with SHA1.
- **Keepalive**
  - **Interval** — Time between keepalive probes (e.g., 00:00:10).
  - **Retries** — Number of failed probes before the tunnel is marked as down.

- **DSCP** — Differentiated Services Code Point value for marking packets. Default is inherit (value is taken from the original packet).
- **Don't Fragment** — Defines packet fragmentation behavior:
  - no — Allows fragmentation when needed.
  - inherit — Uses the DF (Don't Fragment) flag from the original packet.
- **Clamp TCP MSS** — Automatically adjusts the maximum segment size in TCP SYN packets to avoid fragmentation in tunnels. Enabled by default.
- **Allow Fast Path** — When enabled, bypasses standard packet processing for improved performance (if supported by configuration).

Status

- **Last Link Down Time** — Timestamp of the last time the tunnel went down.
- **Last Link Up Time** — Timestamp of the last successful tunnel establishment.
- **Link Downs** — Count of how many times the tunnel has gone down.

Traffic

- **Tx/Rx Rate** — Real-time transmit and receive bandwidth usage (bps).
- **Tx/Rx Packet Rate** — Number of packets transmitted/received per second.
- **FP Tx/Rx Rate** — Bandwidth transferred using fast path processing.
- **FP Tx/Rx Packet Rate** — Packet rate using fast path.
- **Tx/Rx Bytes** — Cumulative data transferred over the interface.
- **Tx/Rx Packets** — Number of packets transmitted/received over the tunnel.
- **Tx/Rx Drops** — Number of dropped packets in each direction.
- **Tx Queue Drops** — Packets dropped due to transmit queue overflow.
- **Tx/Rx Errors** — Number of packets with errors.
- **Byte Graph** — Graphical representation of bandwidth usage.
- **Packet Graph** — Graphical view of packet throughput trends.

4.4.13 Adding a New IPIPv6 Tunnel Interface

- **Enabled** — Toggle to enable or disable the Mesh interface.
- **Comment** — Optional field to describe the interface.

General

- **Name** — The name assigned to this interface (e.g., ipipv6-tunnel1).
- **Type** — This field is set to IPIPv6 Tunnel and is read-only.
- **MTU** — Maximum Transmission Unit for the tunnel.
- **Actual MTU** — Displays the actual MTU; read-only.
- **L2 MTU** — Layer 2 MTU size; read-only.
- **Local Address** — IPv6 address used as the tunnel source.
- **Remote Address** — IPv6 address of the remote endpoint.
- **IPSec Secret** — Pre-shared key for encrypting the tunnel traffic.
- **Keepalive** — Interval and retry count for tunnel keepalive.
- **DSCP** — Packet marking for QoS. Typically set to 'inherit'.
- **Clamp TCP MSS** — Enables TCP MSS clamping for the tunnel.

Status

- **Last Link Down Time** — Timestamp of the last interface down event.
- **Last Link Up Time** — Timestamp of the last interface up event.
- **Link Downs** — Number of times the link has gone down.

Traffic

- **Tx/Rx Rate** — Displays current transmission and receive rates.
- **Tx/Rx Packet Rate** — Number of packets per second.
- **FP Tx/Rx Rate** — Fast Path transmit and receive throughput.
- **FP Tx/Rx Packet Rate** — Fast Path packet rate.
- **Tx/Rx Bytes** — Total data sent and received in bytes.
- **Tx/Rx Packets** — Total number of packets sent and received.
- **Tx/Rx Drops** — Packets dropped during transmission or reception.
- **Tx Queue Drops** — Drops due to transmit queue limitations.
- **Tx/Rx Errors** — Transmission/reception errors.
- **Byte Graph** — Live graph showing byte throughput over time.
- **Packet Graph** — Live graph showing packet flow over time.

4.4.14 Adding a New EoIPv6 Tunnel Interface

- **Enabled** — Toggle to enable or disable the EoIPv6 interface.
- **Comment** — Optional field to describe the interface.

General

- **Name** — Name of the EoIPv6 interface (e.g., eoipv6-tunnel1).
- **Type** — Displays the interface type (EoIPv6 Tunnel).
- **MTU** — Maximum Transmission Unit size for the interface.
- **Actual MTU** — Displays the actual MTU negotiated.
- **L2 MTU** — Layer 2 MTU size.
- **MAC Address** — MAC address assigned to the tunnel.
- **ARP** — Controls how the interface handles ARP requests. Options:
  - enabled: Accept and respond to ARP.
  - disabled: Ignore ARP requests.
  - local-proxy-arp: Respond on behalf of local addresses on other interfaces.
  - proxy-arp: Respond on behalf of remote addresses.
  - reply-only: Respond only to known static ARP entries.
- **ARP Timeout** — Time duration after which unused ARP entries expire.
- **Local Address** — Local IPv6 address used by the tunnel.
- **Remote Address** — Remote IPv6 address of the tunnel peer.
- **Tunnel ID** — Identifier to distinguish between multiple tunnels between the same peers.
- **IPSec Secret** — Optional pre-shared key for IPsec encryption.
- **Keepalive** — Interval and retry count for sending keepalive packets to monitor tunnel health.
- **DSCP** — Differentiated Services Code Point value for packet prioritization. Default is inherit.
- **Clamp TCP MSS** — Toggle to enable adjustment of the TCP MSS to avoid fragmentation.

Loop Protect

- **Loop Protect** — Enables protection against Layer 2 loops. Options include default, yes, or no.
- **Send Interval** — Time interval between loop protect probe messages.

- **Disable Time** — Duration for which the port is disabled if a loop is detected.

Status

- **Last Link Down Time** — Timestamp of the last time the interface went down.
- **Last Link Up Time** — Timestamp of the last time the interface came up.
- **Link Downs** — Number of times the interface has gone down.

Traffic

- **Tx/Rx Rate** — Transmit and receive rates in bits per second (bps).
- **Tx/Rx Packet Rate** — Transmit and receive packet rates in packets per second (p/s).
- **FP Tx/Rx Rate** — Fast Path transmit and receive rates.
- **FP Tx/Rx Packet Rate** — Fast Path transmit and receive packet rates.
- **Tx/Rx Bytes** — Total bytes transmitted and received.
- **Tx/Rx Packets** — Total packets transmitted and received.
- **Tx/Rx Drops** — Number of packets dropped on transmit and receive.
- **Tx Queue Drops** — Number of packets dropped in the transmit queue.
- **Tx/Rx Errors** — Number of transmit and receive errors.
- **Byte Graph** — Graphical view of byte-level throughput.
- **Packet Graph** — Graphical view of packet-level throughput.

4.4.15 Adding a New GRE6 Tunnel Interface

- **Enabled** — Toggle to activate or deactivate the GRE6 tunnel interface.
- **Comment** — Optional text field to describe the interface.

General

- **Name** — The name assigned to the GRE6 tunnel interface (e.g., `gre6-tunnel1`).
- **Type** — Displays the interface type, which is `GRE6 Tunnel`.
- **MTU** — Sets the maximum transmission unit for the interface.
- **Actual MTU** — Displays the current operational MTU size.
- **L2 MTU** — Optional setting for the Layer 2 MTU value.
- **Local Address** — IPv6 source address for tunnel packets sent from this device.
- **Remote Address** — IPv6 destination address for tunnel packets sent to the remote peer.
- **IPsec Secret** — Optional field for entering a pre-shared key to apply dynamic IPsec protection to the tunnel.
- **Keepalive** — Defines how often the system should send keepalive packets and the number of failed attempts before marking the interface as down. Example: `00:00:10,10`.
- **DSCP** — Sets the Differentiated Services Code Point (DSCP) value for tunnel traffic.
    - `inherit`: DSCP is inherited from the encapsulated packet.
    - `0—63`: Manually assign a DSCP value.
- **Clamp TCP MSS** — Toggle to automatically adjust the Maximum Segment Size (MSS) in TCP packets to avoid fragmentation when tunneling traffic.

Status

- **Last Link Down Time** — Timestamp of the most recent time the tunnel link went down.
- **Last Link Up Time** — Timestamp of the most recent time the tunnel link came up.
- **Link Downs** — Number of times the tunnel link has gone down.

Traffic

- **Tx/Rx Rate** — Displays current transmit and receive traffic in bits per second.
- **Tx/Rx Packet Rate** — Current transmit and receive packet rates in packets per second.
- **FP Tx/Rx Rate** — Hardware Fast Path transmit and receive rates, if applicable.
- **FP Tx/Rx Packet Rate** — Fast Path packet rates in packets per second.
- **Tx/Rx Bytes** — Total bytes transmitted and received through this interface.
- **Tx/Rx Packets** — Total packets sent and received.
- **Tx/Rx Drops** — Number of dropped packets during transmit/receive operations.
- **Tx Queue Drops** — Packets dropped due to output queue overflows.
- **Tx/Rx Errors** — Errors encountered during transmission and reception.
- **Byte Graph** — Real-time traffic graph displaying transmitted and received byte trends.
- **Packet Graph** — Real-time graph showing transmit and receive packet activity.

4.4.16 Adding a New MACsec Interface

- **Enabled** — Toggle to enable or disable the MACsec interface.
- **Comment** — Optional field to describe the interface.

General

- **Name** — The name assigned to this MACsec interface (e.g., `macsec1`).
- **Type** — Displays the interface type (`MACsec`).
- **MTU** — Sets the Maximum Transmission Unit size for the interface.
- **Actual MTU** — The resulting MTU value after applying configuration and system limits.
- **L2 MTU** — The Layer 2 MTU, indicating the maximum frame size excluding the Ethernet header.
- **MAC Address** — Displays the hardware address associated with the interface.
- **Interface** — The physical or logical interface (e.g., `ether1`) used for MACsec encapsulation.
- **CAK** — Connectivity Association Key used for encryption. Must match on both ends.
- **CKN** — Connectivity Association Key Name, which identifies the CAK.
- **Profile** — Defines the MACsec profile to use. Options:
    - `default` — Uses the system default MACsec profile settings.

Status

- **Last Link Down Time** — Timestamp of the most recent interface link down event.
- **Last Link Up Time** — Timestamp of the most recent interface link up event.
- **Link Downs** — Total number of times the link has gone down.

Traffic

- **Tx/Rx Rate** — Current transmit and receive bandwidth usage in bits per second.
- **Tx/Rx Packet Rate** — Transmit and receive packet rates in packets per second.
- **FP Tx/Rx Rate** — Fast Path transmit and receive rates.
- **FP Tx/Rx Packet Rate** — Fast Path packet rates.
- **Tx/Rx Bytes** — Total number of bytes transmitted and received.
- **Tx/Rx Packets** — Total number of packets transmitted and received.
- **Tx/Rx Drops** — Number of packets dropped during transmission and reception.
- **Tx Queue Drops** — Number of packets dropped in the transmit queue.
- **Tx/Rx Errors** — Number of transmit and receive errors.

- **Byte Graph** — Real-time graph showing Tx/Rx throughput over time.
- **Packet Graph** — Real-time graph showing Tx/Rx packet rate over time.

4.4.17 Adding a New VXLAN Interface

- **Enabled** — Toggle to enable or disable the VXLAN interface.
- **Comment** — Optional field to describe the interface.

General

- **Name** — The identifier for this VXLAN interface.
- **Type** — Interface type, set as VXLAN.
- **MTU** — Maximum Transmission Unit in bytes.
- **Actual MTU** — The calculated MTU based on configuration.
- **L2 MTU** — The maximum frame size at Layer 2.
- **MAC Address** — MAC address of the interface.
- **ARP** — Controls ARP behavior. Options:
  - enabled — Accept and respond to ARP.
  - disabled — Ignore ARP requests.
  - local-proxy-arp — Respond on behalf of local addresses on other interfaces.
  - proxy-arp — Respond on behalf of remote addresses.
  - reply-only — Respond only to known static ARP entries.
- **ARP Timeout** — Time before ARP entries expire.
- **VNI** — VXLAN Network Identifier.
- **Group** — IP multicast group for VXLAN traffic distribution.
- **Interface** — The physical interface VXLAN is bound to.
- **Port** — UDP port for VXLAN encapsulation (default is 4789).
- **Local Address** — The source IP address for VXLAN tunnels.
- **Don't Fragment** — Controls fragmentation behavior. Options:
  - auto — Automatically determined.
  - enabled — Force fragmentation behavior.
  - disabled — Disable fragmentation.
- **Max FDB Size** — Maximum number of entries in the forwarding database.
- **TTL** — Time To Live value for VXLAN packets. Options include:
  - auto
  - Numeric values (e.g., 64)
- **VTEPS IP Version** — IP version used by VXLAN tunnel endpoints. Options:
  - IPv4
  - IPv6
- **VRF** — Virtual Routing and Forwarding instance for this VXLAN.
- **Allow Fast Path** — Enables high-performance forwarding path for packets.
- **Bridge** — Bridge interface to associate with the VXLAN.
- **Bridge PVID** — VLAN ID assigned to untagged packets entering the bridge.
- **HW** — Enables hardware offload where supported.

Loop Protect

- **Loop Protect** — Enables loop detection and protection. Options:
  - default

    - on
    - off
- **Send Interval** — How often to send loop detection messages.
- **Disable Time** — How long to disable the interface if a loop is detected.

Status

- **Last Link Down Time** — Timestamp of the last time the interface went down.
- **Last Link Up Time** — Timestamp of the last time the interface came up.
- **Link Downs** — Number of times the link has gone down.

Traffic

- **Tx/Rx Rate** — Current transmission and reception rates.
- **Tx/Rx Packet Rate** — Number of packets sent and received per second.
- **FP Tx/Rx Rate** — Fast Path transmit and receive rate.
- **FP Tx/Rx Packet Rate** — Fast Path packet rate.
- **Tx/Rx Bytes** — Total transmitted and received bytes.
- **Tx/Rx Packets** — Total transmitted and received packets.
- **Tx/Rx Drops** — Dropped packets during transmission or reception.
- **Tx Queue Drops** — Number of packets dropped from the output queue.
- **Tx/Rx Errors** — Number of errors encountered.
- **Byte Graph** — Visual graph of byte throughput over time.
- **Packet Graph** — Visual graph of packet throughput over time.

4.4.18 Adding a New WiFi Interface

- **Enabled** — Toggle to enable or disable the WiFi interface.
- **Comment** — Optional field to describe the interface.

General

- **Name**: Sets the name for the new WiFi interface. The default is `wifi<x>`, where `<x>` is the next available number (e.g., if `wifi1` exists, the default name is `wifi2`).
- **Master**: Specifies the physical WiFi interface this virtual WiFi interface is based on. This must be a physical interface such as `wifi1` or `wifi2` on dual-band models.

Click here to go to the WiFi section to view and configure the remaining interface settings.

Note:

- **Mode Inheritance**: A virtual interface inherits the mode of its master. For example, if the master is set to "AP bridge," the virtual interface must also operate in "AP bridge" mode. You cannot assign it a different mode such as "station."
- **Master Interface**: This is the base physical WiFi interface. Virtual interfaces depend on it for functionality.
- **Limitations**:
    - Virtual interfaces cannot operate independently of the master interface.
    - They use the same frequency as the master.

4.4.19 Adding a New WireGuard Interface

- **Enabled** — Toggle to enable or disable the WireGuard interface.
- **Comment** — Optional field to describe the interface.

General

- **Name** — The name assigned to the interface (e.g., wireguard1).
- **Type** — Displays the interface type, which is WireGuard.
- **MTU** — Maximum Transmission Unit for the interface. Defaults to 1420.
- **Actual MTU** — Displays the calculated MTU based on configuration (read-only).
- **Listen Port** — UDP port that this WireGuard interface will listen on (e.g., 13231).
- **Private Key** — A base64 private key. If not specified, it will be automatically generated upon interface creation. Each network interface has a private key and a list of peers.
- **Public Key** — Automatically derived from the private key and used for identification by peers.

Status

- **Last Link Down Time** — Timestamp of the last time the interface went down.
- **Last Link Up Time** — Timestamp of the last time the interface came up.
- **Link Downs** — Number of times the interface has gone down.

Traffic

- **Tx/Rx Rate** — Transmit and receive data rates (bps).
- **Tx/Rx Packet Rate** — Number of packets transmitted and received per second.
- **FP Tx/Rx Rate** — FastPath data rates for transmitted/received traffic.
- **FP Tx/Rx Packet Rate** — Packet rate using FastPath for both directions.
- **Tx/Rx Bytes** — Total bytes transmitted and received.
- **Tx/Rx Packets** — Total number of packets transmitted and received.
- **Tx/Rx Drops** — Dropped transmit and receive packets.
- **Tx Queue Drops** — Packets dropped from the transmit queue.
- **Tx/Rx Errors** — Transmit and receive error counts.
- **Byte Graph** — Real-time graph of byte-level traffic.
- **Packet Graph** — Real-time graph of packet-level traffic.

4.4.20 Adding a New VETH Interface

- **Enabled** — Toggle to enable or disable the VETH interface.
- **Comment** — Optional field to describe the interface.

General

- **Name** — The name of the virtual Ethernet interface (e.g., veth1).
- **Type** - Displays the interface type, which is VETH.
- **Actual MTU** - Displays the effective MTU based on configuration (read-only).
- **L2 MTU** — Displays the Layer 2 MTU size.

Addressing

- **Address** — Use the + button to assign one or more IP addresses to the interface.
- **Gateway** — Optional IPv4 gateway to associate with this interface.
- **IPv6 Gateway** — Optional IPv6 gateway for this interface.

Status

- **Last Link Down Time** — Timestamp of the last time the interface went down.
- **Last Link Up Time** — Timestamp of the last time the interface came up.
- **Link Downs** — Number of times the interface has gone down.

Traffic

- **Tx/Rx Rate** — Transmit and receive data rates (bps).
- **Tx/Rx Packet Rate** — Number of packets transmitted and received per second.
- **FP Tx/Rx Rate** — FastPath data rates for transmitted/received traffic.
- **FP Tx/Rx Packet Rate** — Packet rate using FastPath for both directions.
- **Tx/Rx Bytes** — Total bytes transmitted and received.
- **Tx/Rx Packets** — Total number of packets transmitted and received.
- **Tx/Rx Drops** — Dropped transmit and receive packets.
- **Tx Queue Drops** — Packets dropped from the transmit queue.
- **Tx/Rx Errors** — Transmit and receive error counts.
- **Byte Graph** — Real-time graph of byte-level traffic.
- **Packet Graph** — Real-time graph of packet-level traffic.

4.4.21 Adding a New L2TP Ethernet Interface

- **Enabled** — Toggle to enable or disable the L2TP Ethernet interface.
- **Comment** — Optional field to describe the interface.

General

- Name — The name assigned to the interface (e.g., `l2tp-ether1`).
- Type — Displays the interface type, which is `L2TP Ethernet`.
- Actual MTU — Displays the calculated MTU based on configuration (read-only).
- L2 MTU — Layer 2 MTU value (read-only).
- Max MTU — Defines the maximum transmission unit for the interface.
- Connect To — The IP address or hostname of the remote L2TP server.
- MAC Address — Allows manual configuration of the MAC address.
- Unmanaged Mode — When enabled, the interface behaves without centralized control.
- Use IPsec — Enables IPsec encryption for the connection.
- IPsec Secret — Specifies the pre-shared key used when IPsec is enabled.
- L2TP Protocol Version — Version of L2TP protocol to use (e.g., `l2tpv3 udp`).
- Cookie Length — Length of the L2TPv3 cookie value in bytes.
- Digest Hash — Hash function used to verify packet integrity (e.g., `md5`).
- Circuit ID — Identifier for the pseudowire circuit.
- Use L2 Specific Sublayer — Enables L2-specific sublayer processing.
- Allow Fast Path — Enables fast path processing for improved performance.

Status

- **Last Link Down Time** — Timestamp of the last time the interface went down.
- **Last Link Up Time** — Timestamp of the last time the interface came up.
- **Link Downs** — Number of times the interface has gone down.
- **Active Circuit ID** — The currently active circuit ID.

- **Active Cookie Length** — The active L2TPv3 cookie length in bytes.
- **Header Format** — Indicates the header format in use.
- **L2 Sublayer** — Shows if the L2 sublayer is enabled or disabled.
- **Active Remote Session ID** — ID of the remote session currently active.
- **Active Local Session ID** — ID of the local session currently active.
- **Control Connection ID** — Identifier for the control connection.
- **Peer Address** — The IP address of the remote peer.
- **Encoding** — Encoding format used for data transfer.
- **Uptime** — Time since the interface has been active.

Traffic

- **Tx/Rx Rate** — Transmit and receive data rates (bps).
- **Tx/Rx Packet Rate** — Number of packets transmitted and received per second.
- **FP Tx/Rx Rate** — FastPath data rates for transmitted/received traffic.
- **FP Tx/Rx Packet Rate** — Packet rate using FastPath for both directions.
- **Tx/Rx Bytes** — Total bytes transmitted and received.
- **Tx/Rx Packets** — Total number of packets transmitted and received.
- **Tx/Rx Drops** — Dropped transmit and receive packets.
- **Tx Queue Drops** — Packets dropped from the transmit queue.
- **Tx/Rx Errors** — Transmit and receive error counts.
- **Byte Graph** — Real-time graph of byte-level traffic.
- **Packet Graph** — Real-time graph of packet-level traffic.

4.4.22 Adding a New MACVLAN Interface

- **Enabled** — Toggle to enable or disable the MACVLAN interface.
- **Comment** — Optional field to describe the interface.

