Network Layer Overview

Services and Protocols

- To transport **segments** from the **sending host**, to the **receiving host** the following happens:

  1. The **sender encapsulates segments** into **datagrams** and passes them to the **link layer**.
  2. The **receiver delivers segments** to the **transport layer protocol**.
  3. A **router** is a piece of **network hardware** than manages **traffic between networks**.
     - Routers work by examining the headers in **IP datagrams (Packets)**, and move the datagrams from **input ports** to **output ports**; with the goal of transfering datagrams along the end-end path.
     - Routers work a the **Network Layer (Layer 3)**, and also use layers 1 and 2 to facilitate the data transfer.
     - Routers us **Internet Protocol Addresses (IP Address)** to identify networks / hosts.

Key Network-Layer Functions

- One key network-layer function is **forwarding**, **forwarding** involves **moving packets** from a **rounter's input link** to the appropriate **output link**.

- Another key network-layer function is **routing**, **routing** involves **determining the route taken by packets** from the **source** to the **destination**.

  - There are many routing algorithms that can be used the achieve this.

The Data Plane vs The Control Plane

- The **data plane** is a **local, per-router function** that **determines how packets** arriving on a router's input port **is forwarded to router's output port**.

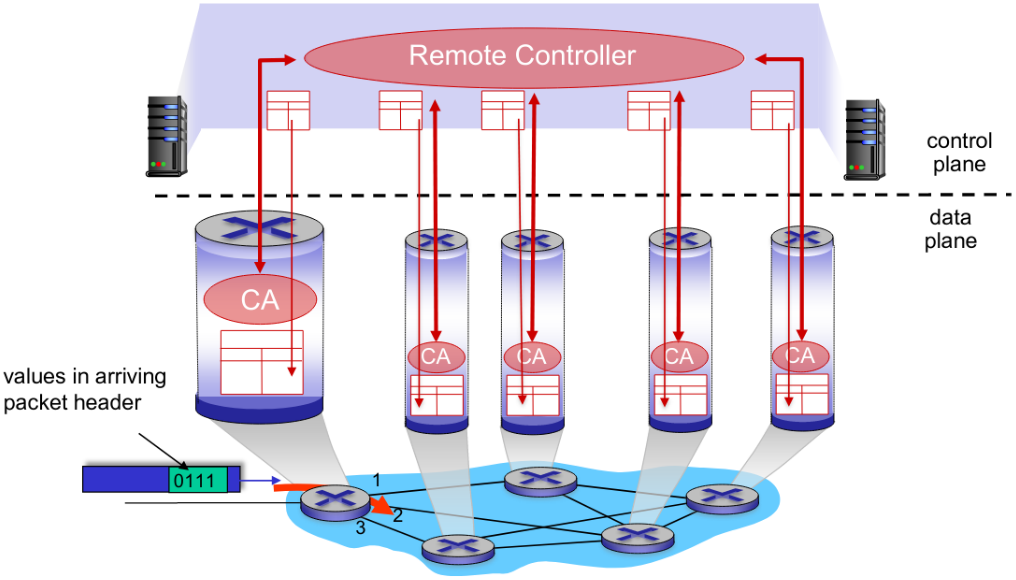
- The **control plane** is a **network-wide** function, that **determines how packets** are **routed amongst routers** along end-end paths from **source host** to **destination host**.

  - There are two control-plane approaches:
    1. **Traditional routing algorithms** that are implemented in routers.
    2. **Software-defined networking (SDN)** that is implemented in remote servers.

Per-Router Control PLane Software-Defined Networking (SDN) Control Plane

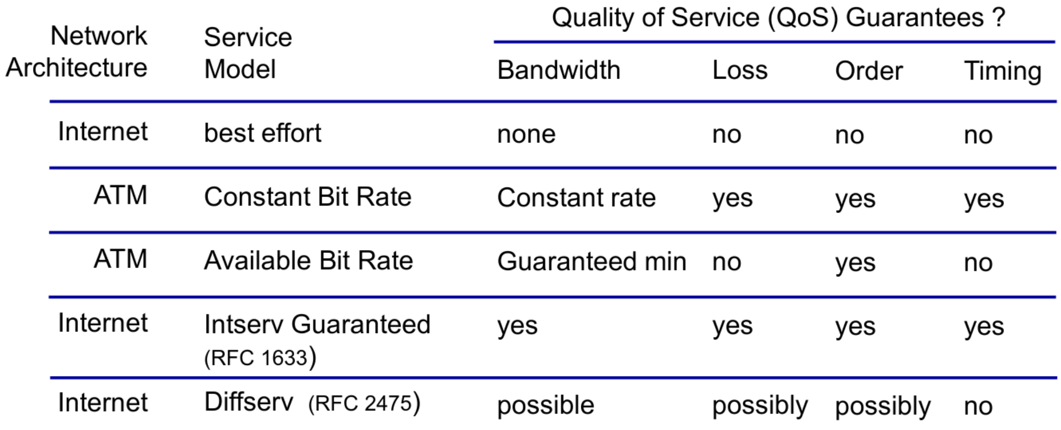
- **Per-Router control plane** consits of a **routing algorithm** in **every router** that interacts with the **control plane**. Each router determines where to route the **packets**.

