perplexity

Understanding `ip addr`, `ip link`, and `ip tuntap` in Linux

This section provides a clear explanation of the `ip addr`, `ip link`, and `ip tuntap` commands, their purposes, and how they interact within Linux networking.

`ip addr`

- **Purpose:**
  `ip addr` is used to display and manage IP addresses assigned to network interfaces.
- **Usage:**
  - List all assigned IP addresses:

    ```
    ip addr
    ```

    or

    ```
    ip addr show
    ```

  - Show addresses for a specific interface:

    ```
    ip addr show dev eth0
    ```

  - Add an IP address to an interface:

    ```
    sudo ip addr add 192.168.1.100/24 dev eth0
    ```

  - Remove an IP address:

    ```
    sudo ip addr del 192.168.1.100/24 dev eth0
    ```

  - List IPv4 or IPv6 addresses only:

    ```
    ip -4 addr
    ip -6 addr
    ```

- **What it shows:**
  Interface names, their IP addresses (IPv4 and/or IPv6), broadcast addresses, and status[1] [2].

`ip link`

- **Purpose:**
  `ip link` is used to display and manage network interfaces (links), focusing on layer 2 (Ethernet, loopback, etc.), not IP addresses.

- **Usage:**
  - Show all network interfaces and their properties:

    ```
    ip link
    ```

  - Bring an interface up or down:

    ```
    sudo ip link set dev eth0 up
    sudo ip link set dev eth0 down
    ```

  - Set interface parameters (e.g., MTU):

    ```
    sudo ip link set dev eth0 mtu 1400
    ```

- **What it shows:**
  Interface index, name, MAC address, state (UP/DOWN), MTU, and flags (BROADCAST, MULTICAST, etc.)[3].

Summary Table: `ip addr` **vs** `ip link`

Command	Focus	Example Output	Typical Use Cases
ip addr	IP addresses (Layer 3)	Shows IPs, netmasks, broadcast, etc.	Assign/remove/view IP addresses
ip link	Network devices (Layer 2)	Shows MAC, state, MTU, flags	Bring interfaces up/down, set MTU

`ip tuntap`

- **Purpose:**
  `ip tuntap` manages TUN/TAP virtual network interfaces, which provide a way for user-space programs to interact with network packets as if they were network devices.
- **Types:**
  - **TUN:** Simulates a point-to-point device (layer 3), used for routing IP packets (e.g., VPN tunnels).
  - **TAP:** Simulates an Ethernet device (layer 2), used for bridging Ethernet frames (e.g., virtual switches, VMs).
- **Usage:**
  - Create a TAP device:

    ```
    sudo ip tuntap add mode tap dev tap0
    ```

  - Create a TUN device:

    ```
    sudo ip tuntap add mode tun dev tun0
    ```

  - Set owner/group for the device:

```
    sudo ip tuntap add mode tap dev tap0 user youruser group yourgroup
    ```

  - Delete a device:

    ```
    sudo ip tuntap del mode tap dev tap0
    ```

- **How it works:**
  The TUN/TAP kernel driver exposes `/dev/net/tun`. When you create a TUN/TAP device (via `ip tuntap` or programmatically), a new network interface appears (e.g., `tun0`, `tap0`). User-space programs can read/write packets to this interface, enabling use cases like VPNs, containers, and network emulation[4] [5] [6].

Key Differences and Use Cases

- `ip addr`: Focuses on IP addresses and network layer configuration.
- `ip link`: Focuses on the physical or virtual network device itself and its properties.
- `ip tuntap`: Used for creating and managing virtual network interfaces for advanced networking scenarios (VPNs, bridges, virtual machines, network namespaces).

Example Workflow

1. **Create a TAP device:**

   ```
   sudo ip tuntap add mode tap dev tap0
   ```

2. **Bring up the device:**

   ```
   sudo ip link set dev tap0 up
   ```

3. **Assign an IP address:**

   ```
   sudo ip addr add 10.1.0.1/24 dev tap0
   ```

This sequence creates a virtual Ethernet interface (`tap0`), brings it online, and assigns it an IP address, making it ready for use in bridging, tunneling, or virtualization scenarios[5] [6].

In summary:

- Use `ip addr` to manage IP addresses.
- Use `ip link` to manage network device properties and state.
- Use `ip tuntap` to create and manage virtual TUN/TAP interfaces for user-space networking applications and advanced networking setups[1] [2] [3] [4] [5] [6].

⁂

1. https://phoenixnap.com/kb/linux-ip-command-examples

2. https://hostman.com/tutorials/how-to-use-the-ip-command-in-linux/
3. https://unix.stackexchange.com/questions/297064/what-is-the-function-of-ip-link-command
4. https://docs.kernel.org/networking/tuntap.html
5. https://stackoverflow.com/questions/15626088/tap-interfaces-and-dev-net-tun-device-using-ip-tuntap-command
6. https://blog.faizahmed.in/tuntap-devices-using-open-vswitch