General

- **Name** — Name assigned to the interface (e.g., `macvlan1`).
- **Type** — Displays the interface type, which is `MACVLAN`.
- **MTU** — Maximum Transmission Unit in bytes. Default is `1500`.
- **Actual MTU** — Displays the effective MTU (read-only).
- **L2 MTU** — Layer 2 MTU (read-only, used for low-level processing).
- **MAC Address** — Optional. MAC address for this virtual interface. Automatically generated if not specified.
- **Interface** — Parent interface (e.g., `ether1`) on which the MACVLAN is based.
- **Mode** — Mode of operation for the MACVLAN interface:
    - `bridge` — Allows multiple MACVLAN interfaces to communicate over the same physical interface.
    - `private` — Disallows communication between MACVLAN interfaces on the same parent.

Status

- **Last Link Down Time** — Timestamp of the last time the interface went down.
- **Last Link Up Time** — Timestamp of the last time the interface came up.
- **Link Downs** — Number of times the interface has gone down.

Traffic

- **Tx/Rx Rate** — Transmit and receive rates in bits per second.
- **Tx/Rx Packet Rate** — Number of packets sent and received per second.
- **FP Tx/Rx Rate** — FastPath transmit and receive rates.
- **FP Tx/Rx Packet Rate** — FastPath transmit and receive packet rates.
- **Tx/Rx Bytes** — Total number of bytes transmitted and received.
- **Tx/Rx Packets** — Total number of packets transmitted and received.
- **Tx/Rx Drops** — Number of dropped packets for transmit and receive.
- **Tx Queue Drops** — Number of transmit queue drops.
- **Tx/Rx Errors** — Number of transmit and receive errors.
- **Byte Graph** — Visual graph of transmitted and received bytes over time.
- **Packet Graph** — Visual graph of transmitted and received packets over time.

4.4.23 Adding a New PPTP Server Binding Interface

- **Enabled** — Toggle to enable or disable the PPTP server binding interface.
- **Comment** — Optional field to describe the interface.

General

- **Name** — The name assigned to the interface (e.g., `pptp-in1`).
- **Type** — Displays the interface type, which is `PPTP Server Binding`.
- **Actual MTU** — Shows the calculated MTU based on current settings (read-only).
- **User** — Specifies the user account associated with this binding.

Status

- **Last Link Down Time** — Timestamp of the last time the interface was disconnected.
- **Last Link Up Time** — Timestamp of the last successful link connection.
- **Link Downs** — The number of times the interface has disconnected.
- **Uptime** — Duration of the current connection.
- **User** — Displays the user currently connected.
- **Caller ID** — Identifies the calling endpoint.
- **Encoding** — Indicates the encoding method used for the connection.
- **VID** — Virtual ID used for this session.
- **MTU** — Maximum Transmission Unit negotiated for this session.
- **MRU** — Maximum Receive Unit negotiated for this session.
- **Local Address** — Local IP address assigned for this connection.
- **Remote Address** — IP address of the remote peer.
- **Local IPv6 Address** — IPv6 address assigned locally.
- **Remote IPv6 Address** — IPv6 address of the remote device.

Traffic

- **Tx/Rx Rate** — Current transmit and receive bandwidth rates (bps).
- **Tx/Rx Packet Rate** — Transmit and receive packets per second.
- **FP Tx/Rx Rate** — FastPath bandwidth rates in both directions.
- **FP Tx/Rx Packet Rate** — Packet rate under FastPath mode.
- **Tx/Rx Bytes** — Total bytes transmitted and received.
- **Tx/Rx Packets** — Total packets sent and received.

- **Tx/Rx Drops** — Packets dropped during transmission or reception.
- **Tx Queue Drops** — Packets dropped from the transmit queue.
- **Tx/Rx Errors** — Errors encountered during transmit/receive operations.
- **Byte Graph** — Real-time graphical representation of bandwidth usage.
- **Packet Graph** — Real-time graphical representation of packet activity.

4.4.24 Adding a New PPTP Client Interface

- **Enabled** — Toggle to enable or disable the PPTP Client interface.
- **Comment** — Optional field to describe the interface.

General

- **Name** — The name of the PPTP client interface (e.g., `pptp-out1`).
- **Type** — Displays the interface type, which is `PPTP Client`.
- **Actual MTU** — Calculated MTU based on settings (read-only).
- **Max MTU** — Sets the maximum transmission unit size (default: `1450`).
- **Max MRU** — Sets the maximum receive unit size (default: `1450`).
- **MRRU** — Maximum packet size that can be received on the link. If a packet is bigger than tunnel MTU, it will be split into multiple packets, allowing full size IP or Ethernet packets to be sent over the tunnel.

Dial Out

- **Connect To** — IP address or hostname of the PPTP server.
- **User** — Username used for authentication.
- **Password** — Password used for authentication.
- **Profile** — The PPP profile to use.
  - `default` — Standard profile with typical settings.
  - `default-encryption` — Enables encryption for added security.
- **Keepalive Timeout** — Time in seconds to wait before assuming the link is inactive (default: `60`).
- **Use Peer DNS** — Whether to accept DNS settings from the server.
- **Dial On Demand** — Enables the interface to establish a connection only when traffic is detected.
- **Add Default Route** — Adds a default route using this interface.
- **Default Route Distance** — Distance value for the default route (active only if **Add Default Route** is enabled).
- **Allow** — Supported authentication protocols:
  - `mschap2`
  - `mschap1`
  - `chap`
  - `pap`

Status

- **Last Link Down Time** — Timestamp when the interface last disconnected.
- **Last Link Up Time** — Timestamp when the interface last established a connection.
- **Link Downs** — Number of times the link has gone down.
- **Uptime** — Duration the interface has been up.
- **Encoding** — Encoding method used.
- **VID** — Virtual ID for the session (if applicable).

- **MTU** — Current MTU in use.
- **MRU** — Current MRU in use.
- **Local Address** — IP address assigned to this side of the tunnel.
- **Remote Address** — IP address assigned to the remote peer.
- **Local IPv6 Address** — IPv6 address assigned locally (if any).
- **Remote IPv6 Address** — IPv6 address assigned to the remote peer (if any).

Traffic

- **Tx/Rx Rate** — Current transmit and receive rates (bps).
- **Tx/Rx Packet Rate** — Number of packets sent and received per second.
- **FP Tx/Rx Rate** — FastPath packet transmission/reception rates.
- **FP Tx/Rx Packet Rate** — FastPath packet count per second.
- **Tx/Rx Bytes** — Total bytes sent and received.
- **Tx/Rx Packets** — Total number of packets sent and received.
- **Tx/Rx Drops** — Dropped packets (Tx and Rx).
- **Tx Queue Drops** — Packets dropped in the transmit queue.
- **Tx/Rx Errors** — Number of transmit and receive errors.
- **Byte Graph** — Graph showing real-time byte throughput.
- **Packet Graph** — Graph showing real-time packet throughput.

4.4.25 Adding a New PPoE Server Binding Interface

- **Enabled** — Toggle to enable or disable the interface.
- **Comment** — Optional field to describe the interface.

General

- **Name** — The system-assigned name for the PPPoE Server Binding (e.g., `pppoe-in1`).
- **Type** — Displays the interface type as `PPPoE Server Binding`.
- **Actual MTU** — Displays the Maximum Transmission Unit in use (read-only).
- **User** — Displays the username associated with this PPPoE session.
- **Service** — The PPPoE service name that this binding is associated with.

Status

- **Last Link Down Time** — Time the link was last disconnected.
- **Last Link Up Time** — Time the link was most recently established.
- **Link Downs** — Number of times the connection has been interrupted.
- **Uptime** — Total duration the connection has remained active.
- **User** — Username of the authenticated session.
- **Caller ID** — Identifier of the connecting client.
- **Encoding** — Method used to encode the session.
- **VID** — VLAN ID used for the connection, if any.
- **MTU** — Maximum Transmission Unit negotiated.
- **MRU** — Maximum Receive Unit value in use.
- **Local Address** — IP address assigned to the local side of the tunnel.
- **Remote Address** — IP address assigned to the remote side.
- **Local IPv6 Address** — IPv6 address assigned locally.
- **Remote IPv6 Address** — IPv6 address assigned to the remote endpoint.

- **Service** — Service name of the binding.
- **Interface** — Physical or virtual interface this binding is using.
- **VLAN ID** — The VLAN tag (if applicable) associated with this connection.

Traffic

- **Tx/Rx Rate** — Current transmit and receive bit rates.
- **Tx/Rx Packet Rate** — Transmit and receive packet rate (packets per second).
- **FP Tx/Rx Rate** — FastPath traffic bit rate.
- **FP Tx/Rx Packet Rate** — FastPath packet rate.
- **Tx/Rx Bytes** — Total bytes sent and received.
- **Tx/Rx Packets** — Total packets sent and received.
- **Tx/Rx Drops** — Number of packets dropped during transmission or reception.
- **Tx Queue Drops** — Packets dropped from the output queue.
- **Tx/Rx Errors** — Transmission or reception errors detected.
- **Byte Graph** — Live graph displaying byte traffic trends.
- **Packet Graph** — Live graph displaying packet traffic trends.

4.4.26 Adding a New PPPoE Client Interface

- **Enabled** — Toggle to enable or disable the PPPoE client interface.
- **Comment** — Optional field to describe the interface.

General

- **Name** — The name assigned to this interface (e.g., `pppoe-out1`).
- **Type** — Displays the interface type, which is `PPPoE Client`.
- **Max MTU** — Sets the maximum transmission unit for the interface.
- **Max MRU** — Sets the maximum receive unit for the interface.
- **MRRU** — Sets the maximum received reassembled unit.
- **Interfaces** — Select the underlying interface for the PPPoE session.

Dial Out

- **Service** — Optional field to specify the PPPoE service name.
- **AC Name** — Access Concentrator name, this may be left blank and the client will connect to any access concentrator on the broadcast domain
- **Host Uniq** — Optional identifier used to match requests and replies.
- **User** — Username for PPPoE authentication.
- **Password** — Password for PPPoE authentication.
- **Profile** — Defines settings for the PPPoE session such as encryption and compression. Options:
  - `default` — Standard profile with typical settings.
  - `default-encryption` — Enables encryption for added security.
- **Keepalive Timeout** — Interval (in seconds) for sending LCP echo requests. Zero disables the check.
- **Dial On Demand** — When enabled, the link will only come up when traffic is detected.
- **Use Peer DNS** — When enabled, the client will use DNS provided by the server.
- **Add Default Route** — When enabled, the interface will install the default route upon successful connection.
- **Default Route Distance** — Distance used for the default route if **Add Default Route** is enabled.
- **Allow** — Checkboxes to select which authentication protocols to allow:

- mschap2 — Microsoft CHAP v2
- mschap1 — Microsoft CHAP v1
- chap — Challenge Handshake Authentication Protocol
- pap — Password Authentication Protocol

Status

- **Last Link Down Time** — Timestamp of the last time the link went down.
- **Last Link Up Time** — Timestamp of the last time the link came up.
- **Link Downs** — Count of link failures.
- **Uptime** — Duration the link has been up.
- **Encoding** — Compression or encryption method used.
- **VID** — VLAN ID used by the session, if applicable.
- **MTU** — Current maximum transmission unit.
- **MRU** — Current maximum receive unit.
- **Local Address** — IP address assigned to the local PPPoE endpoint.
- **Remote Address** — IP address of the remote PPPoE server.
- **Local IPv6 Address** — IPv6 address assigned to the local side (if applicable).
- **Remote IPv6 Address** — IPv6 address assigned to the remote side (if applicable).
- **Active Links** — Number of currently active links.
- **Active Service Name** — Service name currently active in the session.
- **Active AC Name** — Name of the Access Concentrator used.
- **AC MAC Address** — MAC address of the Access Concentrator.

Traffic

- **Tx/Rx Rate** — Current transmit and receive speeds in bits per second.
- **Tx/Rx Packet Rate** — Current packet transmit and receive rates.
- **FP Tx/Rx Rate** — FastPath transmit and receive speeds.
- **FP Tx/Rx Packet Rate** — FastPath packet transmit and receive rates.
- **Tx/Rx Bytes** — Total bytes transmitted and received.
- **Tx/Rx Packets** — Total number of packets transmitted and received.
- **Tx/Rx Drops** — Count of dropped transmit and receive packets.
- **Tx Queue Drops** — Packets dropped from the transmit queue.
- **Tx/Rx Errors** — Total number of transmit and receive errors.
- **Byte Graph** — Live graph of byte throughput.
- **Packet Graph** — Live graph of packet throughput.

4.4.27 Adding a New PPP Server Interface

This section applies to Airolinx 6 units that include a USB Interface such as the Airlinx 6 Din Rail series.

- **Enabled** — Toggle to enable or disable the PPP Server interface.
- **Comment** — Optional field to describe the interface.

General

- **Name** — The name assigned to this interface (e.g., ppp-in1).
- **Type** — Displays the interface type (PPP Server).
- **Actual MTU** — The calculated MTU value based on configuration (read-only).

- **Max MTU** — Sets the maximum transmission unit (e.g., 1500).
- **Max MRU** — Maximum receive unit (e.g., 1500).
- **MRRU** — Maximum Reconstructed Receive Unit. Used for multilink PPP.

Dial In

- **Port** — Specifies the serial port used (e.g., usb1).
- **Data Channel** — The number of the data channel used (default is 0).
- **Modem Init** — Optional field to define modem initialization string.
- **Null Modem** — Toggle to use null modem (direct serial cable).

Authentication

- **Authentication** — Checkboxes to allow the following methods:
  - mschap2
  - mschap1
  - chap
  - pap
- **Profile** — Defines settings for the PPP session such as encryption and compression. Options:
  - default — Standard profile with typical settings.
  - default-encryption — Enables encryption for added security.
- **Ring Count** — The number of rings required before answering the call (e.g., 0).

Status

- **Last Link Down Time** — Timestamp of the last time the interface went down.
- **Last Link Up Time** — Timestamp of the last time the interface came up.
- **Link Downs** — Number of times the interface has gone down.
- **Uptime** — Duration that the interface has been up.
- **User** — Username of the connected client.
- **Caller ID** — Caller identifier if available.
- **Encoding** — Encoding method used.
- **VID** — Virtual ID if used.
- **MTU** — Current MTU value.
- **MRU** — Current MRU value.
- **Local Address** — IP address assigned to the local interface.
- **Remote Address** — IP address assigned to the remote peer.
- **Local IPv6 Address** — Local IPv6 address if configured.
- **Remote IPv6 Address** — Remote IPv6 address if configured.
- **Connect Speed** — Connection speed of the link if available.

Traffic

- **Tx/Rx Rate** — Transmit and receive data rates in bps.
- **Tx/Rx Packet Rate** — Packet transmission rates in p/s.
- **FP Tx/Rx Rate** — Fast Path Tx/Rx rate in bps.
- **FP Tx/Rx Packet Rate** — Fast Path packet rate.
- **Tx/Rx Bytes** — Total transmitted and received bytes.
- **Tx/Rx Packets** — Number of packets transmitted and received.
- **Tx/Rx Drops** — Number of dropped packets.
- **Tx Queue Drops** — Number of packets dropped from Tx queue.

- **Tx/Rx Errors** — Number of transmission and reception errors.
- **Byte Graph** — Real-time graph of Tx and Rx bytes.
- **Packet Graph** — Real-time graph of Tx and Rx packets.

4.4.28 Adding a New PPP Client Interface

This section applies to Airolinx 6 units that include a USB Interface such as the Airlinx 6 Din Rail series.

- **Enabled** — Toggle to enable or disable the PPP client interface.
- **Comment** — Optional field for adding a description or note.

General

- **Name** — Name of the interface (e.g., `ppp-out1`).
- **Type** — Displays the interface type, which is `PPP Client`.
- **Actual MTU** — Read-only field showing the calculated MTU.
- **Port** — Select the USB port the modem is connected to (e.g., `usb1`).
- **APN** — Optional field to specify the Access Point Name.
- **PIN** — Optional field to enter the SIM card PIN code.

PPP

- **User** — Username used to authenticate the connection.
- **Password** — Password associated with the user.
- **Remote Address** — Remote address to connect to, if applicable.
- **Keepalive Timeout** — Timeout in seconds to wait before re-initiating the connection (e.g., `30`).
- **Dial On Demand** — If enabled, the connection initiates only when traffic is detected.
- **Use Peer DNS** — If enabled, the DNS server offered by the remote peer is used.
- **Add Default Route** — If enabled, this interface adds the default route to the routing table.
- **Default Route Distance** — Sets the administrative distance for the default route (e.g., `1`).

Status

- **Last Link Down Time** — Timestamp of the last disconnection.
- **Last Link Up Time** — Timestamp of the last successful connection.
- **Link Downs** — Number of times the link has gone down.
- **Uptime** — Duration the interface has remained connected.
- **Encoding** — Indicates the encoding method used.
- **VID** — VLAN ID used, if applicable.
- **MTU** — Maximum Transmission Unit.
- **MRU** — Maximum Receive Unit.
- **Local Address** — IP address assigned to the local interface.
- **Remote Address** — IP address of the remote endpoint.
- **Local IPv6 Address** — Local IPv6 address, if used.
- **Remote IPv6 Address** — Remote IPv6 address, if used.
- **Connect Speed** — Reported link speed of the connection.

Traffic

- **Tx/Rx Rate** — Transmit and receive data rates (bps).
- **Tx/Rx Packet Rate** — Number of packets transmitted and received per second.

- **FP Tx/Rx Rate** — FastPath transmit and receive rates (bps).
- **FP Tx/Rx Packet Rate** — FastPath packet rates.
- **Tx/Rx Bytes** — Total bytes sent and received.
- **Tx/Rx Packets** — Total packets sent and received.
- **Tx/Rx Drops** — Dropped packets.
- **Tx Queue Drops** — Drops from the transmit queue.
- **Tx/Rx Errors** — Errors during transmission and reception.
- **Byte Graph** — Real-time byte-level traffic graph.
- **Packet Graph** — Real-time packet-level traffic graph.

4.4.29 Adding a New L2TP Server Binding Interface

- **Enabled** — Toggle to enable or disable the interface.
- **Comment** — Optional description or label for the interface.

General

- **Name** — Specifies the interface name (e.g., l2tp-in1).
- **Type** — Displays the interface type: L2TP Server Binding (read-only).
- **Actual MTU** — Shows the dynamically calculated MTU value.
- **User** — Username of the L2TP client associated with this binding.

Status

- **Last Link Down Time** — Timestamp of when the link was last down.
- **Last Link Up Time** — Timestamp of when the link was last established.
- **Link Downs** — Number of times the link has gone down.
- **Uptime** — Duration of the current session.
- **User** — Logged-in user on this interface session.
- **Caller ID** — Identification from the calling side.
- **Encoding** — Encryption or encoding method used.
- **VID** — VLAN ID, if applicable.
- **MTU** — Maximum Transmission Unit for the session.
- **MRU** — Maximum Receive Unit for the session.
- **Local Address** — Local IPv4 address assigned to this interface.
- **Remote Address** — Remote IPv4 address assigned to the peer.
- **Local IPv6 Address** — Local IPv6 address if assigned.
- **Remote IPv6 Address** — Remote IPv6 address if assigned.

Traffic

- **Tx/Rx Rate** — Real-time transmit and receive bandwidth usage (bps).
- **Tx/Rx Packet Rate** — Number of packets transmitted and received per second.
- **FP Tx/Rx Rate** — FastPath transmit/receive data rate.
- **FP Tx/Rx Packet Rate** — FastPath transmit/receive packet rate.
- **Tx/Rx Bytes** — Total bytes sent and received.
- **Tx/Rx Packets** — Total packets sent and received.
- **Tx/Rx Drops** — Number of packets dropped during transmission or reception.
- **Tx Queue Drops** — Packets dropped in the transmission queue.
- **Tx/Rx Errors** — Errors encountered during packet transmission or reception.

- **Byte Graph** — Graph showing byte-level traffic over time.
- **Packet Graph** — Graph showing packet-level traffic over time.

4.4.30 Adding a New L2TP Client Interface

- **Enabled** — Toggle to enable or disable the interface.
- **Comment** — Optional field to add notes or descriptions about the interface.

General

- **Name** — Name assigned to the interface (e.g., l2tp-out1).
- **Type** — Displays the interface type, which is L2TP Client.
- **Max MTU** — Sets the maximum transmission unit. Default: 1450.
- **Max MRU** — Sets the maximum receive unit. Default: 1450.
- **MRRU** — Maximum Reconstructed Receive Unit, used when multilink PPP is enabled.

Dial Out

- **Connect To** — IP address or domain name of the L2TP server to connect to.
- **User** — Username for authentication.
- **Password** — Password for authentication.
- **Profile** — Specifies the PPP profile to use. Options:
  - default — Standard profile with typical settings.
  - default-encryption — Enables encryption for added security.
- **Keepalive Timeout** — Time in seconds before sending keepalive packets. Default: 60.
- **Src. Address** — Optional source address for the L2TP tunnel.
- **Random Source Port** — Enables use of a random source port for outbound connections.
- **Use Peer DNS** — If enabled, uses DNS servers provided by the peer.
- **Use IPsec** — Enables IPsec encryption. When checked, enter a shared secret in the IPsec Secret field.
- **IPsec Secret** — Shared secret used for IPsec encryption (if enabled).
- **Allow Fast Path** — Enables fast path processing if supported.
- **Dial On Demand** — Establishes the connection only when there is traffic.
- **Add Default Route** — Adds a default route through this interface if checked.
- **Default Route Distance** — Sets the route distance for the default route.

