Linux Kernel Management of Ethernet Devices and Network Interfaces

Linux kernel supports efficient communication over Ethernet using its network stack consisting of network protocol layers, user-space interfaces, and device drivers. Its kernel controls the devices on the network using the network subsystem to route data efficiently through the hardware as well as the software layers.

1. Detection and Initialization of Ethernet Devices

The kernel detects Ethernet interfaces at boot time by scanning the PCI, USB, or other hardware buses. The corresponding network driver is then loaded to bind to the corresponding Ethernet hardware.

- The udev system assigns network interfaces logical names (e.g., eth0, eth1).

- The NIC (Network Interface Card) kernel driver is loaded dynamically (modprobe [driver\_name]).

- It is managed through /sys/class/net/ and /proc/net/dev.

2. Network Interface Configuration

After an Ethernet device has been initialized, it needs to be configured to exchange information. This includes the assignment of an IP address, MAC address, and network parameters.  
  
- ip link show shows available interfaces.  
- ip command and ifconfig (obsolete) can configure IP addresses.  
- The device's state changes between DOWN, UP, and RUNNING.

3. Packet Handling in the Kernel

The kernel handles network packets at several layers:  
- Device Driver Layer: Talks directly to the NIC hardware.  
- Network Layer: Manages IP routing, packet forwarding, and filtering.  
- Transport Layer: Has support for TCP, UDP, and others.  
- User-Space Interaction: Programs communicate through sockets (socket() system call).

4. Data Transmission and Reception

- TX Path (Transmit): The kernel buffers packets for transmission, sends them over the appropriate interface, and passes them on to the driver to send over Ethernet.  
- RX Path (Receive): The NIC driver receives the arriving packets, directs them to the network stack, and delivers them to the target application.  
- Packet filtering by iptables and nftables can be done at different levels.

5. Kernel Features for Ethernet Devices

- Multi-driver support: Open-source drivers for Realtek, Broadcom, and Intel NICs.  
- Virtual Network Interfaces: They include veth, bridge, and bonding interfaces.  
- Performance Optimizations: - Parallel processing with Receive-Side Scaling (RSS). - Generic Receive Offload (GRO) and TCP Segmentation Offload (TSO). - eBPF/XDP for fast packet processing.

6. Debugging and Monitoring Tools

- dmesg displays kernel messages related to NIC initialization.

- /proc/net/dev provides interface data.

- Wireshark and tcpdump examine traffic.

- ethtool reads and modifies NIC configuration.