Advanced

- **Allow** — Specifies authentication methods allowed:
  - mschap2
  - mschap1
  - chap
  - pap
- **L2TP Protocol Version** — Select the protocol version. Options:
  - l2tpv2
  - l2tpv3 ip
  - l2tpv3 udp

Status

- **Last Link Down Time** — Timestamp of the most recent disconnect.

- **Last Link Up Time** — Timestamp of the most recent successful connection.
- **Link Downs** — Total number of times the interface has gone down.
- **Uptime** — Duration of current session.
- **Encoding, VID, MTU, MRU** — Technical status values provided by the system.
- **Local / Remote Address** — IPv4 address details for tunnel endpoints.
- **Local / Remote IPv6 Address** — IPv6 address details if applicable.

Traffic

- **Tx/Rx Rate** — Transmit and receive data rates.
- **Tx/Rx Packet Rate** — Packets per second for each direction.
- **FP Tx/Rx Rate** — Fast Path traffic rates.
- **FP Tx/Rx Packet Rate** — Fast Path packet rates.
- **Tx/Rx Bytes** — Total transmitted/received bytes.
- **Tx/Rx Packets** — Total packets sent and received.
- **Tx/Rx Drops** — Dropped packets.
- **Tx Queue Drops** — Dropped packets from transmit queue.
- **Tx/Rx Errors** — Transmit and receive error count.
- **Byte Graph** — Visual traffic over time by byte count.
- **Packet Graph** — Visual traffic over time by packet count.

4.4.31 Adding a New OVPN Client Interface

- **Enabled** — Toggle to enable or disable the interface.
- **Comment** — Optional field to describe the interface.

General

- **Name** — Name of the OVPN client interface (e.g., `ovpn-out1`).
- **Type** — Interface type, automatically set to `OVPN Client`.
- **Actual MTU** — Displays the effective MTU after configuration (read-only).
- **MAC Address** — The MAC address associated with this interface.
- **Max MTU** — The maximum transmission unit, typically set to `1500`.

Dial Out

- **Connect To** — IP or domain name of the OpenVPN server.
- **Port** — UDP or TCP port to connect to the server (default: `1194`).
- **Mode** — Network mode used by the client. Select between `ip` or `ethernet`.
- **Protocol** — Transport protocol (`tcp` or `udp`).
- **User** — Username for authentication with the OpenVPN server.
- **Password** — Password field for OpenVPN user credentials.
- **Profile** — Configuration profile.
  - `default` — Standard profile with typical settings.
  - `default-encryption` — Enables encryption for added security.
- **Certificate** — Choose a certificate if required (e.g., `none`).
- **Verify Server Certificate** — Toggle to enable or disable server certificate verification.
- **TLS Version** — Version of TLS to use (e.g., `any`).
- **Auth.** — Authentication algorithm (e.g., `md5`).
- **Cipher** — Encryption cipher (e.g., `aes 128 cbc`).

- **Use Peer DNS** — Toggle to use DNS settings provided by the server.
- **Add Default Route** — If enabled, adds a default route through this interface.
- **Don't Add Pushed Routes (route-nopull)** — Prevents the client from installing routes pushed by the server.

Status

- **Last Link Down Time** — Timestamp of the last time the link went down.
- **Last Link Up Time** — Timestamp of the last time the link came up.
- **Link Downs** — Number of times the link has gone down.
- **Uptime** — Duration the link has been up.
- **Encoding** — Encoding type used on the interface.
- **VID** — Virtual Interface Descriptor.
- **MTU** — MTU in effect.
- **Local Address** — Local IP address (if assigned).
- **Remote Address** — Remote IP address (if assigned).
- **Local IPv6 Address** — IPv6 local address, if applicable.
- **Remote IPv6 Address** — IPv6 remote address, if applicable.

Traffic

- **Tx/Rx Rate** — Transmit/Receive bandwidth usage (bps).
- **Tx/Rx Packet Rate** — Transmit/Receive packets per second.
- **FP Tx/Rx Rate** — FastPath transmit/receive rate.
- **FP Tx/Rx Packet Rate** — FastPath transmit/receive packets per second.
- **Tx/Rx Bytes** — Cumulative bytes sent and received.
- **Tx/Rx Packets** — Number of packets sent and received.
- **Tx/Rx Drops** — Packets dropped during transmission or reception.
- **Tx Queue Drops** — Dropped packets due to queue overflows.
- **Tx/Rx Errors** — Packet errors during transmit or receive.
- **Byte Graph** — Visual graph of byte throughput over time.
- **Packet Graph** — Visual graph of packet count over time.

4.4.32 Adding a New OVPN Server Binding Interface

- **Enabled** — Toggle the interface on or off.
- **Comment** — Optional field to add notes or labels for this interface.

General

- **Name** — Identifier for the interface (e.g., `ovpn-in1`).
- **Type** — Fixed as `OVPN Server Binding`.
- **Actual MTU** — Read-only field showing calculated MTU.
- **User** — Username tied to this server binding interface.

Status

- **Last Link Down Time** — Timestamp of when the interface last went down.
- **Last Link Up Time** — Timestamp of when the interface last became active.
- **Link Downs** — Number of times the interface has gone down.
- **Uptime** — How long the interface has been active.

- **User** — Username associated with this binding session.
- **Caller ID** — Identifier from the remote caller.
- **Encoding** — The encoding method in use.
- **VID** — Virtual ID tag for the session.
- **MTU** — Maximum transmission unit in use.
- **Local Address** — IP address assigned locally.
- **Remote Address** — IP address assigned to the remote endpoint.
- **Local IPv6 Address** — Local IPv6 address if configured.
- **Remote IPv6 Address** — Remote IPv6 address if configured.

Traffic

- **Tx/Rx Rate** — Transmission and reception rate in bits per second.
- **Tx/Rx Packet Rate** — Transmission and reception rate in packets per second.
- **FP Tx/Rx Rate** — FastPath (hardware-accelerated) transmission and reception in bits per second.
- **FP Tx/Rx Packet Rate** — FastPath packet rate for both directions.
- **Tx/Rx Bytes** — Total number of bytes sent and received.
- **Tx/Rx Packets** — Total number of packets sent and received.
- **Tx/Rx Drops** — Dropped packets during transmission or reception.
- **Tx Queue Drops** — Packets dropped in the transmission queue.
- **Tx/Rx Errors** — Transmission or reception errors recorded.
- **Byte Graph** — Visual real-time graph showing byte usage.
- **Packet Graph** — Visual real-time graph showing packet transmission and reception.

4.4.33 Adding a New SSTP Server Binding Interface

Overview

Secure Socket Tunneling Protocol (SSTP) transports a PPP tunnel over a TLS channel. The use of TLS over TCP port 443 allows SSTP to pass through virtually all firewalls and proxy servers.

- **Enabled** — Toggle to enable or disable the SSTP server interface.
- **Comment** — Optional field to describe the interface.

General

- **Name** — The name assigned to the interface (e.g., sstp-in1).
- **Type** — Displays the interface type, which is SSTP Server Binding.
- **Actual MTU** — Read-only field showing the actual calculated MTU.
- **User** — Username associated with this SSTP session.

Status

- **Last Link Down Time** — Timestamp when the interface was last down.
- **Last Link Up Time** — Timestamp when the interface was last up.
- **Link Downs** — Number of times the connection has gone down.
- **Uptime** — Current duration the interface has been up.
- **Caller ID** — ID of the remote caller (if applicable).
- **Encoding** — Encryption algorithm used during the session.
- **VID** — Virtual ID used for session tracking.
- **MTU** — Configured MTU value for the session.

- **MRU** — Configured Maximum Receive Unit.
- **Local Address** — IP address assigned to the local SSTP endpoint.
- **Remote Address** — IP address of the connected client.
- **Local IPv6 Address** — Local IPv6 address if used.
- **Remote IPv6 Address** — IPv6 address of the remote peer if assigned.

Traffic

- **Tx/Rx Rate** — Transmit and receive data rate in bits per second.
- **Tx/Rx Packet Rate** — Packets per second transmitted and received.
- **FP Tx/Rx Rate** — FastPath transmission and reception rates.
- **FP Tx/Rx Packet Rate** — Packet rate under FastPath.
- **Tx/Rx Bytes** — Total bytes transmitted and received.
- **Tx/Rx Packets** — Total number of packets transmitted and received.
- **Tx/Rx Drops** — Count of dropped packets.
- **Tx Queue Drops** — Packets dropped from the transmission queue.
- **Tx/Rx Errors** — Errors encountered during transmission and reception.
- **Byte Graph** — Graph showing real-time data transfer in bytes.
- **Packet Graph** — Graph showing real-time packet transmission and reception.

4.4.34 Adding a New SSTP Client Interface

- **Enabled** — Toggle the SSTP client interface on or off.
- **Comment** — Optional description for the interface.

General

- **Name** — The name of the interface (e.g., `sstp-out1`).
- **Type** — Set to `SSTP Client`.
- **Actual MTU** — Shows the effective MTU (read-only).
- **Max MTU** — Maximum Transmission Unit, default is `1500`.
- **MRRU** — Maximum Received Reconstructed Unit, used with multilink PPP (optional).

Dial Out

- **Connect To** — IP address or domain name of the SSTP server.
- **Port** — Server port (default `443`).
- **Proxy** — Optional proxy configuration.
- **Proxy Port** — Port for the proxy, default `443`.
- **Certificate** — SSL certificate to use (if any).
- **TLS Version** — Version of TLS to use.
  - `any`
  - `only-1.2`
- **Verify Server Certificate** — Toggle whether to validate the server's certificate.
- **Verify Server Address From Certificate** — Ensures the server address matches the certificate CN.
- **Add SNI** — Sends the server name indication (SNI) extension with TLS handshake.
- **PFS** — Specifies which TLS authentication to use. With pfs=yes, TLS will use ECDHE-RSA- and DHE-RSA-. For maximum security setting pfs=required will use only ECDHE.
  - `yes`
  - `no`

    - required
- **Ciphers** — Select available ciphers such as AES256-SHA and AES256-GCM-SHA384.
- **User** — Username for PPP authentication.
- **Password** — Password for PPP authentication.
- **Profile** — Configuration profile.
    - default — Standard profile with typical settings.
    - default-encryption — Enables encryption for added security.
- **Keepalive Timeout** — Time in seconds before considering the peer disconnected (default 60).
- **Dial On Demand** — Enables interface only when needed.
- **Add Default Route** — Automatically adds default route to routing table.
- **Default Route Distance** — Distance for default route (default 1).
- **Allow** — Authentication protocols allowed: mschap2, mschap1, chap, pap.

Status

- **Last Link Down Time** — Timestamp of the last disconnection.
- **Last Link Up Time** — Timestamp of the last successful connection.
- **Link Downs** — Number of times the interface has gone down.
- **Uptime** — Duration the interface has been up.
- **Encoding** — Encryption method used.
- **VID** — Virtual ID (if assigned).
- **MTU** — Active MTU value.
- **MRU** — Active MRU value.
- **Local Address** — Local IP address assigned.
- **Remote Address** — IP address of the connected SSTP server.
- **Local IPv6 Address** — Local IPv6 address assigned.
- **Remote IPv6 Address** — Remote IPv6 address assigned.

Traffic

- **Tx/Rx Rate** — Transmit and receive rates in bps.
- **Tx/Rx Packet Rate** — Packet transmit/receive rate in packets per second.
- **FP Tx/Rx Rate** — FastPath transmission rate.
- **FP Tx/Rx Packet Rate** — FastPath packet rate.
- **Tx/Rx Bytes** — Total bytes transmitted/received.
- **Tx/Rx Packets** — Total packets transmitted/received.
- **Tx/Rx Drops** — Dropped packets.
- **Tx Queue Drops** — Queue overflow drops.
- **Tx/Rx Errors** — Transmission/reception errors.
- **Byte Graph** — Real-time byte-level traffic graph.
- **Packet Graph** — Real-time packet-level traffic graph.

4.5 Interface List

Summary

The Interface List allows defining a set of interfaces for easier management in different interface-based configuration sections such as Neighbor Discovery, Firewall, Bridge, and Internet Detect.

Lists

This menu contains information about all interface lists available on the router. There are four predefined lists:

- **all** — Contains all interfaces
- **none** — Contains no interfaces
- **dynamic** — Contains interfaces with the dynamic flag
- **static** — Contains interfaces that are not dynamic

It is also possible to create additional custom interface lists.

Tip: Dynamic interfaces are interfaces that have a dynamic flag. Any interface that does not have the dynamic flag will be part of the static interface list.

To Create a New List

1. Click on the **Interface List** tab (under the **Interfaces** section in the left navigation menu).
2. Click on **Lists** in the "Actions" section on the right side.
3. Click **New** to create a new list.
4. Enter the desired list name.
5. Enter the default lists to include/exclude. Valid entries:
    - all
    - dynamic
    - none
    - static
6. Click **OK** to save the list.
7. Click **Close** (under Actions) to return to the main Interface List tab.

To Add Interfaces to the List

1. Click **New** from the main Interface List tab (the list must already be created).
2. Select the desired list from the drop-down.
3. Select the interface from the drop-down that you want to add to the list.

4.6 Ethernet

Overview

The **Ethernet** tab under the Interfaces section displays all physical Ethernet interfaces on the device. By double-clicking an interface, users can view and modify detailed configuration options such as auto-negotiation, loop protection, and traffic statistics. The Ethernet interface can also be configured directly from the Interface tab which list all interfaces.

Settings

- **Enabled** — Enables or disables the Ethernet interface.
- **Comment** — Optional field to describe the interface.

General

- **Name** — Name of the interface (e.g., ether1).
- **Default Name** — System-assigned name.
- **Type** — Displays the interface type (e.g., Ethernet).

- **MTU** — Maximum Transmission Unit (default: 1500).
- **Actual MTU** — The operational MTU.
- **L2MTU** — Layer 2 MTU, typically higher than the regular MTU.
- **Max L2MTU** — Maximum Layer 2 MTU supported by the interface.
- **VRF** — Associated Virtual Routing and Forwarding instance.
- **MAC Address** — Unique identifier for the Ethernet interface.
- **ARP** — Address Resolution Protocol mode (e.g., enabled).
- **ARP Timeout** — Time to keep ARP entries (if applicable).

Ethernet

- **Tx/Rx Flow Control** — Controls transmission and reception pause frames.
- **Auto Negotiation** — Enables or disables automatic negotiation of link settings.
- **Advertising** — Manually specify which link modes to advertise (e.g., 10M/100M/1G Full/Half Duplex).

Loop Protect

- **Loop Protect** — Enables loop detection to prevent Ethernet loops.
- **Send Interval** — Interval at which loop detection packets are sent.
- **Disable Time** — How long the port is disabled if a loop is detected.
- **Status** — Shows whether a loop was detected or not.

Overall Stats
Provides a summary of packet and byte counts segmented by packet size ranges.

Rx Stats
Displays receive-specific error counters such as alignment, CRC, and length errors.

Tx Stats
Displays transmit-specific counters including broadcasts, pauses, and collisions.

Status

- **Last Link Down/Up Time** — Timestamps of the last interface state changes.
- **Link Downs** — Count of how many times the link has gone down.
- **Auto Negotiation** — Whether the interface auto-negotiated link settings.
- **Rate** — Negotiated link speed.
- **Full Duplex** — Indicates if the link is operating in full-duplex mode.
- **Advertising / Partner Advertising** — Shows what modes are being offered and received.

Traffic

- **Tx/Rx Rate** — Current transmit and receive data rates.
- **Tx/Rx Packet Rate** — Current packet-per-second rate.
- **FP Tx/Rx Rate** — FastPath transmit and receive rate.
- **FP Tx/Rx Packet Rate** — FastPath packet rate.
- **Tx/Rx Bytes** — Total bytes sent and received.
- **Tx/Rx Packets** — Total number of packets sent and received.
- **Tx Queue Drops / Errors** — Counters for dropped and errored packets.
- **Graphs** — Real-time traffic graphs for bytes and packets.

4.7 EoIP Tunnel

Ethernet over IP (EoIP) is a tunneling protocol that enables the creation of a Layer 2 Ethernet link between two devices across an IP network. It encapsulates Ethernet frames within IP packets, allowing seamless extension of Ethernet networks over routed infrastructures.

The **EoIP Tunnel** tab under the Interfaces section displays all configured EoIP Tunnel Interfaces on the device. The Ethernet interface can also be configured directly from the Interface tab, which lists all interfaces. Adding a new EoIP Tunnel interface can be achieved on both the Interface Tab and the EoIP Tunnel Tab.

When bridging is enabled at both tunnel endpoints, the EoIP tunnel can transparently forward Ethernet traffic, effectively simulating a direct Ethernet cable between geographically separated devices.

Note: EoIP adds approximately 42 bytes of overhead (8 bytes GRE, 14 bytes Ethernet, and 20 bytes IP). To avoid fragmentation, it is recommended to set the MTU to 1500 bytes to accommodate full-size Ethernet frames.

Why Use EoIP Tunnels?
EoIP tunnels allow devices on separate networks to behave as though they are on the same local Ethernet segment. This is especially useful for:

- **Extending LANs** across sites or facilities while maintaining broadcast and multicast compatibility.
- **Bridging VLANs** over routed networks while preserving tagging and isolation.
- **Layer 2 VPNs** to support legacy protocols or broadcast-based applications.
- **Transparent bridging** over wireless or WAN links when Layer 2 continuity is required.

Key Benefits:

- Maintains Layer 2 transparency between endpoints
- Simple to configure and deploy across IP infrastructure
- Supports legacy or non-IP protocols that rely on broadcast or Ethernet-specific behavior

For EoIP Tunnel configuration instructions, click here.

Troubleshooting EoIP Tunnel Issues

- **EoIP tunnel not establishing**: Confirm that both devices are reachable by IP and have matching tunnel IDs. The tunnel will not form if the ID or remote address is incorrect.
- **Bridge not passing traffic**: Ensure the EoIP interface is added to the correct bridge on both sides. Also verify that STP or other bridge filtering features are not blocking packets.
- **Excessive packet loss or latency**: Check the underlying network for instability. EoIP is sensitive to high latency and dropped packets, especially over long-distance or Internet-based links.
- **Looping behavior on the network**: Avoid bridging multiple EoIP tunnels in a way that creates loops. Use Spanning Tree Protocol (STP) where appropriate to prevent broadcast storms.
- **MTU mismatch or fragmentation**: EoIP adds ~42 bytes of overhead. Lower the MTU on EoIP interfaces (e.g., 1458) to avoid issues, especially when bridging full-size Ethernet frames.

- **Intermittent connection drops**: If tunnel reliability is poor, consider switching to a more robust VPN-based tunnel option that offers encryption and better error handling.

4.8 IP Tunnel

An IP Tunnel is a simple Layer 3 tunneling interface that encapsulates IP packets within another IP packet, allowing two endpoints to establish a virtual point-to-point connection over an IP-based infrastructure. This method enables the transparent transport of IP traffic between geographically separate networks.

The IP Tunnel tab under the Interfaces section displays all configured IP Tunnel interfaces on the device. A new tunnel can be created from either the main Interfaces tab or directly within the IP Tunnel tab.

IP Tunnels, also referred to as **IPIP** tunnels, are defined in RFC 2003 and are supported by many routing devices.

When deployed, the IP Tunnel interface acts as a virtual interface listed under the device's interfaces. It can be used in routing, firewall, or VPN applications where a logical path over an IP network is required.

Note: IP Tunnel encapsulation adds approximately 20 bytes of overhead to each packet. Ensure that MTU settings on intermediate devices account for this to avoid fragmentation issues.

Why Use IP Tunnels?

- To connect separate intranet sites securely across public infrastructure.
- To route IP traffic through intermediate IP networks without reconfiguring source or destination addressing.
- To create test or isolated networks across existing IP backbone infrastructure.

Key Benefits

- Lightweight and standards-based encapsulation (RFC 2003 compliant).
- Compatible with nearly all routing platforms and minimal configuration overhead.
- Can be used as a foundational technology in VPN setups, GRE, or advanced routing topologies.

For IP Tunnel configuration instructions, click here.

Troubleshooting IP Tunnel Issues

- **No traffic passing through the tunnel**: Ensure both endpoints are configured with correct local and remote IP addresses. Double-check firewall settings and routing rules that may block tunnel traffic.
- **Tunnel interface not coming up**: Verify that the remote endpoint is reachable via ICMP (ping). If not, resolve any connectivity issues first.
- **MTU-related packet drops**: IPIP tunnels add 20 bytes of overhead. Reduce the MTU on tunnel interfaces (e.g., set to 1480) to prevent fragmentation or dropped packets.
- **Asymmetric routing behavior**: Make sure return routes are correctly defined on both ends. Missing or incorrect routes may cause unidirectional communication.

- **Performance issues**: Avoid running tunnels over unstable WAN links. High latency, jitter, or packet loss can significantly affect tunnel performance.

4.9 GRE Tunnel

GRE (Generic Routing Encapsulation) is a protocol that enables the encapsulation of a wide range of network layer protocols into point-to-point connections across IP networks. It creates a virtual tunnel between devices, allowing them to communicate as if they were directly connected—even across different networks.

The GRE Tunnel tab under the Interfaces section displays all configured GRE Tunnel interfaces on the device. A new tunnel can be created from either the main Interfaces tab or directly within the GRE Tunnel tab.

Although GRE tunnels are similar to other stateless tunneling protocols like IPIP and EoIP, GRE offers more flexibility. One known issue with stateless tunnels is that if the remote peer becomes unreachable, traffic continues to be routed through the tunnel, resulting in dropped packets. To address this, Airolinx 6 supports a keepalive feature on GRE tunnels to monitor peer availability and mitigate blackhole routing.

Why Use GRE Tunnels?
GRE tunnels are commonly used to establish VPN-like connections across the Internet, interconnect remote networks, or encapsulate non-IP protocols. They're useful when a secure or dynamic tunnel is not required, but you still need a flexible and lightweight solution for packet forwarding between sites.

Key Benefits

- Encapsulates various network layer protocols, not just IP
- Supports dynamic routing protocols over the tunnel
- Simple to configure and lightweight
- Compatible with most modern networking equipment
- Allows tunnel creation even behind NAT when combined with keepalive

Note: GRE adds 24 bytes of overhead (4 bytes GRE header + 20 bytes IP header). GRE can encapsulate only IP and IPv6 traffic (EtherType 0x0800 and 0x86DD). Avoid using the "Check Gateway" option set to "ARP" on GRE tunnel interfaces, as GRE does not support ARP.

For GRE Tunnel configuration instructions, click here.

Troubleshooting Tips:

- Ensure that firewall rules on intermediate devices do not block GRE protocol traffic (protocol 47).
- Check for correct local and remote IP addresses; GRE tunnels will not establish if either endpoint is unreachable.
- If routing over the GRE tunnel fails, verify that no MTU mismatches are causing fragmentation issues.
- Use the keepalive feature to prevent routing loops or traffic blackholing when the remote peer is down.

- Review tunnel interface status under the Interface tab to confirm the link is active and passing traffic.

4.10 VLAN

Virtual Local Area Networks (VLANs) are a Layer 2 technology that allows multiple logically separated networks to coexist on a single physical interface. VLANs enable traffic segmentation by tagging Ethernet frames with identifiers that distinguish them from other network traffic.

The VLAN tab under the Interfaces section displays all VLAN interfaces configured on the device. VLAN interfaces can be created directly in this tab or from the main Interfaces tab. VLANs are supported on various interface types including Ethernet and wireless.

The most widely used VLAN encapsulation standard is IEEE 802.1Q, which inserts a 4-byte VLAN header into Ethernet frames. Each VLAN is treated as a unique broadcast domain, enabling isolated communication. Inter-VLAN communication typically requires a Layer 3 device, such as a router — and since the Airolinx 6 includes Layer 3 routing capabilities, it can route between VLANs directly.

VLAN interfaces can be used for segmenting traffic, enhancing security, and managing bandwidth more effectively. Trunk links between switches or routers carry multiple VLANs simultaneously by tagging traffic with appropriate VLAN IDs.

Note: The IEEE 802.1Q standard reserves certain VLAN IDs for special purposes. Avoid using the following VLAN IDs in general setups: 0, 1, and 4095.

Why Use VLANs?

- Segment networks for improved traffic control and security
- Support multi-tenant or departmental separation on shared infrastructure
- Optimize performance by limiting broadcast domains
- Enable flexible network design and easier scalability

Key Benefits

- Supports up to 4094 VLAN interfaces per physical port
- Fully compatible with 802.1Q and 802.1ad (Q-in-Q) standards
- Trunking support for multi-VLAN communication between devices
- Works across Ethernet and wireless interfaces

For VLAN interface configuration instructions, click here.

Troubleshooting Tips:

- Ensure VLAN IDs are consistent across devices when trunking between switches.
- If a VLAN interface isn't passing traffic, verify that the parent interface supports VLAN tagging and is enabled.
- When using VLANs over wireless, note that VLAN bridging is limited in station mode due to MAC address transport limitations.

- Avoid using reserved VLAN IDs (0, 1, 4095) in standard configurations.

4.11 VXLAN

Virtual eXtensible Local Area Network (VXLAN) is a network virtualization technology designed to overcome the 4096 VLAN ID limitation found in IEEE 802.1Q. VXLAN expands the identifier space to 24 bits, allowing for over 16 million unique network segments. It creates a Layer 2 overlay network on top of an existing Layer 3 infrastructure using UDP encapsulation.

VXLAN enables the extension of Ethernet networks across routed IP networks, effectively simulating a broadcast domain across geographically separated locations. It is especially useful in cloud, data center, and multi-tenant environments where scalability and segmentation are essential.

Note: VXLAN introduces an overhead of 50 bytes when operating over IPv4 and 70 bytes over IPv6. To prevent fragmentation, ensure that the MTU of the underlying network is adjusted accordingly or limit the size of the encapsulated Ethernet frames.

Each VXLAN segment is identified by a unique 24-bit VXLAN Network Identifier (VNI). Only devices within the same VNI can communicate with each other. VXLAN operates in a 1-to-N configuration, rather than a simple point-to-point tunnel, using VXLAN Tunnel Endpoints (VTEPs) to encapsulate and decapsulate traffic.

Remote VTEPs are statically defined. For unicast traffic, VTEPs can dynamically learn destination IPs in a manner similar to MAC learning on a bridge. For broadcast, multicast, or unknown-unicast traffic, VXLAN uses head-end replication or multicast techniques to forward traffic to all relevant VTEPs in the same segment.

Why Use VXLAN?

- Overcomes the VLAN ID limit of 802.1Q (4096) by offering over 16 million unique identifiers.
- Extends Layer 2 networks across Layer 3 boundaries.
- Ideal for scalable multi-tenant environments and data centers.
- Encapsulation over IP allows easy integration with existing routed networks.

Key Benefits

- Supports both IPv4 and IPv6 transport.
- Scalable segmentation of network traffic.
- Stateless tunneling reduces overhead on control planes.
- Improved flexibility for network design and virtual machine mobility.

For VXLAN configuration instructions, click here.

Troubleshooting VXLAN Interfaces:

- **No communication between VTEPs**: Ensure both endpoints are configured with the same VNI and static remote IPs.

- **Fragmentation or packet loss**: Adjust the MTU to accommodate the 50/70 byte overhead, especially when using IPv6.
- **Broadcast not reaching all VTEPs**: Verify that head-end replication is configured correctly and all VTEPs are included.
- **Unicast drops**: Check that MAC address learning is functioning and that the destination VTEP has been dynamically learned or statically set.
- **Interoperability issues**: Ensure other devices in the VXLAN domain support static VTEP configurations, as dynamic discovery is not supported.

4.12 VRRP

The Virtual Router Redundancy Protocol (VRRP) provides network redundancy by automatically assigning available routers to participate in a virtual router group. This ensures uninterrupted network service in the event of a router failure.

Virtual Router (VR)

- **Virtual MAC Address** — Each VRRP instance uses a unique virtual MAC address, which allows devices to respond to ARP requests consistently even if the active router changes.
- **Owner** — The router with the same IP address as the virtual address becomes the VRRP owner. The owner has priority and assumes control when available.
- **Master** — The currently active router forwarding traffic for the virtual address. Initially, this is the owner or the router with the highest priority.
- **Virtual Address** — A shared IP address that remains consistent regardless of which router is currently master.
- **IPv4 ARP** — The master responds to ARP requests for the virtual IP with the virtual MAC, ensuring traffic reaches the current master.
- **IPv6 ND** — For IPv6, the master handles Neighbor Discovery (ND) for the virtual address, ensuring similar failover behavior.

VRRP State Machine

- **Init State** — Initial state before a router becomes active in the VRRP group. No traffic is processed in this state.
- **Backup State** — A router in backup monitors the master. If the master becomes unreachable, the backup may assume control based on priority and advertisement intervals.
- **Master State** — The active router currently forwarding packets. Only one router should be in this state at any time for a given VRRP group.

Why Use VRRP?

- Provides automatic failover to backup routers
- Minimizes downtime due to router failure
- Simplifies gateway redundancy without complex dynamic routing

Key Benefits

- Improved network availability and resilience
- Seamless transition between primary and backup gateways
- Supports both IPv4 and IPv6 environments

For VRRP interface configuration instructions, click here.

Troubleshooting VRRP

- Ensure all routers in the VRRP group are in the same Layer 2 segment.
- Check that the virtual address is not assigned to multiple routers simultaneously.
- Verify that advertisement intervals and priorities are correctly set—conflicting values may cause flapping.
    - Rapidly switches between two states — typically **up/down** or **available/unavailable** — in a short period of time
- Ensure firewall rules or network policies do not block VRRP multicast advertisements (224.0.0.18 for IPv4).
- For IPv6, verify that Neighbor Discovery settings are properly configured for the virtual address.

4.13 VETH

The Virtual Ethernet (VETH) interface allows the creation of virtual Layer 2 interfaces within the device. These interfaces behave like traditional Ethernet ports but exist entirely in software. They are primarily used in containerized environments and for internal bridging or tunneling scenarios.

VETH interfaces can be linked to specific container instances or network namespaces to facilitate communication between internal virtual elements and the rest of the system. When two VETH interfaces are paired (veth0 and veth1), traffic sent to one appears on the other, enabling internal bridging or isolation between services.

Use Cases

- Linking containers to bridge interfaces or firewalled zones.
- Creating isolated test environments within the same device.
- Internal routing or traffic redirection between virtual functions.

Why Use VETH?

- To support containerized applications or internal logical segmentation.
- To simulate Ethernet behavior in software for testing or tunneling purposes.
- To connect isolated processes within the same device through Layer 2 networking.

Key Benefits

- Provides flexible, software-defined networking capabilities.
- Enables better integration with container management systems or Linux-style namespaces.
- Supports bridge configurations and VLAN tagging like physical interfaces.

For VETH interface configuration instructions, click here.

Troubleshooting Tips:

- VETH interfaces must be created in pairs. If one end is missing, traffic will not pass.
- Ensure bridge or IP configuration is assigned correctly if using VETH in routing scenarios.
- VETH interfaces are purely software-based and do not link to physical NICs.

- Check container bindings if using VETH for container connectivity; ensure interface names match expected formats.

4.14 MACsec

Media Access Control Security (MACsec) is a Layer 2 security protocol that provides point-to-point encryption and authentication for Ethernet frames. It is defined by the IEEE 802.1AE standard and protects data integrity and confidentiality on Ethernet links, including wired LAN and fiber connections.

MACsec operates at the data link layer and encrypts entire Ethernet frames (excluding the destination MAC address, source MAC address, and the EtherType field), making it transparent to higher-layer protocols and applications.

Why Use MACsec?

- Provides low-latency, high-speed encryption at Layer 2.
- Prevents traffic snooping or tampering on local and metro-area networks.
- Secures control plane protocols and management traffic.

Key Benefits

- **Confidentiality** — Encrypts Ethernet payloads to prevent data leaks on the wire.
- **Integrity** — Detects and discards altered frames.
- **Authentication** — Ensures frames are from a trusted source using secure associations and key exchanges.
- **Performance** — Offers line-rate encryption with minimal overhead.

For MACsec interface configuration instructions, click here.

Troubleshooting MACsec

- Ensure both endpoints support and are configured for MACsec with matching key and cipher suite.
- Verify that the Physical Layer (Ethernet) is operational before MACsec negotiation begins.
- Check secure channel configuration — mismatched Secure Association Keys (SAKs) or parameters can block traffic.
- Review system logs for handshake errors or dropped packets due to authentication failure.
- MACsec is not supported over interfaces that use NAT, as it expects point-to-point Layer 2 connectivity.

4.15 MACVLAN

Summary

The **MACVLAN** interface type allows you to create a virtual Ethernet interface with its own MAC address on top of an existing parent interface. This can be useful in scenarios requiring multiple MAC addresses on a single physical port, such as when hosting multiple services or virtual machines behind a NAT firewall or when connecting to ISPs that limit access by MAC address.

Configuration Options

- **Name** (read/write): The logical name of the MACVLAN interface. Naming convention should avoid spaces or special characters.
- **Type** (read-only): Always displays `macvlan` for this interface type.
- **MTU** (read/write): Sets the maximum transmission unit. Must be equal to or smaller than the parent interface's MTU.
- **MAC Address** (read/write): Manually set a MAC address. If left blank, the system will auto-generate one.
- **ARP** (read/write): Controls ARP behavior:
  - `enabled` — Standard ARP
  - `proxy-arp` — Responds on behalf of other devices
  - `reply-only` — Only responds to static ARP entries
  - `disabled` — Disables ARP processing
- **ARP Timeout** (read/write): Duration before ARP entries expire. Typical format is `00:01:00` for 1 minute.
- **Interface** (read/write): Select the parent Ethernet interface on which this MACVLAN is created (e.g., `ether1`).

Use Cases

- Connect to an ISP that assigns services based on MAC address, while maintaining additional interfaces internally.
- Assign multiple IP addresses or services to a single port without using VLANs.
- Run isolated containers or virtual machines with separate MAC identities.

For MACVLAN interface configuration instructions, click here.

Troubleshooting

MACVLAN interface not passing traffic: Ensure the parent interface is enabled and not blocked by a firewall or bridge setting.

Duplicate MAC address error: Assign a unique MAC address to each MACVLAN to avoid conflicts with the parent or other interfaces.

No internet access: Confirm the parent interface has IP routing or NAT rules allowing traffic from the MACVLAN interface.

4.16 Bonding

Summary

The **Bonding** interface allows you to combine multiple Ethernet interfaces into a single logical link. This can increase bandwidth, provide hardware redundancy, or both, depending on the selected bonding mode. Bonding is often used to ensure link resilience in critical network environments such as automation backbones or security system uplinks.

Configuration Overview

To add a bonding interface in the WebGUI:

1. Navigate to **Interfaces** > **Bonding**.

2. Click the **Add** (+) button.
3. Configure the bonding parameters as described below.

Bonding Parameters
For Bonding interface configuration instructions, click here.

Bonding Modes
Each mode affects performance and redundancy differently:

- **balance-rr**: Round-robin balancing. Distributes packets sequentially across all interfaces. Provides load balancing and redundancy, but requires support from the connected switch.
- **active-backup**: One interface is active while others remain in standby. If the active link fails, a standby takes over. Ideal for redundancy without requiring switch configuration.
- **balance-xor**: Transmits based on a hash of MAC addresses. Requires support from the switch. Good for simple load balancing.
- **broadcast**: Sends all packets on all interfaces. Ensures absolute redundancy, but is not efficient. Useful in very specific scenarios (e.g., legacy systems).
- **802.3ad (LACP)**: IEEE standard for link aggregation. Dynamically negotiates the best configuration. Requires LACP support and configuration on the switch.
- **balance-tlb**: Transmit Load Balancing. Dynamically balances outbound traffic based on load, with incoming traffic received on the current active interface. No switch support required.
- **balance-alb**: Adaptive Load Balancing. Includes TLB and also receives load balancing via ARP negotiation. Does not require switch configuration.

Best Practices
- Use 802.3ad for high-throughput links to compatible switches (LACP must be configured on the switch).
- Use active-backup when switch configuration is not possible and redundancy is critical.
- Ensure all slave interfaces operate at the same speed and duplex mode to prevent bonding issues.
- Avoid mixing interface types (e.g., copper and fiber) in the same bond whenever possible.

For bonding interface configuration instructions, click here.

Troubleshooting

Bonded link not working: Verify that the slave interfaces are not used elsewhere and are linked up. Check that the mode matches the switch configuration if applicable.

Performance is lower than expected: Some modes (like active-backup) do not provide bandwidth aggregation. Use 802.3ad with proper hashing for optimal performance.

ARP issues or dropped packets: Check that ARP targets are reachable and adjust ARP intervals to better match network timing.

Mixed link speeds: Ensure all interfaces in the bond operate at the same speed to prevent imbalance or failure.

5 WireGuard

5.1 Introduction

WireGuard is a modern VPN protocol known for its simplicity and high performance. The Airolinx 6 integrates WireGuard as a VPN interface. It allows you to create secure tunnels between the router and other WireGuard peers (routers, servers, or mobile clients) using state-of-the-art cryptography. WireGuard aims to be faster and leaner than traditional VPNs like IPsec, and is well-suited for both site-to-site and remote access VPNs.

WireGuard works by creating a virtual network interface (a tunnel) on each peer. Each interface has a private/public key pair, and each peer's public key is configured on the other side along with allowed IPs. All traffic between peers is encrypted using these keys.

5.2 Configuration Steps

For Wireguard interface configuration instructions, click here.

1. **Add WireGuard Interface:** Go to **Interfaces**, click "+", and select **WireGuard**. Give it a name (e.g., wg1). Upon creation, Airolinx 6 will generate a key pair for this interface automatically (you can also import your own keys).
2. **Configure WireGuard Interface:** Open the WireGuard interface settings:
    - **Listen Port:** The UDP port on which this router will listen for WireGuard packets. By default it might choose 13231, but you can set (e.g., 51820 which is the standard).
    - The **Private Key** is generated and shown (keep it secret). The **Public Key** (read-only field) is derived from it. Copy the public key — you will need to give this to the peers.
    - You typically do not set an IP here — instead, assign IP addresses to the WireGuard interface under **IP > Addresses** (for example, give wg1 an IP like 10.10.10.1/24 and the peer 10.10.10.2/24 in the same subnet).
3. **Add Peer:** Still in the WireGuard interface settings, click on the **Peers** tab and add a peer:
    - **Public Key:** enter the public key of the peer (the other router or client's public key).
    - **Allowed Addresses:** specify which IPs are allowed through this peer. For site-to-site, this is typically the remote subnet or remote WireGuard IP. For a road-warrior client, it could be 0.0.0.0/0 if you want to allow it to reach all networks via this tunnel.
    - **Endpoint (optional):** If the peer has a static endpoint, you can set its IP/port. If left blank, the router will learn the endpoint when the peer connects.
4. **Assign IP and Routes:** As mentioned, set an IP address on the WireGuard interface (e.g., 10.0.0.1/24 on wg1). On the peer side, set a matching IP (10.0.0.2/24 on their wg interface). Add routes if needed: e.g., if this tunnel is for reaching a remote network 192.168.2.0/24 through peer, add a route with gateway = wg interface or the peer's IP.
5. **Firewall:** WireGuard uses UDP. Ensure input firewall rules allow the WireGuard UDP port on the WAN interface. By default, the rule allowing established/related connections should suffice once the handshake is initiated outbound by one side.

5.3 Troubleshooting

Handshake not forming: In **Interfaces > WireGuard > Peers**, there is a "Last Handshake" timestamp. If it stays empty, traffic isn't establishing. Check that the UDP port is open (on both sides) and the

keys/allowed IPs are correct. Remember that at least one side must initiate traffic for the tunnel to start (WireGuard is stateless until traffic flows).

Peer connects but no traffic passes: Verify Allowed Address settings. If you forgot to include the traffic's destination in the Allowed Addresses on either side, that traffic will be dropped. For example, if Router A's peer entry for Router B only allows 10.0.0.2/32 but Router B is trying to send 192.168.2.0/24, it won't pass. Include remote LAN subnets in the Allowed Addresses.

Multiple peers on one WG interface: This is supported; ensure each peer has a unique Allowed Address range (so the router knows which peer's key to use for a given packet's destination). Overlaps can cause confusion and are not allowed.

MTU issues: WireGuard's default MTU is 1420 bytes. If you experience unusual issues like certain websites not loading through the VPN, you might have an MTU problem. You can adjust the **MTU** in the WireGuard interface settings (e.g., to 1380) or enable "Clamp TCP MSS" in IP > Firewall > Mangle for that traffic.

WireGuard status can be observed in real-time. Under the WireGuard interface, each peer entry will show bytes transferred once there is activity. This simplicity (a single interface with peers, no complex phase negotiations) is what makes WireGuard both efficient and easy to troubleshoot compared to IPsec. Always distribute the correct public keys and keep private keys secure; if a key is compromised, simply generate a new key pair for that peer and update the configs.

6 PPP

6.1 Introduction

The **PPP** menu covers Point-to-Point Protocol connections and related services. PPP is a family of protocols used to transport network traffic over point-to-point links. In Airolinx 6, PPP encompasses:

- **PPPoE (PPP over Ethernet):** common for ISP DSL or fiber connections where the router must authenticate (with username/password) to gain internet access.
- **PPTP/L2TP/SSTP:** VPN tunneling protocols for connecting to remote servers or accepting client VPN connections. PPTP and L2TP are older (PPTP is not considered secure by modern standards; L2TP often used with IPsec). SSTP (Secure Socket Tunneling) uses HTTPS for traversal of firewalls.
- **OVPN (OpenVPN):** Airolinx 6 supports OpenVPN in client and server forms (with certain limitations, like TCP mode only in current versions).
- **PPP Serial:** If the device had a serial modem or is using a cellular modem in PPP mode, those connections appear here too.

The PPP section also includes configuration for client authentication (**Secrets**), profiles (setting up default parameters for PPP connections), and active connections.

6.2 Configuration Steps

Example: Setting up a PPPoE Client for WAN

1. Go to **Interfaces > PPP** (or directly **PPP > Interfaces** depending on UI section grouping). Click "+" and select **PPPoE Client**.
2. Enter a name (or leave default). Set the **Interface** (this is the physical interface the PPPoE will run over, e.g., ether1 if that's connected to ISP). Enter the **Username** and **Password** provided by your ISP.
3. For most cases, enable **Add Default Route** (so that once connected, the dynamic default route is added pointing through this PPPoE). Also, ensure **Use Peer DNS** is on if you want to accept DNS servers from ISP.
4. Click "OK" — the PPPoE interface will try to connect. You can see its status as "connecting" then hopefully "connected" with an IP address in the Interface list. At this point, your router's WAN is up through PPPoE.

Example: PPTP/L2TP Client (as a client connecting to a VPN server)

1. Click "+" in PPP and choose **PPTP Client** (or L2TP Client similarly).
2. Enter the **Server** (IP or DNS name of the VPN server) and the account credentials. For L2TP, you might also need to specify if it uses IPSec (in which case you'd set IPsec Secret as well).
3. After adding, enable the client. The status can be checked in the **Status** tab of that interface. If connected, you'll get an IP (often from the server side).
4. If this VPN is to be used for reaching specific networks, add the appropriate routes (e.g., route 10.5.5.0/24 via the PPTP interface). Alternatively, if it's internet-bound (rare for PPTP), set as default route with proper metric.

Example: L2TP/SSTP Server (to accept VPN from clients)

1. In **PPP > Interfaces**, click "L2TP Server" (or SSTP Server) and enable it. For SSTP, you'd need certificates; for L2TP, you often use it with IPsec for security (see IPsec section).

2. Go to **PPP > Secrets**. Add a **Secret** (user account) with Name (username), Password, Service (pick "l2tp" or "sstp" depending on what you enabled), and assign a **Profile**. The profile determines IP addresses (local and remote) given to the client, routes, etc. Set the **Remote Address** in the secret or profile to an IP (from a pool) that the client will get.
3. Clients should now be able to connect with that username/password. Monitor active connections in **PPP > Active Connections**.

6.3 Troubleshooting

PPPoE not connecting: Ensure the physical link is up. Check the **Logs** — PPPoE will log if authentication fails (wrong user/pass) or no reply (possibly no PPPoE server detected — then verify VLAN or interface selection). Some ISPs require a specific service name; if so, set the "Service Name" field to what they specify.

PPTP/L2TP connects but no traffic: Remember to configure **IP > Firewall > NAT** to masquerade traffic exiting via the VPN interface if it's meant to reach the internet. Also, if the VPN client's IP is in a different subnet, ensure you have proper routes and that the other end knows how to reach your LAN (often requiring **PPP Profile** settings for "routes" or adding routes on server).

VPN speed is slow: PPTP and L2TP are single-threaded and not well optimized. If possible, use faster protocols (WireGuard or hardware-accelerated IPsec) for better performance. SSTP can be CPU intensive due to encryption over TCP (and subject to TCP overhead).

Too many concurrent VPN users: Airolinx 6 on this device can handle multiple, but track resource usage in **System > Resources**. If CPU is maxed with many SSTP/OVPN, consider offloading some tasks or use simpler protocols.

PPP settings are quite extensive (especially under Profiles and the individual protocol settings). For instance, each PPP profile can set idle timeout, session uptime limits, address pools, and whether to use RADIUS for authentication. This manual touches the basics; for complex deployments (like ISP with hundreds of PPPoE clients).

7 Bridge

7.1 Introduction

A **Bridge** in Airolinx 6 connects multiple interfaces at Layer2, similar to a network switch. By adding Ethernet ports and wireless interfaces to a bridge, they all become part of the same broadcast domain (same LAN). For example, if you want your Wi-Fi and LAN ports to act as a single network, you bridge them so that devices can communicate transparently.

Bridging is powerful: you can implement spanning tree protocol (STP/RSTP) to prevent loops, apply VLAN filtering for VLAN tagging/untagging on ports, and even use bridges to integrate tunnels with local LANs. By default, many Antaira/Antaira routers come with a bridge named "bridge" that has all LAN ports and the Wi-Fi interface included, with a single IP on that bridge for management.

Key bridge features:

- **STP/RSTP:** Prevents loops in case you accidentally connect ports in a ring. RSTP is enabled on bridges by default.
- **Bridge Ports:** The list of member interfaces in a bridge. Adding a port to a bridge is like plugging that port into a virtual switch.
- **VLAN Filtering:** When enabled, the bridge will handle VLAN tags according to a VLAN table. This allows the router to act as a managed switch, isolating VLANs.
- **Bridge Settings:** Control things like whether the bridge uses IP Firewall (to apply firewall on bridged traffic), etc.

7.2 Configuration Steps

1. **Create a Bridge:** Go to **Bridge** menu, click "+" to add a new bridge. Give it a name (e.g., `bridge1`). For most cases, STP can be left as RSTP (default) unless you have a good reason to change. Click OK.
2. **Add Ports to Bridge:** Switch to the **Ports** tab in the Bridge menu. Click "+". Select the Interface you want to add (e.g., `ether2`) and the Bridge (e.g., `bridge1`) you just created. Click OK. Repeat for other interfaces (e.g., `wifi1`) that should join this bridge.
3. Now those ports are bridged. If you assign an IP address to the bridge (instead of individual ports), that IP will cover all bridged ports. For example, if this is a LAN bridge, put your LAN IP (192.168.X.1/24) on `bridge1`, and all ports in the bridge share that IP for router services.
4. **Enable VLAN Filtering (if needed):** If you plan to manage VLANs on this bridge, set up the VLANs under the **VLANs** tab (define VLAN IDs, which bridge they apply to, and list of tagged/untagged ports) and then turn on the **vlan-filtering=yes** on the bridge. It is often recommended to configure VLAN entries first, then enable filtering to avoid losing access (when enabling VLAN filtering, untagged traffic might be dropped unless VLAN 1 is configured appropriately).
5. **STP Verification:** If you have redundant links, ensure STP is working. In the Bridge menu, the **Bridge** tab will show "root bridge" etc. By default, the bridge with lowest MAC becomes root. You can adjust STP priority if needed.

7.3 Troubleshooting

Can't access router after enabling VLAN filter: This often happens if your management PC is on an untagged VLAN and filtering cut it off. Always make sure the management port is configured in the

bridge's VLAN table. For example, if you manage on VLAN 10, the bridge should have VLAN10 with the CPU (bridge itself) tagged or untagged accordingly so the router will respond on that VLAN. You might need to use the MAC Winbox (Layer2 access) or console to recover if locked out.

Bridged ports not passing traffic: Check the "HL" (hardware offload) flags in the Ports tab. If some say H (hardware) and others not, they might not be hardware switched together. If mixing different media (Ethernet and Wireless), it's fine. If you expected hardware offload and it's not happening, ensure consistent bridge settings. Sometimes certain features (e.g., use IP firewall on bridge) disable hardware offload.

Loop detected, port disabled: If you see in log that a port was put in blocking or disabled state, STP likely detected a loop. This is good—it prevented a network storm. Check cabling to remove the loop or, if intentional (e.g., redundancy), ensure only one is forwarding as per STP. The STP status can be seen under Bridge > Ports (shows roles like designated, root port, etc.).

In summary, bridging is straightforward: create a bridge, add ports. It effectively turns your router into a switch for those ports, while still allowing it to route between that bridge and other interfaces (e.g., between LAN bridge and a WAN port). When you incorporate Wi-Fi into a bridge with LAN, you create a typical access point environment. Airolinx 6's bridging implementation is very powerful, including features like Layer2 firewall (filter rules on the bridge if needed), MAC NAT (not commonly used), and protocol mode switching. However, those are beyond basic usage and only needed in specialized scenarios.

8 Mesh

8.1 Introduction

The **Mesh** menu refers to wireless meshing using Antaira's HWMP+ protocol, which is a Layer2 mesh routing protocol. Mesh networking allows multiple wireless routers to connect to each other dynamically, extending coverage without a central AP. However, with Wi-Fi 6 devices, a different approach (CAPsMAN-based "Home Mesh") is typically used for ease of deployment, as HWMP+ is not supported on the new WiFi interfaces.

HWMP+ Mesh can still be used if you have legacy wireless interfaces (i.e., if a Wi-Fi 5 radio was present or an external wireless interface using the old wireless package). It creates a virtual mesh interface to which you add physical interfaces (wireless or even Ethernet) as mesh ports. The protocol then manages forwarding over those ports, finding optimal paths and avoiding loops, similar to how STP works but using mesh routing.

8.2 Configuration Steps (HWMP+ Mesh)

1. Go to **Mesh** (or **Interfaces > Mesh**). Click "+" to add a Mesh interface. Give it a name (e.g., mesh1). By default, it will have HWMP+ protocol active.
2. After creating the mesh interface, select it and click on the **Ports** tab (similar to Bridge). Click "+" to add ports to this mesh. For instance, choose a wireless interface (that is running in mode "mesh" or "ap-bridge") and add it to the mesh. You could also add an Ethernet if you want mesh over Ethernet as well (mesh is not limited to wireless).
3. Ensure all devices that are supposed to mesh have the mesh interface configured with the same mesh ID (the name, effectively) and that their wireless interfaces are set to a common SSID/channel for meshing. The HWMP+ protocol will automatically form links between those that can hear each other.
4. You can assign an IP address to the mesh interface (for management or for layer3 traffic across the mesh). Alternatively, bridge the mesh interface with other ports if integrating into a larger LAN.

8.3 Troubleshooting & Notes

Mesh not establishing links: All participating nodes should use the same wireless frequency/SSID and authentication (if any). If using encryption, note that HWMP+ doesn't inherently secure links, so you might rely on WPA2 on the mesh AP/Station interfaces or use secure tunnel inside the mesh.

Performance: Mesh networks send more overhead (each node periodically announces itself). Over many hops, throughput will diminish. For Wi-Fi 6, consider using "Home Mesh" mode which essentially sets up a CAPsMAN controller — it might be simpler for home scenarios, as new APs can join by a reset button mechanism.

HWMP+ vs CAPsMAN Mesh: As noted, the dedicated Mesh menu is for HWMP+. If your device is pure Wi-Fi6 (with wifiwave2), you likely will use CAPsMAN-based mesh (which uses CAPsMAN to manage APs and establish a mesh-like network via controller). The Quick Set "Home Mesh" does exactly that behind the scenes. It's recommended for ease-of-use. HWMP+ remains available for advanced custom setups or compatibility with older Antaira devices.

In many deployment cases, especially industrial or campus, a **wireless mesh** can increase resilience (multiple path redundancy). But for small-scale or indoor Wi-Fi, using a central controller (CAPsMAN) and ethernet backhaul or a simple repeater is often enough. Mesh should be used when every node needs wireless backhaul to extend range and may have multiple neighbors (not just a chain).

9 IP

9.1 Introduction

The **IP** menu contains most of the fundamental network configuration for Layer3 (the IP protocol) on the router. This includes IP addressing, ARP, DNS settings, DHCP services, firewall, and various IP-based services. We will go through each sub-menu in the order they appear:

- **ARP** — Address Resolution Protocol settings and ARP table.
- **Addresses** — IP address assignments on interfaces.
- **DHCP Client/Relay/Server** — Components of the DHCP (Dynamic Host Configuration Protocol) system for IPv4.
- **DNS** — DNS cache and static entries.
- **Firewall** — Rules for filtering, NAT (Network Address Translation), mangle, etc.
- **Hotspot** — Integrated hotspot gateway feature for managing captive portal access.
- **IPsec** — IP security policy management (for VPN tunnels, etc.).
- **Kid Control** — A simple parental control feature for scheduling internet access.
- **NAT PMP** — NAT Port Mapping Protocol, for automatic port forwards (like Apple's protocols).
- **Neighbors** — Discovery of other devices (uses Antaira's MNDP or CDP/LLtd depending on settings).
- **Packing** — IP packet packing/aggregation configuration to optimize slow links.
- **Pool** — IP address pools used by DHCP and other services.
- **Routes** — Static routing table (for dynamic routing, see separate Routing menu).
- **SNMP** — SNMP agent settings for network monitoring.
- **SSH** — Secure Shell server configuration for remote CLI access.
- **Services** — Management of IP services (telnet, www, api, winbox, etc. — which ports they listen on).
- **Settings** — Miscellaneous IP settings (like ARP timeouts, etc.).
- **Socks** — SOCKS proxy server settings.
- **TFTP** — Trivial File Transfer Protocol server configuration.
- **Traffic Flow** — NetFlow/IPFIX traffic export settings.
- **UPnP** — Universal Plug and Play settings (for automatic NAT hole punching).
- **VRF** — Virtual Routing and Forwarding instances for multiple routing tables (advanced use). Also ties into MPLS/VPN usage.
- **Web Proxy** — Internal web proxy server for caching and URL filtering.
- **Accounting** — Traffic accounting features (historical usage data per host).

We'll break these down into sub-sections now. Note that many of these features are optional — you may not use them all. For instance, you might use DHCP server on your LAN, configure a few firewall rules, and that's it. Others (Hotspot, Web Proxy, etc.) are only if you need those specific capabilities.

9.2 ARP

The **ARP** (Address Resolution Protocol) submenu shows the router's ARP table — mappings of IP addresses to MAC addresses on networks the router is connected to. Normally, ARP is handled automatically. Each interface can be in different ARP modes (enabled, proxy-arp, etc.):

- On LAN interfaces, ARP is "enabled" by default — the router will learn MACs of devices as they send traffic.

- Proxy ARP: If enabled on an interface, the router will answer ARP queries on behalf of addresses that it can reach via other interfaces. This is used in certain advanced setups (e.g., when you want the router to bridge two subnets logically or for VPN clients to appear as local). It's disabled by default unless needed.
- ARP **table entries**: You can add static ARP entries here (IP to MAC mapping) if you want to lock an IP to a specific MAC for security (or in case of devices that don't do ARP properly).

In normal operation, you don't need to touch ARP settings. If something isn't reachable, checking the ARP table can be a good diagnostic step. For example, if 192.168.1.100 is online but the router can't ping it, check if an ARP entry is present. If not, perhaps that device is not responding to ARP — or if it's present but wrong MAC, there might be an IP conflict.

9.3 Address

All IP addresses configured on the router's interfaces are managed here. Typically, you will assign at least one IP for LAN and possibly set IPs for VLAN interfaces or loopback interfaces for management.

To add an IP: click "+", enter the address in CIDR notation (e.g., 192.168.10.1/24) and select the interface (e.g., bridge1 or ether2). Click OK. The IP will become active (blue D flag indicates dynamic if from DHCP, etc., but static ones you add have no flags).

You can add multiple IPs on an interface if needed (alias IPs). Just be cautious with overlapping subnets or unintended duplicates.

9.4 DHCP Client

This is where you set an interface to obtain an IP configuration dynamically from a DHCP server (likely your ISP or an upstream router). Many routers have a DHCP Client enabled by default on their WAN port. In this menu, you'll see any active DHCP client.

Configuration involves selecting the interface and then options like "Use Peer DNS" and "Add Default Route". If your WAN is a simple DHCP from ISP, ensure these are enabled so the router automatically gets DNS servers and default route from the ISP.

An active DHCP client will show status: bound and the IP, gateway, DNS info. If it says searching... then it's not getting a response (cable unplugged? No server?).

9.5 DHCP Relay

DHCP Relay forwards DHCP requests from one network to a DHCP server on another network. Use case: You have a central DHCP server and the router should relay requests from a different subnet to that server.

To configure, specify the interface where requests come from and the IP of the DHCP server to forward to. The router will then listen and forward accordingly.

9.6 DHCP Server

The DHCP Server provides IPs to clients on a local network. If you ran the default setup, your router likely already has a DHCP server running on your LAN bridge.

Key items in DHCP Server config:

- **Networks:** Defines settings per subnet (like gateway, DNS, NTP to distribute).
- **IP Pools:** The range of addresses to give out (see IP > Pool). Ensure your pool matches your LAN subnet and doesn't include the router's own IP.
- **Leases:** The list of active (and past) leases, showing which MAC got which IP and for how long. You can make a lease static (bind MAC to IP) by clicking "Make Static".

To set up a new DHCP server (if not already present):

1. Create an IP pool under IP > Pool (e.g., name "LANpool", addresses 192.168.5.100-192.168.5.200).
2. Go to IP > DHCP Server, click DHCP Setup. It will guide you through selecting interface (e.g., bridge1), subnet, pool (select the one created), gateway (it will suggest the router's IP), and DNS (you can use router's IP or external DNS), and lease time.
3. Finish the wizard — it automatically creates the server, network, etc. Now clients on that interface should get IPs.

9.7 DNS

Airolinx 6 includes a DNS cache which can serve client DNS requests and perform lookups. In IP > DNS, you can:

- Enable **Allow Remote Requests** if you want the router to answer DNS queries from your LAN devices. If enabled and your DHCP server hands out the router's IP as DNS, then the router will resolve for clients (caching results).
- Set upstream DNS servers (if not taken from DHCP client automatically). You can list multiple (e.g., 8.8.8.8, 1.1.1.1 or your ISP's).
- Add static DNS entries (e.g., `server.localnet` -> 192.168.1.50) under Static section. Useful for local hostnames not in public DNS.

When troubleshooting, IP > DNS > Cache will show cached entries. You might flush the cache if needed (rarely necessary).

9.8 Firewall

The Firewall in Airolinx 6 is very powerful. It includes multiple chains:

- **Filter Rules:** For permitting or blocking traffic (in `input` chain for traffic to the router itself, `forward` chain for traffic through the router, `output` for traffic from the router).
- **NAT Rules:** For source NAT (masquerading) and destination NAT (port forwarding).
- **Mangle:** For marking packets, connection marking, or altering certain properties — typically used in advanced QoS or policy routing.
- **RAW:** A raw table to bypass connection tracking for certain traffic (advanced usage for performance or special filtering).

- **Address Lists:** Groups of IPs that can be referenced in rules (e.g., list of "blocked_ips").
- **Layer7 Protocols:** Regex-based patterns for matching traffic by content (not commonly used now due to HTTPS everywhere).

A typical basic firewall (which default config provides) does:

- Allow established and related traffic (so return traffic for connections you initiated is allowed) in forward and input.
- Drop invalid packets.
- Drop all input from WAN that is not established/related (i.e., new connections to router from internet are blocked, except maybe ping).
- Masquerade out the WAN interface for NAT.

You can tailor rules as needed. For example, to port-forward (dst-nat) port 80 to an internal server 192.168.1.100: add a NAT rule in chain=dstnat, dst-port=80, action=dst-nat to 192.168.1.100:80 (and ensure a firewall filter rule allows that forwarded traffic in forward chain, though established/related usually covers it if connection tracking sees it).

Be careful with firewall — an incorrect rule could lock you out or disrupt traffic. Use safe mode or out-of-band access when experimenting.

9.9 Hotspot

The Hotspot feature allows you to set up a captive portal — users connecting via the hotspot interface are redirected to a login page. It's often used in guest Wi-Fi or cafes to provide a Terms of Service or require login vouchers.

Airolinx 6's hotspot system includes user management, walled garden (allow some sites before login), trial mode, etc. It can be configured via the Hotspot Setup wizard, which will create a hotspot server on an interface (e.g., guest bridge), set up an IP pool for guests, and provide default login pages.

Once set up, any HTTP request from a new client triggers redirect to the router's hotspot page. Users can log in with a universal password, individual users, or just click through (depending on config). After authentication, the firewall opens their access.

This is a specialized feature — if you need it, follow the wizard and then fine-tune in IP > Hotspot (where you have tabs for Users, Walled-garden, etc.). The manual's scope is limited here since hotspot has many options (RADIUS integration, data limits, etc.). If not using, simply ignore this menu.

9.10 IPsec

IPsec provides security (encryption and authentication) for IP traffic. In Airolinx 6, IPsec is used both for site-to-site VPNs and to secure other tunnels (like L2TP/IPsec for remote access VPN). The IPsec menu is where you configure Proposals (encryption algorithms), Peers (remote IP and authentication), Identities (credentials like pre-shared key or certificates), and Policies (which traffic to encrypt).

A simple site-to-site IPsec example: two sites want to encrypt traffic between 192.168.1.0/24 and 192.168.2.0/24. You'd set up matching Peers (each pointing to the other's WAN IP, with a pre-shared-key),

and Policies that say source=192.168.1.0/24 dest=192.168.2.0/24 uses IPsec (and the reverse policy on the other). Once configured, IPsec will establish (you'll see Security Associations under Active Peers) and the traffic is encrypted.

Wi-Fi 6 Device OS also supports WireGuard (which we covered separately) and still supports older IPsec with IKEv2, etc. For remote-access, a popular method is L2TP over IPsec (which can be set up partly here and partly under PPP). You may not need to manually edit IPsec if using the L2TP/IPsec built-in support (as the profile can auto generate policies).

IPsec is an advanced area; ensure time is synced (use NTP) especially if using certificates. The **Firewall** should allow UDP 500, 4500 and IP protocol 50/ESP if NAT traversal or AH/ESP are used.

9.11 Kid Control

Kid Control is a simple feature to schedule internet access for certain devices. In this menu, you can add "kids" and assign their devices (by MAC or IP). Then you set allowed times (like internet allowed from 7:00 to 20:00) and possibly a daily time quota.

When outside allowed times, the router will block traffic from those devices (likely by dynamically adding firewall rules or address-list entries). It's an easy way to enforce bedtime for Wi-Fi usage without manually writing firewall rules.

It's straightforward: add entry, select targets (e.g., child's phone MAC or IP), set schedule. You can pause or disable as needed. Under the hood, this uses firewall to drop packets when disallowed.

9.12 Media (DLNA)

The Media feature in Airolinx 6 is a DLNA compliant media server. If your router has a USB port and you attach storage (like a flash drive with videos or music), enabling the Media server will allow Smart TVs or computers on the LAN to discover and stream those media files via DLNA/UPnP.

Configuration (IP > Media):

- **Enable** the media server.
- Add a media **Share** (with "Add"): specify an **interface** (the network on which to advertise, e.g., bridge1), a **path** (the directory on the router's disk or USB to share), and an optional friendly name. E.g., path=disk1/Movies, name "MyRouterMedia".
- Once added, any DLNA client can see "MyRouterMedia" and browse the files. You can restrict access by allowed IPs or hostnames if desired (for example, only let certain devices use it).

Keep in mind performance: the router's CPU and USB speed will limit streaming performance. But for occasional use it's a nifty feature (basically turning the router into a mini-NAS for media).

9.13 NAT PMP

NAT Port Mapping Protocol (NAT-PMP) is similar to UPnP, used by devices (like some Apple products or peer-to-peer apps) to request port mappings on the router automatically. Airolinx 6 implemented NAT-PMP in recent versions. Under IP > NAT-PMP, you can simply enable it and specify which interfaces are "internal" (LAN) and which is "external" (WAN).

Once enabled, devices on internal interfaces can send NAT-PMP requests to create port forwards. The router will then dynamically create corresponding DST-NAT and filter rules (visible marked as dynamic and labeled nat-pmp in firewall). This serves similar purpose to UPnP IGD.

For security, ensure you only add trusted internal interfaces as type=internal. NAT-PMP does not have authentication — any host on internal can open a port. If that's a concern, do not enable it or use firewall to restrict which internal IPs can use it.

9.14 Neighbors

The Neighbors menu shows other Antaira routers or supported devices on directly connected networks via the Antaira Neighbor Discovery Protocol (MNDP) or CDP/LLDP. This can be useful to find devices IPs or MACs if you don't know them. The list will show MAC, identity, IP, platform of discovered devices.

Under IP > Neighbors > Discovery Settings, you can toggle which interfaces will perform discovery. For example, you usually want discovery on LAN, but not on WAN (no need to advertise your device on the internet). By default, only LAN is enabled.

It's mostly an informational tool; double-clicking a neighbor in Winbox can allow you to connect to it by MAC (if Winbox) or open WebFig if it responds with an IP.

9.15 Packing

IP Packing aggregates small packets into larger ones to improve efficiency on slow or high-latency links. It can also compress them. This is an advanced feature seldom needed on modern broadband, but might help on, say, low bandwidth radio links. Both ends need to support it (Airolinx 6 devices).

To use it: enable Packing on an interface at both ends. Set mode to "simple" (aggregate without compression) or "compress-headers" or "compress-all" to compress data. There's also an **IP neighbor discovery** requirement — the discovery protocol should be on for that interface for packing coordination.

You'll see counters of how many packets are aggregated. If not needed, leave it off as it adds some CPU overhead. It's off by default.

9.16 Pool

IP Pools define ranges of IP addresses for dynamic assignment. DHCP Server, PPP, and other services use pools. For example, a pool "dhcp_pool1" might be 192.168.10.100-192.168.10.199. The DHCP server then draws from this pool for leases.

You can have multiple pools and even chain them (a pool can have a "next pool" if it runs out of addresses.

Ensure pools do not include the router's own address or any statically assigned addresses on that subnet. The DHCP setup wizard usually handles this by excluding network and gateway addresses.

9.17 Routes

The routing table lists how the router forwards packets. Each entry has a destination prefix (like 0.0.0.0/0 for default route, or 192.168.5.0/24 for a LAN), a gateway (next hop IP or interface), distance (priority), etc. A connected network (one directly on an interface) appears as a connected (C) route automatically. For instance, adding IP 192.168.5.1/24 on ether3 creates a route 192.168.5.0/24 reachable on ether3 (C).

To add a static route, go to IP > Routes, click "+". Enter Destination (e.g., 10.10.0.0/16) and Gateway (either an IP or select an interface for special cases like reachable via that interface). Typically you put the IP of the next router as gateway. Distance can stay 1 if it's primary.

The default route is 0.0.0.0/0 with gateway = your ISP's gateway (if static) or dynamically added by DHCP/PPPoE. If two default routes exist, the one with lower distance is preferred. The second can act as backup if distance is higher.

Routes can also mark if they're dynamic (added by DHCP or routing protocols). You can filter or check status: connected routes have an "A C" flag (active, connected), static user-added routes "A S", dynamic from DHCP "A S D" (D=dynamic), etc. "S" alone means it's set but maybe not active (like missing gateway).

9.18 SMB

The SMB server allows the router to share files over SMB (Windows file sharing). This can turn the router's local storage into a network share (handy if you store backups or logs on it). Only SMB2/3 are supported (not SMB1).

To use SMB:

- Enable the SMB service and set a **NetBIOS name/Workgroup** if desired (defaults are fine, it will appear as "Antaira" or similar in Network Neighborhood).
- Add a **Share**: point it to a directory. By default, Airolinx 6 has disk folders like flash or disk1. You could share the entire disk or a subfolder.
- Users: define SMB users (Airolinx 6 uses its own user list under ip smb users). There's a guest account by default (which you can enable/disable). You can add a user with a password for access.

Once configured, from a PC you can map a network drive to \<router_ip>\ <share_name> using the credentials. Keep in mind file performance is limited by router hardware (and not all routers have a lot of storage space). But it's good for quick file transfers or using the router as a drop box for files.

9.19 SNMP

Here you configure SNMP (Simple Network Management Protocol) for network monitoring systems. Airolinx 6 supports SNMP v1,2c,3. You can set the community strings, allowed networks, and engine ID for v3, etc.

For basic use, enable SNMP, set a community name (like "public") with at least read access, and restrict it to your management network address (so not anyone on the internet can query). Then your NMS (like PRTG, Zabbix, etc.) can poll the router's OIDs for data (traffic, CPU, etc.).

Note: On a busy router, reduce SNMP polling frequency if possible to lower CPU usage. Also SNMPv3 for secure authenticated access is recommended in sensitive environments.

9.20 SSH

The SSH server is enabled by default on Airolinx 6 for secure shell access. In IP > Services, you can see "ssh" service running on port 22. You typically don't need to configure much in IP > SSH (that submenu is for key management like importing public keys for key-based auth, and adjusting ciphers if needed).

To use it, simply SSH to the router's IP with your user (usually "admin") and password. For increased security, you can:

- Use RSA/DSA/ed25519 keys — import your public key under System > Users or IP > SSH Keys for your user. Then disable password auth (via `/ip ssh set allow-password-login=no` or in that menu).
- Change the default port (in IP > Services, change SSH from 22 to something else). This isn't high security but avoids constant internet background noise scanning port 22.

9.21 Services

IP > Services lists all management services and their ports:

- telnet (port 23) — often disabled by default nowadays.
- ftp (21) — used for transferring files, can disable if not needed or secure (since it's cleartext).
- www (80) — the HTTP WebFig. By default, Airolinx 6 might have www service enabled but redirecting to HTTPS.
- www-ssl (443) — the HTTPS WebFig/WinBox over web.
- winbox (8291) — the Winbox GUI tool port.
- api (8728) and api-ssl (8729) — ports for the Airolinx 6 API (for automation scripts or external software).

You can disable services you don't use to reduce attack surface. For instance, if you always use Winbox or HTTPS, you might turn off plain HTTP and telnet. You can also set "Available From" to restrict access to certain IPs (e.g., only your IT subnet can Winbox in).

9.22 Settings

Misc IP settings: here you find toggles like **RP Filter** (reverse path filtering for anti-spoofing), **Accept Source Route** (generally leave off for security), etc. Usually defaults are fine. This also is where you can enable IPv4 FastTrack (though that's normally done by firewall rule, not here, in current Airolinx 6). If you have specific need to adjust ARP timeouts or other system-wide IP parameters, you'd do it here.

9.23 Socks

Airolinx 6 includes a SOCKS proxy server. A SOCKS proxy allows TCP connections to be relayed through the router. For example, if you point a PC's browser at the router's SOCKS proxy, websites you browse will actually be fetched by the router (useful in some network setups or for testing).

To enable, set **enabled=yes** and choose a port (default 1080). You should also add an **Access List** entry to allow only your LAN or specific hosts, otherwise it could become an open proxy. The access list rules can allow or deny by source address and destination addresses/ports.

Security: If not using, leave it disabled. If using, absolutely restrict it. An open SOCKS proxy can be abused (e.g., by spammers).

9.24 TFTP

The router can act as a TFTP server, which is useful for network booting devices (PXE) or for Airolinx 6's own Netinstall process. By default, TFTP is inactive.

Under IP > TFTP, you add rules specifying interface, IP range allowed, and real file path mapping for requested files. A common use is to serve Airolinx 6 packages or config files to devices.

A simple setup: Suppose you want to netboot a device with a file Airolinx 6.npk. You'd put that file in files (accessible in Files menu) or flash, then add a TFTP rule with req-filename matching that name and real-filename pointing to the actual file path. Also specify an IP range (e.g., 192.168.88.0/24 if the device is in that subnet) and enable read requests. Set allow-rollover if needed for large file. Now a TFTP client requesting "Airolinx 6.npk" would get that file from the router.

9.25 Traffic Flow

Traffic Flow is Antaira's name for NetFlow/ipfix exporter. When enabled, the router will send summaries of traffic flows to a specified collector server. This is used for traffic analysis, accounting, etc.

To configure, you add a **Target** (the collector's IP and port, and version — typically NetFlow v9 or IPFIX). You also enable Traffic Flow itself and select what to export (interfaces, or just overall). The output is UDP packets to the collector containing flow records like source/dest IP, ports, bytes transferred, etc., which can be graphed in software like ntop, etc.

It's primarily for advanced monitoring; home users don't need it. On high traffic, enabling this can add overhead, but for reasonable loads it's fine. Remember that hardware switching (fastpath) traffic might not be seen by CPU to report (e.g., wire-speed switch chips bypass CPU and thus traffic-flow, unless you configure otherwise).

9.26 UPnP

UPnP (Universal Plug and Play) allows devices on the LAN to request the router to open ports for them (used by games, Skype, etc., to become reachable from outside without manual port forwards). Airolinx 6 implements UPnP IGD (Internet Gateway Device) standard.

To use, enable UPnP and add at least one **internal** and one **external** interface in IP > UPnP > Interfaces. Internal is usually your LAN (where requests come from), external is WAN. Leave "Allow Disable External" default (it's a somewhat unsafe feature of UPnP IGD that can let a LAN device turn off the WAN — Airolinx 6 allows you to disallow that).

Once set, PCs or consoles can automatically open needed ports. In IP > Firewall > NAT, you'll see dynamic entries appear with comment "UPnP" for those mappings. Monitor those to ensure they're expected. If a malicious app tried to open lots of ports, you might reconsider using UPnP.

In small networks UPnP is convenient, but in secure environments it's often disabled to prevent unauthorized opening of ports.

9.27 VRF

VRF (Virtual Routing and Forwarding) allows multiple separate routing tables on the same router. This is primarily used by ISPs or advanced setups where you have to segregate traffic (e.g., MPLS/VPN scenarios). If you don't explicitly need it, you won't interact with it — all interfaces belong to the "main" routing table by default.

In Airolinx 6, creating a VRF is done by specifying which interfaces belong to that VRF and giving it a name. Then any routes for those networks are kept separate under that name. You may also use routing rule to direct certain traffic into a VRF.

An example might be if you had one router serving two clients and you want their routes separate (like two virtual routers in one). You'd put each client's interface in its own VRF and that's it — then their IPs won't mix routes. VRFs tie into BGP for MPLS VPN deployments heavily.

For most enterprise or personal usage, VRFs won't be needed. But it's good to know the device supports them if necessary.

9.28 Web Proxy

The Web Proxy in Airolinx 6 can act as an HTTP proxy server, caching web content and filtering URLs. It's not enabled by default. It supports two modes: normal (where clients explicitly configure the router as proxy) and transparent (where the router intercepts HTTP traffic via firewall NAT rule).

In IP > Web Proxy, set **Enabled=yes**. Choose a port (8080 is default). Set max cache size if you want caching (if the router has USB storage or enough flash). Also can set cache-On-Disk if storage allows.

You can define **Access** list rules to allow/deny certain URLs or destinations (for example, block access to certain websites by pattern matching on URL). There's also a Cache list for fine tuning what is cached.

For transparent mode, a firewall rule is needed: dst-nat port 80 traffic from LAN to the router's LAN IP on port 8080 (where proxy listens). This effectively directs all HTTP to the proxy. Keep in mind HTTPS cannot be transparently proxied by this method (it will just pass through, which is fine).

The integrated proxy is good for small scale or specific tasks (like captive portal advertisement injection or simple URL blocks). For heavy use or modern web (HTTPS), its utility is limited (can't cache HTTPS well). Many production environments use external dedicated proxies if needed.

9.29 Accounting

IP Accounting is a simple traffic accounting feature. When enabled, the router tracks how many bytes each host sends/receives through it. You can query these via SNMP or the CLI. It's mainly legacy; most will use **Traffic Flow** or SNMP counters for more detailed accounting.

However, if you enable IP > Accounting, you can then see in IP > Accounting > Hosts a list of top talkers and traffic counts. It's handy if you want a quick built-in way to see usage. Just note it's approximate and limited (and might impact performance if many hosts).

10 IPv6

10.1 Introduction

The IPv6 menu mirrors many of the IPv4 features, but for IPv6 addresses and services. Airolinx 6 treats IPv6 as a separate package (ensure it's enabled). You'll find:

- **Addresses** — IPv6 address assignments (128-bit addresses, written in hex notation).
- **DHCP Client/Server/Relay** — For IPv6, typically DHCPv6-PD (Prefix Delegation) for WAN client, and maybe DHCPv6 server for handing out IPv6 addresses, though SLAAC (stateless address auto-config) is more common for clients via ND.
- **Firewall** — Separate IPv6 firewall rules (similar chains: input, forward, output). Note: by default, Airolinx 6 might not have as strict default IPv6 firewall, so don't forget to configure it (e.g., drop external input, etc.).
- **ND** — Neighbor Discovery settings (similar to ARP and also includes Router Advertisement settings).
- **Neighbors** — IPv6 neighbor cache (like ARP table for v6).
- **Pool** — IPv6 pools (for DHCPv6 or prefix delegation).
- **Routes** — IPv6 routing table (you'll see connected routes for each /64 on interfaces, plus typically a ::/0 default route via your ISP if you have upstream IPv6).
- **Settings** — Misc IPv6 settings (e.g., accepting router advertisements, etc.).

Using IPv6 on the router might involve assigning a global IPv6 address to your LAN, either static or via prefix delegation from an ISP. Many ISPs hand out a /64 or /56 prefix via DHCPv6-PD. The router, as DHCP client, can request a prefix, then distribute it to LAN via SLAAC or DHCPv6.

10.2 Addresses (IPv6)

You can add an IPv6 address to an interface similarly to IPv4. Example: add address `2001:db8:100::1/64` on `bridge1`. That network then is on your LAN. By default, Airolinx 6 will also automatically configure a link-local address (fe80::) on each interface (needed for ND).

Often, you'll not manually add global addresses if you use prefix delegation; the DHCPv6 client will add them. But you may add ULA (unique local addresses, analogous to private IPv4 like fd00::/8 range) for internal use.

10.3 DHCP Client (IPv6)

Typically used on WAN interface to request an IPv6 prefix from ISP. Configure: specify interface, and request type. Usually you check "**Request Address**" if you want a specific WAN address and "**Request Prefix**" with a size (like 56 or 64) to get a delegated prefix. The ISP will respond with, say, a /60. The client will then automatically create a dynamic IPv6 address on the WAN if address requested, and list the received prefix in **Pool** (as a pool named DHCPv6-PD or similar).

You can then use that pool in a DHCPv6 server or simply let SLAAC handle it by advertising one /64 out of it on LAN. Airolinx 6 doesn't automatically do that; you must assign addresses from the prefix to LAN. A common approach: set `Address=::1` on LAN with prefix length /64 from pool. This tells Airolinx 6 to take the delegated prefix (e.g., 2001:db8:aa00::/56) and assign 2001:db8:aa00::1/64 to LAN. The pool's "prefix"

field gets replaced by actual prefix. This is a bit advanced usage of config — there are guides for how to do prefix delegation in Airolinx 6.

10.4 DHCP Relay (IPv6)

Similar to IPv4 DHCP relay but for v6. Rarely used outside specific enterprise setups, since SLAAC often negates the need for relays — if you have multiple segments you usually run separate /64s rather than relay one server's assignments across networks. But it's available if needed.

10.5 DHCP Server (IPv6)

An IPv6 DHCP server in Airolinx 6 is typically for handing out info that SLAAC doesn't cover (like DNS). It can hand out addresses too (Stateful DHCPv6), but many clients prefer SLAAC (Stateless Address Auto Configuration) for address and stateless DHCPv6 for just DNS.

Airolinx 6's IPv6 > ND (Neighbor Discovery) settings actually include options to specify whether to use M (Managed) or O (Other config) flags in the router advertisements. That triggers clients to use DHCPv6.

Setting up DHCPv6 server: define a pool (maybe from a ULA prefix, or from your PD prefix if subdividing). Then add server on interface, set that pool. Also define **Options** like DNS servers to advertise. Ensure ND on that interface has "Advertise" enabled and proper flags to tell clients to do DHCPv6.

10.6 Firewall (IPv6)

The IPv6 Firewall works like IPv4 but defaults might be more open. Best practice is to at least drop all inbound to the router except needed (ICMPv6 important to allow some types for proper ND and PMTUD), and possibly restrict forward if you want to limit what IPv6 can go through (though generally one doesn't NAT IPv6, so if you give clients global addresses, you likely want to firewall them from external threats). Antaira default IPv6 firewall (if applied via "firewall IPv6" default config) usually drops input to local except established/related and neighbor discovery, and allows forward established/related and drop new connections from WAN.

Don't forget to allow ICMPv6 types 133,134,135,136 which are ND (router solicitation/advertisement and neighbor sol/adv) — Airolinx 6 has default rules for that typically. Also allow DHCPv6 if you're using it (UDP port 546/547).

10.7 ND (Neighbor Discovery)

In IPv6, ND serves a similar role to ARP (resolving L3 to L2 addresses) and also handles Router Advertisements (RA) for SLAAC. The ND menu in Airolinx 6 lets you configure RA parameters per interface.

Select your LAN interface in ND and set:

- **Advertise**: yes (to send RA). If you only want clients via DHCPv6, you might still advertise but with prefix info managed =1 etc.
- **Reachable Time, Retransmit Interval**: usually leave default.
- **RA Interval**: how often to send unsolicited RA (and the valid lifetime for prefix).

- **Prefix**: Add your prefix(es) to advertise. e.g., if you have 2001:db8:aa00:1::/64 on LAN, ensure it's listed to advertise to clients. You can also allow them to auto-config addresses from that prefix (which they will if RA says so).
- **MTU**: can advertise an MTU (often 1480 if PPPoE, else 1500).
- **Managed/Other flags**: If you set "Managed" flag, hosts know to use DHCPv6 for address. "Other" means use DHCPv6 for extra info. For SLAAC + stateless DHCP (most typical), set Managed=No, Other=Yes (meaning they'll auto-address by prefix, but also seek DHCPv6 for DNS). If you want full stateful DHCP (less common), set Managed=Yes (and run a DHCPv6 server).

Properly setting ND ensures your clients get IPv6 addresses and default gateway (the router's link-local). Check with `ipv6 address` on a client — you should see an address from the prefix and a gateway (fe80::... router's MAC-derived LL).

10.8 Neighbors (IPv6)

This is the neighbor cache for IPv6 (like ARP table). It shows link-local addresses and MACs of neighbors, and also if a global address was resolved, etc. If connectivity issues arise, seeing the neighbor cache can help (e.g., if incomplete). You can ping link-local addresses from the router (remember to specify interface, e.g., `ping fe80::1%ether1` to ping a link-local on ether1).

10.9 Pool (IPv6)

Same idea as IPv4 pool but for IPv6. Often used for DHCPv6 server, or the DHCPv6 client will populate a pool with the delegated prefix (so you can then carve it up). Pools can supply /64s from a /56 etc, by setting prefix length appropriately.

10.10 Routes (IPv6)

Works like IPv4 routes. If you have a default route, it likely came from your ISP via DHCPv6 (which can be done by them sending a RA with default route info or via DHCPv6 option). You can also add static IPv6 routes if needed (`::/0` as default via your gateway's link-local typically).

Connected routes (C) for each /64, and perhaps a connected for your link-local "network" on each link (fe80::/64 etc) will appear. There's no ARP in IPv6 route, instead next-hop is usually a link-local of the neighbor.

10.11 Settings (IPv6)

General IPv6 config toggles. For example, **Enable IPv6** (should be yes if using). **Max Neighbors** in cache, etc. Usually fine as default. One thing: if you want the router itself to use DHCPv6 client to get DNS, ensure "Accept Router Advertisements" is set appropriately (but typically the DHCPv6 client will directly get DNS, or you static configure).

11 MPLS

11.1 MPLS

MPLS (MultiProtocol Label Switching) is a high-performance transport technique often used by ISPs for VPNs and traffic engineering. In Airolinx 6, MPLS allows you to enable label switching on interfaces, effectively preparing the router to forward labeled packets. The MPLS menu includes enabling the MPLS subsystem and setting up things like interface **MPLS MTU** (which needs to account for extra labels), Label Distribution Protocol (LDP) settings, etc.

On Wi-Fi 6 routers, MPLS is probably not a common use-case unless used in an ISP or large network core scenario. If needed, you'd enable MPLS on core interfaces (not on user-facing ones typically), and you'll see the router assign labels to routes. The **Bindings** section might list static label bindings for particular routes if any are set.

11.2 Settings (MPLS)

MPLS settings include the range of labels the router can use (like 16-1048575 are default usually), and whether to propagate TTL and such. Also, whether LDP is enabled by default on all interfaces or only certain ones. In an MPLS network, you often only run LDP on backbone links. You might have to disable it on others.

11.3 Traffic Eng

Traffic Engineering (TE) refers to MPLS TE tunnels (often signaled by RSVP-TE). Airolinx 6 supports RSVP-TE for MPLS tunnels. This sub-menu would be where you configure TE tunnels, their bandwidth reservations, etc.

Again, an advanced ISP feature — you'd need to set up an RSVP interface, TE policies, etc. If you don't specifically plan to use MPLS TE, you won't touch this.

11.4 VPLS

VPLS (Virtual Private LAN Service) is a Layer2 VPN technology running over MPLS. It allows two or more sites to be in the same Ethernet broadcast domain over an MPLS cloud. In Airolinx 6, you can configure VPLS interfaces either using BGP signaling or LDP-based signaling.

A VPLS interface will appear as something you can put into a bridge (like it's an Ethernet cable to the remote). Setting up VPLS requires either a BGP MPLS-VPN configuration or manual LDP setup of pseudowires. If doing BGP-based, you'll define a BGP VPLS instance with Route Targets, etc.

For completeness: a simple two-router VPLS via LDP can be done by creating VPLS interface on each, specifying the remote peer ID and a VLAN ID (to differentiate if multiple). They will find each other over LDP and establish the pseudowire, if MPLS is running underneath.

Unless you're building a lab or an ISP MPLS network, you won't use VPLS on a typical Wi-Fi router. It's included for those advanced scenarios though.

12 Routing

12.1 BFD

BFD (Bidirectional Forwarding Detection) is a protocol for rapid detection of link failures between two devices. It's often used in conjunction with routing protocols like OSPF or BGP to quickly sense when a neighbor is down (within milliseconds, much faster than default timers).

In Airolinx 6, BFD can be configured per interface or per routing peer. You set desired transmit/receive intervals and multiplier (how many missed packets before declaring down). If BFD is active between two routers on a path, they can detect failures in ~sub-second and inform the routing protocols to reconverge faster.

To use BFD, both ends must support it (Airolinx 6 does). You usually configure it inside OSPF/BGP settings or directly link it to neighbors. For example, in OSPF interface template, enable BFD. Then the OSPF neighbor relationship will drop as soon as BFD says the link is down.

Check BFD status in this menu (sessions, timers) when in use. If not using dynamic routing, you won't need BFD.

12.2 BGP

BGP (Border Gateway Protocol) is the routing protocol used to exchange routes between different networks (autonomous systems). If your device acts as a border router (for example, connecting to an ISP with BGP to exchange routes or participating in an internal BGP for large networks), you'd configure BGP here.

Airolinx 6 BGP supports both IPv4 and IPv6 (address families), communities, route reflectors, etc. Under Routing > BGP, you set up a BGP instance (which identifies your AS number), then add **Peers** with neighbor IPs and remote AS, and optionally use Routing Filters to control announcements.

For example, to accept a default route from ISP and advertise your prefix: set Instance AS to your AS (if you have one) or use AS provided by them if it's private. Add Peer with their IP, remote AS (ISP's), set multihop if not directly connected, and that's the basics. Use filters to only accept 0.0.0.0/0 or to only announce your prefix.

BGP on a small router is feasible if the route load isn't too high. Full internet BGP table (800k+ routes) might be too heavy for a modest device memory. Typically, small devices do BGP only for default route or partial routes. Keep that in mind.

12.3 Filters (Routing)

Routing Filters are rules that manipulate routes learned or advertised by dynamic protocols (BGP, OSPF, RIP). For BGP, you might set filters to only allow certain prefixes out or in, to set BGP communities or local-pref, etc. For OSPF, filters can set distance on certain routes, etc.

They operate with conditions (match prefix, length, protocol) and actions (accept, reject, set values). They are processed sequentially for each route event.

One common filter for BGP might be: if prefix length > 24, reject (to avoid too specifics). Or set default route from ISP to have distance 110 to use as secondary, etc.

This section can get complex — essentially if using dynamic routing, set up filters as needed to enforce your routing policy. For static or no dynamic protocols, nothing to do here.

12.4 IGMP

This likely refers to IGMP settings or pim IGMP interface (the text had "GMP" which likely is IGMP). IGMP is used for IPv4 multicast group management between hosts and local routers (for example, IPTV networks where clients join multicast streams). Airolinx 6 allows IGMP Proxy or PIM (below) to handle multicast.

There's no explicit IGMP config needed usually aside from enabling IGMP Proxy if using that approach. If acting as a multicast router, the IGMP memberships can be viewed under PIM or IGMP menus. If you have no multicast streaming usage, you won't deal with this.

12.5 IGMP Proxy

IGMP Proxy is a simpler form of multicast routing where one upstream interface and one or more downstream interfaces exist. Airolinx 6's **IP > IPTV** (in some older Winbox, or in Routing menu IGMP Proxy) is where you configure it. Essentially, specify the upstream (where router joins groups from an ISP) and downstream (LAN where hosts request groups). The router will proxy IGMP membership reports upstream. This is used by some ISPs for TV multicast when you only have one router in between.

Configuration: add an instance, set upstream interface role=upstream, and downstream interfaces role=downstream. Possibly add altnet for any source subnets allowed. Then when clients send IGMP joins, the router will subscribe upstream.

If using PIM (below) in a more complex multi-hop scenario, you wouldn't use IGMP-Proxy at same time typically.

12.6 Nexthops

Likely refers to a menu showing ECMP (equal-cost multi-path) nexthop details or active BGP multipath. Wi-Fi 6 Device OS introduced a new routing engine where "Nexthop" objects exist separate from routes, for resolution. This might display resolved nexthops (like if a route's gateway is an IP, the actual link-layer neighbor or interface is a resolved "nexthop" entry).

This is quite internal. Typically not manually configured, but you might see entries when using policy routing or VRFs.

12.7 OSPF

OSPF (Open Shortest Path First) is a link-state interior gateway protocol for dynamic routing within an AS (like within a company). Airolinx 6 supports OSPFv2 for IPv4 and OSPFv3 for IPv6 (OSPFv3 is configured separately under IPv6 menu or integrated now with new routing possibly).

Setting up OSPF involves creating an OSPF instance (specify router ID), possibly areas (default 0.0.0.0 backbone), and then adding interfaces or networks to OSPF. In Airolinx 6, an easy way is using **Networks** :

specify network like 192.168.1.0/24 in area backbone, and any interface with IP in that network will run OSPF. Alternatively, define interface templates to set cost, etc.

When OSPF is running on two routers on a link, they form adjacency (check with Routing > OSPF > Neighbors). Then they exchange LSAs and each installs routes accordingly. OSPF requires careful area design if multiple areas, and possibly route filtering if you need to restrict some routes distribution.

For a small network, OSPF might not be needed if static routes suffice. But it shines in medium to large networks for auto rerouting around failures. Ensure to adjust interface cost if you want to prefer one path over another. OSPF in Airolinx 6 also has stub area support, etc.

12.8 PIM SM

PIM-SM (Protocol Independent Multicast - Sparse Mode) is a multicast routing protocol to handle multiple routers distributing multicast between networks. If you have a complicated multicast setup (multiple subnets with viewers and maybe sources), PIM is used with a Rendezvous Point (RP) configuration to manage group joins across the network.

Airolinx 6 supports PIM-SM. You need to enable PIM on interfaces that have multicast listeners or sources, and configure an RP (or use auto-RP/BSR). PIM will then build multicast distribution trees (Shared tree via RP, and possibly shortest-path trees). This is advanced and typically used by ISPs or large campus with IPTV systems.

On small scale, IGMP-Proxy is easier if topology is simple tree. PIM-SM is for general topologies. Config: define PIM instance, RP (static RP address or AutoRP). Ensure PIM enabled on interfaces and that IGMP is enabled on edge interfaces (to pick up group joins from clients). Then verify via PIM neighbors and multicast route entries.

12.9 RIP

RIP (Routing Information Protocol) is a simpler, older dynamic routing protocol (distance-vector). Airolinx 6 supports RIPv2 for IPv4 and RIPng for IPv6. It's rarely used nowadays except in very simple or legacy networks, since OSPF or even better protocols exist. But it's easy to set up.

To use RIP, enable it on interfaces (set networks or interfaces in RIP settings). All routers with RIP on those nets will exchange routes (limited to hop count 15, etc.). It's basically plug-and-play but lacks advanced features and converges slower than OSPF.

If you do use it, remember to set up any needed filtering (RIP can flood a lot if not careful). But likely not used unless required for compatibility with a device that only speaks RIP.

12.10 RPKI

RPKI (Resource Public Key Infrastructure) in Airolinx 6 relates to BGP, specifically origin validation for received routes. Airolinx 6 can connect to RPKI caches to fetch ROAs (Route Origin Authorizations) and then mark BGP routes as valid, invalid or unknown based on if the origin AS of a route is authorized for that prefix.

In Routing > RPKI, you can add an RPKI server (like an RPKI Validator IP and port). The router will fetch data. Then in BGP filters, you can use "rpki-validation" condition to perhaps drop invalid routes or prefer valid ones.

For an operator on BGP, enabling RPKI helps avoid accepting potentially hijacked prefixes. This is advanced BGP security. It requires you run or have access to an RPKI cache (there are public ones you might point to).

12.11 Router ID

Router ID is an identifier often an IP address (not necessarily an address the router has, but often one of them or a manually set one) that identifies the router in OSPF and BGP (and other routing protocols). Airolinx 6 in v7 has a concept of a global Router ID configuration, perhaps here, that if set, is used for any protocol that needs it (unless overridden in protocol instance settings).

It's good practice to set a Router ID (like use a loopback's IP or any unique number in IPv4 format) to ensure consistency. If not set, Airolinx 6 might pick one (like highest IP). That can change if that interface goes down. So better to set it explicitly.

12.12 Rules (Routing)

Routing Rules in Airolinx 6 allow policy routing decisions. For instance, you can route certain source IPs via a different routing table (as created in VRF or via separate table config). A rule can match on source address, destination, firewall mark, etc., and then direct the packet to a named routing table or VRF.

Use case: dual WAN policy routing. You mark connections from a certain LAN with a routing mark "useWAN2" via mangle. Then add a Routing Rule: if routing-mark=useWAN2, lookup in table "WAN2". And your WAN2 specific routes (like default via WAN2 gateway) live in that table. This way those connections use WAN2 regardless of main table default route.

Routing Rules can also drop or lookup in main or alternate. They are powerful when default behavior of single routing table doesn't suffice.

12.13 Settings (Routing)

General settings for the routing subsystem. Possibly toggles for ECMP algorithm (per-connection or per-packet), etc. Wi-Fi 6 Device OS's routing is much improved and these settings might control some aspects of FIB installation, etc. Usually, default fine, unless you need a specific behavior (like choosing how ECMP load-balance, there's an "ecmp symmetric" option to keep flows symmetric).

12.14 Tables

This refers to the actual routing tables configured. We have the main table by default and any user-defined ones (like if VRFs create new tables or we manually add via /routing table add fib name=XYZ). In Airolinx 6, each VRF adds a table of same name by default. The Tables section will list those and if they are FIB (installed in kernel) and how many routes, etc.

You generally don't need to touch this unless adding a custom routing table for policy routing. If so, you'd add one here, then use in rules or address assignments, etc.

13 System

13.1 Certificates

The Certificates menu is for managing x.509 certificates on the router. These are used for services like SSTP, OpenVPN, WWW-SSL (HTTPS certificate for WebFig), or EAP-TLS, etc. You can create self-signed certs, generate CSRs, import CA certificates, etc.

For instance, to set up SSTP (an HTTPS-based VPN) you might create a CA cert, then a server cert signed by it. Or import a Let's Encrypt certificate for the HTTPS GUI.

Airolinx 6 supports running a simple certificate authority: you generate a template, sign it to create a CA, then generate server certs signed by that CA. The entries show as CA or L (labeled) for ones with private keys.

Double-click a certificate to see details or set it to be used for particular services (some services allow specifying which cert to use). Ensure correct time on router when dealing with cert validity.

13.2 Clock

Simply shows or sets the system date/time. On initial setup, the router might not have correct time until NTP or manual set. You can manually set time here or via command. But better is to use NTP Client.

However, make sure the time zone is correct (System > Clock has Time Zone Name setting). It's often UTC by default. Set your appropriate zone or offset.

13.3 Console

This might refer to settings for the serial console or terminal console behavior. For instance, on devices with a serial port, you can choose the speed (115200 by default for Airolinx 6 on many). Also, you can enable/disable serial console (if you want that port for other uses).

Additionally, Airolinx 6 has a "virtual console" concept and logging. But likely here it's the serial console port config. If you have no physical serial, rarely needed.

13.4 Health

The Health menu shows hardware health metrics: board temperature, CPU temperature, voltage, fan speed (if applicable). For PoE devices, maybe current draw. It's a monitoring readout; you don't configure anything here, just view.

If your router is in a hot environment, check these readings. You can use /system health print in CLI too. If temp is beyond spec (>70C CPU sustained), consider cooling or lighter load.

13.5 History

History logs events or command history. It might list users who logged in and what commands executed (Airolinx 6 does have a command history per user, and also a system history for config changes maybe). In Winbox, "History" shows recent configuration changes with time stamps. This helps track what was changed and possibly revert if needed.

It is separate from the sys log. It's more of a configuration history. Use it to audit or recall changes.

13.6 Identity

System Identity is simply the router's name (hostname). This is shown in Winbox neighbor list and SNMP system name. Set it to something descriptive, e.g., "MainRouter" or location-based. It's one of the first things to customize. (It's also used in CapsMAN for naming, etc.)

13.7 LEDs

If your device has programmable LEDs, this menu lets you assign triggers to them. For example, many Antaira devices have a "USR" LED or others which can indicate wireless activity, LTE signal, etc. You can set an LED to follow an interface's link status or blink on traffic, etc.

This is mostly cosmetic or for quick visual checks. Each LED entry can tie to an interface and condition. E.g., LED1 -> wlan1 signal strength (blinks faster on good signal).

13.8 License

Shows your Airolinx 6 license level, due date (if trial or demo), and allows pasting a new license key if you upgrade license. Antaira devices likely come with a Airolinx 6 license appropriate for the model (levels 4,5,6). This page also shows the software ID which is used to get a license key from Antaira if re licensing.

If cloud license is used (for CHR), it would also be managed here. But typically, a physical device's license is perpetual and already installed.

13.9 Logging

Here you configure system logging. By default, some topics (error, warning, info) go to memory buffer and maybe to the console. You can add new rules to send logs to disk or a remote Syslog server.

For example, to log firewall drops to an external syslog, you set firewall rules with action=log and topics, then in Logging, add a rule for topic firewall to go to remote. But primarily:

- **Actions**: pre-defined destinations like memory (log buffer, viewable in Log), disk (if router has storage), echo (to current CLI session), email (send log via email), remote (to syslog). You can add custom remote actions with specific IP/port.
- **Rules**: Each rule says for topics (one or multiple) use an Action. E.g., topic=system,info goes to memory by default.

You can adjust to reduce noise (e.g., disable info logging) or capture more (set firewall logging as needed). Check /log print to see current log buffer. Keep in mind disk logging can wear out flash if very frequent; remote logging is preferred for heavy logs.

13.10 NTP Client

The NTP client synchronizes the router's clock with internet time servers. In older Airolinx 6, there was separate "System > NTP Client", but now often they integrate under System > SNTP or so. Either way, you input NTP server addresses (e.g., pool.ntp.org) and enable it. The router will periodically sync time.

Keeping time correct is crucial for logs and any certificate validation.

13.11 NTP Server

Airolinx 6 can act as an NTP server (if you have NTP package or SNTP server enabled). This could serve time to your LAN devices. For simple networks, not often used as many devices directly use public NTP. But in isolated networks or where you want a central time source, you could use it.

Configuration would involve enabling it and possibly specifying if it should act as a peer in symmetric mode or broadcast, etc. Most small scale usage doesn't turn the router into an NTP server unless needed.

13.12 Note

You can store a custom note on the router — e.g., asset ID, installation date, etc. This is purely for administrative reference. Useful if managing many routers to leave notes for colleagues or future yourself, like "Installed in Building A, 3rd floor on 2025-01-01."

13.13 Password

This allows changing the password of the default admin (or currently logged in user). On modern Airolinx 6, there is a user management section too (System > Users). But "Password" here is a quick way to change your own password. It will prompt for old (if any) and new. This is critical to set on first use (if not done in Quick Set).

13.14 Ports

This likely refers to Serial ports or USB port management. For example, if the device has a USB and you plug a 3G modem, it shows up under Ports (like usb1). You might need to configure serial port settings or channel usage for certain modems or GPS devices.

Also, on boards with serial, "ports" menu allows splitting serial ports or using them for different features (like a serial terminal server set via Special Login as earlier). For most users, nothing needed here unless working with serial-connected devices.

13.15 Reboot

Clicking this will soft reboot the router. There's no configuration except acknowledging the action. (From CLI, /system reboot does same.) After reboot, the router reloads configuration from store.

13.16 Reset Configuration

This allows you to wipe the current configuration and optionally set it to defaults. Use with caution. You might choose "No Default Configuration" if you want it blank or "Reset to Defaults" to load the factory default (the quick set config). You can also choose to keep user's password or not on reset.

Often used if you want to start over. It will cause a reboot into a clean config as selected. It's like factory reset (which you can also do via physical button on many devices).

13.17 Resources

Shows system resource usage and info: CPU architecture, frequency, RAM total/free, storage free, firmware version, etc. It's useful to see memory usage (especially if you run many features or full BGP, watch that free memory doesn't drop too low).

Also shows uptime, CPU load % (though CPU load is often better seen with Tools > Profile). But Resources gives a snapshot of health and capacity. If you see high CPU often or memory low, consider upgrading hardware or optimizing config.

13.18 Scheduler

The scheduler can run scripts or commands at specified times or intervals. It's like cron. You add an entry with a start date/time and interval or specific timing, plus the script to run (or call an existing script from /system script).

Examples: schedule a daily reboot at 3am (not usually needed, but some do). Or schedule a backup file email every week. Or enable an interface at 9:00 and disable at 17:00 daily, etc.

Be careful that scheduled tasks have proper permissions (they run as admin by default). Also monitor that they actually did what you want (check logs if script uses :log info to report something).

13.19 Scripts

Here you can create and store script files (sets of Airolinx 6 commands) to automate tasks. The scripting language allows variables, loops, conditionals. Scripts can be run manually, or by scheduler, or triggered by netwatch, etc.

Create a new script, give it a name and write commands in the source. For example, a script "blink-led" that toggles an LED in a loop a few times. Or a script "failover" that checks a ping and swaps routes (though netwatch might be easier for pings).

Scripting is powerful but be mindful of side effects (and infinite loops or heavy tasks can cause issues). Also secure any sensitive info (like if script contains passwords). Airolinx 6 script is plaintext stored (unless you encrypt it externally).

13.20 Shutdown

Performs a graceful shutdown (for devices with NAND flash to avoid corruption). After clicking, the router will power off (if supported) or halt. Typically used before removing power if it's a device where that matters (Antaira devices with NAND should be fine to just power off usually, but recommended to shutdown if possible).

Devices without a power button basically need power cut after issuing shutdown. They won't restart themselves until power-cycled.

13.21 Special Login

As touched on earlier, special-login is used to forward a login to the serial port for managing another device. For instance, if you have a switch or another router console cabled into this router's serial, you can

create a special user (like username “serial”) tied to that port. When you SSH to the router with that user, it directly connects you to the serial-attached device’s console.

In System > Users, such a user would have “port” set to serial0 and group special. That’s managed under special login. This effectively makes the router a console server.

Unless you have that scenario, you might not use this. But for out-of-band management deployments, it’s handy.

13.22 Users

User accounts management. Default is admin (full). You can add users with groups (full, read, write, custom). If giving someone limited access, use group write (can do most but not sensitive like user management) or read (monitor only). You can also define custom policy sets (like allow specific subset of commands).

It’s advisable to disable or rename the default “admin” user and use a new one (to make brute-force slightly harder). Also enable only secure access (SSH, HTTPS, Winbox which is encrypted) and use strong passwords or keys.

You can also set up login via RADIUS for users if integrated with a central AAA server — not common in small setups, but possible (see /radius menu).

13.23 Watchdog

Watchdog can automatically reboot the router if it becomes unresponsive or if certain conditions occur. There’s a basic watchdog that pings a certain host — if ping fails, it can reboot (useful for remote sites where a router might hang due to a glitch, or if it loses connectivity maybe reboot helps recover cellular link etc.).

Configuration: enable watchdog, set a ping-target (like 8.8.8.8 or gateway IP). If that target is unreachable for a period, it triggers reboot. Also, watchdog can trigger on system hang (if Airolinx 6 crashes or goes unresponsive, hardware watchdog timer reboots it).

There’s also an option to run a script on no-ping instead of reboot (Netwatch does that usually though). Another piece: “automatic supout” generation on crash which is helpful for support analysis (supout.rif file).

Use with care — if ping target fails due to ISP but router is fine, you might get unnecessary reboot loops. Possibly point it to something reliable like directly attached IP or use netwatch with more logic instead of hard watchdog.

14 Queues

14.1 Introduction & Types

Airolinx 6 offers bandwidth management through the **Queue** system. Queues can limit and prioritize traffic. There are two kinds of queues:

- **Simple Queues**: Easiest way to limit bandwidth per IP or group of IPs. Each simple queue can have a target (one IP or a subnet or an interface) and max limits (upload/download). Simple queues are processed in a list order.
- **Queue Tree**: More advanced, hierarchical shaping. You attach queue trees to interfaces or global, mark packets (via firewall mangle) for classes, and then set bandwidth per class with priority. Use this for complex setups (like guaranteeing certain service minimum bandwidth).

Also, underlying **Queue Types** define the scheduling algorithm (pfifo, bfifo, SFQ, PCQ, FQ-CoDel, etc.). For example, PCQ (Per Connection Queue) is great for evenly splitting bandwidth among multiple IPs (e.g., a queue to share 100M among all 50 users fairly).

14.2 Configuration Steps

Simple Queue Example: Limit 192.168.1.50 to 5 Mbps download, 1 Mbps upload.

1. Go to Queues > Simple Queue, click "+". Set **Name** : e.g., "User50-limit".
2. In **Target** , put 192.168.1.50 (or choose the interface if targeting whole interface).
3. Set **Max Limit** : target upload (Tx) = 1M, target download (Rx) = 5M (values can be in bits or bytes, e.g., "5M" means 5 Megabit). Optionally set burst if using bursting.
4. Leave other settings default (pfifo queue type default, etc.). Click OK. Now that IP's traffic should be throttled accordingly. You can observe in the queue list the current rates.

Simple queues are applied in order; you can nest them (one parent queue with overall cap, children per user with individual cap). If multiple simple queues match the same traffic, the first match will catch it. So you usually ensure they don't overlap unless using parent/child relationship explicitly.

Queue Tree Example: Suppose you want to limit overall download on WAN to 50M and ensure VoIP (marked packets) gets priority within that.

1. First, use Firewall > Mangle to mark VoIP packets (e.g., mark packets with protocol UDP and port 5060 as "voip").
2. In Queue Tree, add a queue on interface WAN (assuming WAN interface is where download comes in). Set **Parent** = WAN (or could use global-in for all incoming). No packet mark (this will be the total traffic). Set **Max Limit** = 50M.
3. Add another queue, Parent = the above queue (or parent=WAN with queue type "prio" classes). Set Packet Mark = voip, give it **Max Limit** maybe 5M and priority=1 (high priority). Add another queue for other traffic (no mark) with the rest of bandwidth and lower priority.
4. Now, the tree will ensure combined doesn't exceed 50M, and voip up to 5M is prioritized. If voip isn't using that, other traffic can use full 50M.

Queue Trees require careful use of packet marks and choosing correct parent (global-in, global-out, or interface) depending on direction of traffic and whether pre/post routing. Typically, download shaping is done on global-in or on LAN interface for control after NAT. Upload on WAN interface (as leaving router).

14.3 Queue Types

Under Queue Types, you can see default types like default-small (pfifo 10 packets), default (pfifo 50), ethernet-default (pfifo 50). Also PCQ types for download/upload default — these are used for PCQ simple queues automatically. And "default-fast" might be FQ-CoDel (for new Airolinx 6 enabling Cake or fq_codel to reduce bufferbloat).

You usually don't need to modify queue types unless implementing something specific, like making a PCQ type to equalize per IP. For example, PCQ with classifier dst-address for download ensures each destination IP gets equal share. Then you use that type in a simple queue that targets a whole subnet with that PCQ type.

Wi-Fi 6 Device OS added Cake queue type as well (which is good for home use to handle bufferbloat easily). If present, you might set your simple queue type to cake (in Queue Simple, set Queue Type for target to cake) and it will fair-queue traffic nicely with minimal setup.

14.4 Troubleshooting & Tips

Queue not working: Check that the target traffic is actually going through the queue. If you set target to an IP but traffic is forwarded (not to router), it should catch. But if hardware offloading is on (e.g., FastTrack firewall rule active), those flows might bypass queues. For QoS, you should disable FastTrack for that traffic because FastTrack will skip queue processing. Remove fasttrack rule or add exceptions for the IPs you want to shape.

High CPU usage: Queueing is done in software. At high throughput, the CPU might become a bottleneck, especially with complex queue trees or many simple queues. If you need to shape hundreds of Mbps, ensure the router's CPU is powerful. PCQ and Cake are heavier than simple pfifo but provide fairness. Monitor /tool profile to see CPU in "queueing" tasks.

Prioritization: If using priority in simple queues, note that simple queue priority is relative only among siblings under same parent. In simple queue top-level, lower number = higher priority, meaning if bandwidth is contested, higher priority queues get their limit first. But it's not strict priority like dropping others entirely, it's about who gets leftover above guarantees.

Testing: Use Tools > Torch or interface traffic to see current usage. The Queues window also shows current rates and drops. If a queue is dropping packets (you'll see drop counter), that means it's actively limiting traffic (which is normal when hitting the limit). If no drop and target is exceeding set limit, likely configuration issue.

QoS can be complex. For many, a simple overall rate-limit per user (with maybe burst for short term speed) is enough. Others might require ensuring time-critical traffic (VoIP, gaming) is prioritized — that's where queue tree with priorities or use of newer FQ-CoDel/Cake which automatically tries to prevent any one flow from causing latency, is beneficial. Airolinx 6 gives the tools, but one must tune them to the specific network needs.

15 Dot1X

15.1 Introduction

The Dot1X menu corresponds to IEEE 802.1X port-based authentication (for Ethernet or WLAN). This allows a device (supplicant) to authenticate with the network before being allowed access. Airolinx 6 supports both acting as a supplicant (client) on an interface, or as an authenticator (server) on a port, and even as an authentication server (if User Manager is installed).

15.2 Supplicant Configuration

If you want the router to authenticate itself on a network (say, ISP requires 802.1X on WAN port), configure under **dot1x client**:

- Interface: which port to run 802.1X on (e.g., ether1).
- Identity and Password: credentials given by the provider to authenticate (or certificate if EAP-TLS method).
- EAP method: typically EAP-PEAP or EAP-TTLS with MSCHAPv2 for username/password, or EAP-TLS for certificate-based.
- Client certificate, if needed, and CA certificate if validating server.

Enable the client. The status can be checked; if success, the port will be authorized and traffic will pass.

15.3 Authenticator Configuration

If you want the router to enforce 802.1X on one of its ports (for example, only allow computers that have valid credentials onto a certain LAN port), configure **dot1x server**:

- Interface: the port which will require auth.
- Radius: set up a RADIUS client under /radius to forward the auth requests to a RADIUS server (could be Airolinx 6 itself with user manager or an external one). Or use the local user database if simple (Airolinx 6 might support only RADIUS for 802.1x though).
- Allowed status: you can have a guest VLAN for unauthenticated, which you can set in options if supported (Airolinx 6 had "guest" for wireless, for wired not sure if implemented similarly).

Then when a device connects, the router will challenge it for 802.1X. Until success, the port won't forward normal traffic (except EAP packets). On success, the port becomes active for that client.

Note that on a managed switch you typically have multiple MACs and dynamic VLAN assignment; on Airolinx 6, since it's mostly a router, authenticator on a port means essentially one device behind it. But some Antaira devices (like CRS switches with Airolinx 6) can use this for wired access control.

15.4 Troubleshooting

Supplicant failing: Run `/interface dot1x client print statistics` to see if there are request retries or specific failure codes. Ensure the identity/password are correct and method matches what server expects. If certificate based, check validity and that time is correct on router (certs won't work with wrong dates).

Authenticator issues: If using RADIUS, watch the RADIUS server logs. Perhaps the shared secret mismatched (check /radius settings). Also ensure the user is configured in RADIUS with correct service (login vs dot1x, etc.) as needed. If local User Manager, verify the user exists with 802.1x service and profile.

Wireless 802.1X (WPA-Enterprise): Note that for Wi-Fi, Airolinx 6 wireless (especially new WiFiwave2) uses AAA profile which ties into Radius for EAP, rather than this Dot1X menu. The Dot1X menu is mainly for wired and for router as client. For WPA2-Enterprise, configure under WiFi > Security (enterprise EAP modes) and set Radius.

802.1X adds security but adds complexity in key distribution. It's often used in enterprises with a RADIUS server (like Windows NPS or FreeRADIUS) and sometimes with certificates on both sides for high security. Antaira routers can fit into this as either the client device or the enforcer. It's a less common feature for small offices, but essential in larger secure networks.

16 Files

The Files menu shows the file system content of the router. This includes configuration exports, backup files, Airolinx 6 package files, logs (if disk logging), or any uploads (like certificates, scripts, etc.).

From here you can:

- Download files (using Winbox or WebFig, or via FTP).
- Upload new files (e.g., upload a new Airolinx 6 upgrade package .npk file — once uploaded, a reboot will install it).
- Create directories (to organize files). Some RouterBOARDs have separate partitions or disk vs flash — typical small ones just have one flash.
- See backups: if you use "Backup" command, the .backup file appears here, which you can then download off the router for safekeeping.

Be mindful of storage space — some routers have limited memory. Delete unnecessary files (like old package files after upgrade, or old backups you have elsewhere). Do not delete important system files (like the license file if present, typically you won't see it separately though).

17 Log

This displays the log buffer. By default, it's typically a few hundred entries in RAM. You'll see system events, errors, interface up/down notices, etc., depending on what's enabled in Logging (which we covered under System > Logging). You can filter by topics to quickly find entries.

It's read-only here; configuration of what goes into log is via Logging settings. If you need persistent logs, set action to disk or remote.

18 RADIUS

In this menu, you configure the router as a RADIUS client — meaning the router will contact a central RADIUS server for various authentication/authorization/accounting functions. Services that can use RADIUS on Airolinx 6 include:

- PPP (for PPTP/L2TP/SSTP/OpenVPN logins or PPPoE, even Hotspot and DHCP if using user authentication).
- Hotspot — to authenticate hotspot users via RADIUS.
- Login — for router's own user authentication (so you can centralize admin login accounts). This is under System > Users, if set to use RADIUS.
- Dot1X — as discussed, to authenticate LAN port access.

To set up, click "+", enter the RADIUS server IP, secret, and select which services it should apply to (tick login, ppp, hotspot, etc., as needed). You can add multiple RADIUS servers (for redundancy or different services to different servers).

Ensure that your RADIUS server (like FreeRADIUS, Windows NPS, or Antaira User Manager if on another device) is configured with matching secret and has entries for the router (by IP or name). Also ensure firewall isn't blocking UDP 1812/1813 (auth/acct).

Once configured, test by performing the action — e.g., try to login via hotspot or VPN with a user that is defined only on RADIUS, and see if it succeeds. The router's log will show if RADIUS accepted or rejected or if no response (timeout).

You might also configure interim accounting updates if you need usage stats (especially for Hotspot or PPP users). That's in the RADIUS settings (Accounting Interval etc.). The RADIUS server needs matching client config (Antaira often sends attributes like Antaira-Rate-Limit for PPP/Hotspot to dynamically shape speeds, etc.).

19 Tools

The Tools menu provides various utility functions for diagnostics and management:

19.1 Bandwidth Test Server (BTest Server)

The Bandwidth Test Server allows the router to act as a responder for Antaira's Bandwidth Test utility. If enabled, another device running Bandwidth Test (or the Winbox integrated BTest tool) can connect to this router to measure throughput. You can require authentication (so only authorized tests happen).

Use case: You want to measure link speed between two Airolinx 6 devices — you enable BTest server on one, and from the other you run Tools > Bandwidth Test targeting the first. It generates traffic and measures throughput. It's recommended to use UDP for pure throughput, and note it does consume CPU on both sides, possibly showing the max CPU handling rather than actual link if CPU limited. This is mainly a quick check tool, not to be left running normally.

19.2 Bandwidth Test (client)

The Bandwidth Test tool (client) can initiate throughput tests to a BTest server or even to a Antaira which has username/password (it can test to that via btest service). In Winbox, it's under Tools > Bandwidth Test. You input remote IP, credentials (if needed), protocol (TCP/UDP), direction (send, receive, both) and hit Start. It will show current, average throughput and loss if any.

It's useful to check how fast a link can go or if any packet loss at high speeds. But again, be aware that the test itself uses resources and will temporarily consume the link fully (potentially disrupting other traffic during the test).

19.3 Email

Here you set up SMTP settings for the router's ability to send emails (e.g., to email backups or alerts). Configure the server (SMTP server address), port, startTLS or not, user/password if using authenticated SMTP, and sender email address. Once set, you can use /tool e-mail send command in scripts or terminal to send messages, or features like Netwatch or logging can email events.

Test by sending a manual email: /tool e-mail send to="your@address.com" subject="Test" body="hello". Check email arrives. If not, check log for email send error (maybe authentication or TLS needed adjustments).

19.4 Flood Ping

Flood Ping is a rapid ping used to test packet loss and jitter under load. It sends pings as fast as possible (or a set count as fast as possible). In CLI, ping has a "flood" option. In Winbox, Tools > Flood Ping will continuously send until stopped. Use with caution because it's essentially generating a lot of ICMP traffic (which can itself cause high CPU or saturate the link).

If you suspect packet loss on a link, flood ping to see if any drops (the tool will count percent lost). It's a quick way to stress test connectivity at ICMP level.

19.5 Graphing

Antaira has an internal graphing feature that can produce simple RRD graphs for interface traffic, CPU, etc., accessible via the router's web interface. In Tools > Graphing, you can enable graphing for resources or interfaces and specify which IPs can retrieve the graphs (you don't want everyone to see them).

When enabled, pointing a web browser to http://router_ip/graphs/ will show the graphs. They update on a schedule (like every 5 minutes points). This is a lightweight monitoring if you don't have an external system. But it stores data in router's memory (or disk) and is somewhat limited (good for basic long-term throughput visuals).

19.6 IP Scan

This tool scans an IP range or a subnet to find alive hosts. It sends ARP or ping and lists responding devices with IP, MAC, possibly NetBIOS name. Useful to discover devices on a network (like "what's on 192.168.88.0/24?").

It's basically a simple network scanner. Not stealthy or anything, just a convenience for quickly seeing active addresses and possibly open ports 80/8291 for Antaira, etc (it has columns for some common services if it detects Antaira MAC, etc.).

19.7 MAC Server

Antaira has a MAC-layer connection capability (used by Winbox to connect via MAC address in broadcast domain, or MAC telnet/MAC Winbox protocols). MAC Server settings allow enabling/disabling the ability to connect to the router via MAC (using proprietary MAC Winbox protocol and old Winbox "neighbors"). You can set which interfaces to allow it on.

By default, MAC Winbox and MAC Telnet are enabled on LAN interfaces. For security, it's wise to disable on any untrusted networks (so that only you in LAN can use it, not on WAN). Under MAC Server, you'll see Winbox and Telnet services — can turn them off or restrict interfaces.

There's also MAC Ping and MAC Telnet tools for connecting to other Mikrotiks via MAC if on same broadcast domain. This is helpful if IP isn't set or known.

19.8 Netwatch

Netwatch monitors a given host (by ping) and takes actions when the host goes "down" or "up". You specify an IP to ping, interval, and you can write a Up script and Down script.

Example: Monitor 8.8.8.8 every 1 minute. If down for a cycle (no ping response), run a script that switches default route to backup or sends alert. When it comes back up, run another script to revert or log restoration.

Netwatch is a simpler alternative to using scripts+scheduler for link monitoring. It's fairly straightforward — just be mindful if you put heavy tasks in those scripts, the interval. Also netwatch uses ping so ensure the target replies reliably (monitoring an ISP gateway or a dependable public IP). It's good for basic failover automation.

19.9 Packet Sniffer

This is a built-in tool to capture packets flowing through or to the router. You can specify interface(s), filter by protocol/port, and output. It's like Wireshark but on the router. You can either stream it live or save to file (which can then be downloaded and opened in Wireshark).

Usage: If debugging why something isn't working, run sniffer on relevant interface with filter (e.g., host x.x.x.x) to see if packets are arriving or leaving. Or to capture PPPoE exchange, etc. Be careful on busy interfaces — capture may impact performance if writing to file at high rates. Use streaming to an IP (you can stream to your PC running Wireshark — using Tools > Packet Sniffer > Streaming, put your PC's IP and port UDP 37008, and in Wireshark capture from UDP at that port using Antaira's dissector).

It's extremely useful for deep debugging without a separate hub or mirror port. But remember to turn it off when done (if left capturing, it eats memory/disk).

19.10 Ping

Standard ping tool (ICMP Echo). You can specify count, packet size, source address (if you want to ping from a specific interface/IP), and see results (latency, loss). Extended options allow specifying do-not-fragment, etc. In CLI `ping` has those options, in Winbox it's a dialog.

Basic reachability and latency measure. We all know ping. Use it also to do domain name resolution if needed (ping DNS name and see if it resolves — if not, DNS issue likely).

19.11 Ping Speed

Ping Speed sends not just tiny ICMP but tries sending larger packets as fast as possible, essentially measuring throughput between two Antaira routers (target must be Antaira that replies to this specific test).

It's an older tool and often people just use Bandwidth Test now.

19.12 Profile

The Profile tool shows CPU usage by process. Running it (or in Winbox, opening the Profile window) will list tasks like routing, firewall, networking, etc., with percentages. It's essential for performance troubleshooting — e.g., if CPU is high and you want to know why. If "firewall" is high, your filter rules or FastTrack off might be culprit; if "networking" high, maybe too much traffic overhead; "encrypt/decrypt" high — heavy VPN/IPsec load.

You can use it in CLI with /tool profile. It can also show per-core distribution if you have multi-core (with parameter). For instance, one core may be maxed by single-thread process (like PPPoE input or a single queue), while others idle. That identifies bottlenecks that aren't spread across cores.

19.13 RoMON

RoMON (Router Management Overlay Network) is Antaira's proprietary management network that can tunnel management connections through routers even without IP connectivity. If enabled, each router forms a Layer2 mesh for management. Winbox can connect to a RoMON ID and reach a router behind multiple hops of Antaira even if those hops have no IP routes or if it's an isolated management network.

You can enable RoMON, set an ID (perhaps match router identity), and set a password for security (otherwise, theoretically anyone knowing your RoMON ID could connect if they have L2 access to one of your devices). Then in Winbox, you enable RoMON in neighbors and you'll see devices via RoMON and can connect using RoMON as transport.

It's very useful for large layer2 networks or complex topologies — essentially an automatic management VPN. But if not needed, keep it off for security minimalism.

19.14 Speed Test

Wi-Fi 6 Device OS introduced a new Speed Test tool which is similar to Bandwidth Test but can measure TCP/UDP throughput and ping and CPU in one go, more like an integrated iPerf. Possibly it's what Bandwidth Test is replaced by. It also requires a remote Antaira as server (which might be just an open port on them, or use one Antaira as reflector). The idea is to measure throughput and latency together.

You select remote address (another Antaira), test type (TCP or UDP, number of streams), duration. It will output results in CLI or a small GUI. Good to check link quality, and might give more realistic results with TCP than old btest, including showing jitter, etc. Always ensure production traffic not affected when running these heavy tests.

19.15 Torch

Torch is a real-time traffic monitoring tool for an interface. When you run Torch on an interface, it lists flows: src/dst addresses, ports, protocol, and bandwidth usage for each. It's like a lightweight real-time NetFlow viewer. Very handy to see "who is using my bandwidth right now and to where."

You can filter in Torch by protocol, subnet, port, etc. E.g., torch ether1 with filter dst-port=80 to see all web traffic flows. Or no filter to see everything (but on a busy link, it will update quickly and might group small flows into "other" if too many).

Torch is great for quick diagnostics e.g., if a link saturated, torch the WAN to see top talkers. Or if suspicious activity, see flows and their endpoints. Remember that on switch chips or fast path flows, Torch might not capture all (if hardware offload, CPU may not see it; disabling fasttrack temporarily might be needed to see those flows in torch).

19.16 Traceroute

Standard traceroute tool. By default, uses UDP or ICMP (Airolinx 6 default is UDP ports 33434+ like typical). It'll list hops and timings. If behind NAT or firewall, consider that or use appropriate protocol (there's an option to use ICMP instead of UDP which can help if firewalls drop UDP probes).

It's used to see the path packets take through the network. If it stops at a certain hop, that's often where an issue or filtering occurs.

19.17 Traffic Generator

This advanced tool can generate custom traffic patterns for testing. More elaborate than Bandwidth Test, it allows crafting packets of certain size, rate, even specific contents, and send out of an interface. Good for stress testing or testing how other devices handle certain packet patterns.

You could simulate 1000 UDP streams from fake IPs to test firewall performance, etc. It's scriptable via /tool traffic-generator commands. Typically used by network engineers in labs. Not needed for everyday use.

19.18 Traffic Monitor

This tool can watch an interface's traffic rate and execute a script on threshold crossing. For instance, if traffic on ether1 goes above 10Mbps, do something (like send alert or adjust queue). You define a monitor on interface with threshold and interval. If threshold is exceeded or drops below, it triggers.

It's like netwatch but for traffic levels. Could be used to detect DDoS or heavy usage periods and respond, e.g., automatically enable a second WAN when usage high (just an idea). Again, a niche tool, but nice to have for automation beyond just link up/down.

19.19 WoL (Wake on LAN)

Wake on LAN tool can send the magic packet to wake computers on the LAN. You need the MAC address of the target (and it must be in same broadcast domain or you supply an IP broadcast). Use /tool wol send interface=LAN mac=AA:BB:CC:DD:EE:FF.

This is helpful if you want to remotely boot a turned-off PC. Often combined with port knocking or some trigger (e.g., you text the router via SMS, router runs wol to turn on a server).

It only works if the target PC is just soft-off and has WoL enabled in NIC and BIOS, and it's listening on that MAC. The router can reach it via broadcast or directed subnet broadcast.

That concludes the Tools overview. Each of these can greatly aid in maintaining and troubleshooting your network. Familiarity with them means you have a solid toolkit at your disposal directly on the router, often negating the need for external PC tools.