- **SDN** is composed of **remote controller computers**, that **install forwarding tables** in router. The routers then use these tables to forwards **packets**.



Network Service Models

- Internet service models:



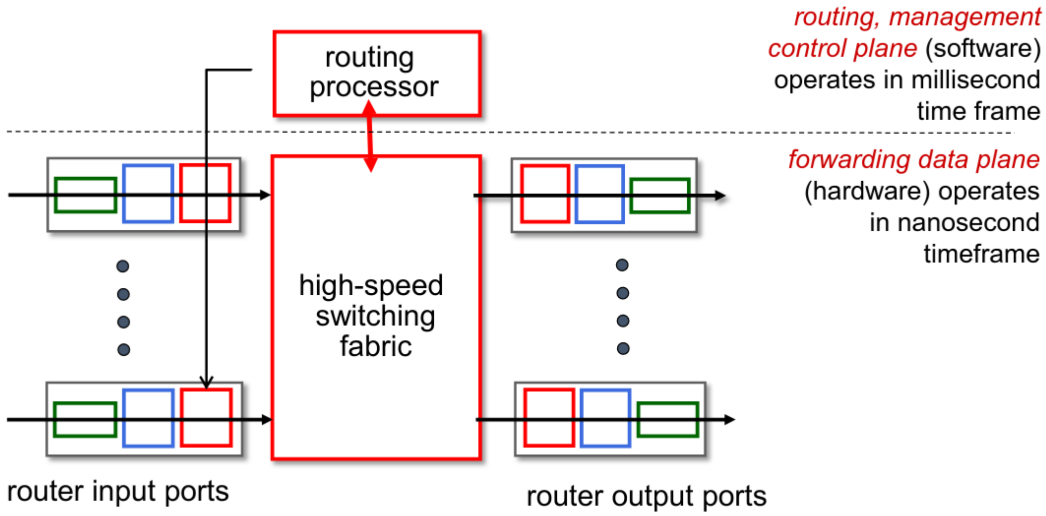
Network Architecture	Service Model	Quality of Service (QoS) Guarantees ? Bandwidth	Loss	Order	Timing
Internet	best effort	none	no	no	no
ATM	Constant Bit Rate	Constant rate	yes	yes	yes
ATM	Available Bit Rate	Guaranteed min	no	yes	no
Internet	Intserv Guaranteed (RFC 1633)	yes	yes	yes	yes
Internet	Diffserv (RFC 2475)	possible	possibly	possibly	no

- Though the **best effor service model** may not provide any guarantees, it allowed the internet to be widely deployed, and adopted.

Router Architecture Overview

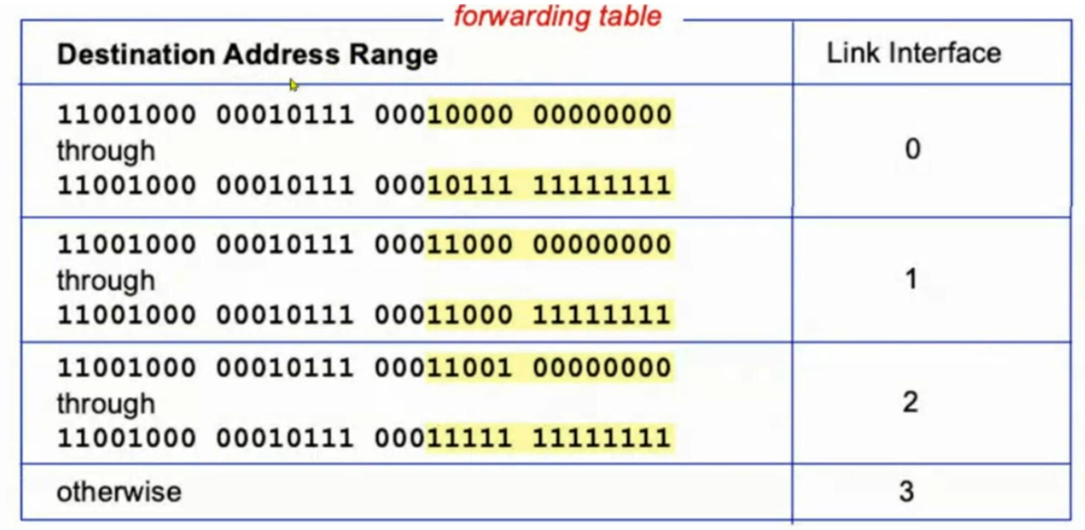
Routers

- A **router** is a **networking device** that **forwards** and **router data packets** between **networks**.
- **Routers** have **input ports** and **output ports**, to **receive** and **forward** packets respectively.



- The green boxes represents the **physical layer**, the blue boxes represent the **link layer**, and the red boxes represent the **network layer**.
- The first red box continas a queue of packets that need to be forwarded, and the lookup table, that map headers to ports.

- **Destination-based forwarding** is forwarding based only on the **destination IP Address** (traditional).
- **Generalized forwarding** is forwarding based on **any set of header field values**.
- The following is an example of a lookup table:



forwarding table

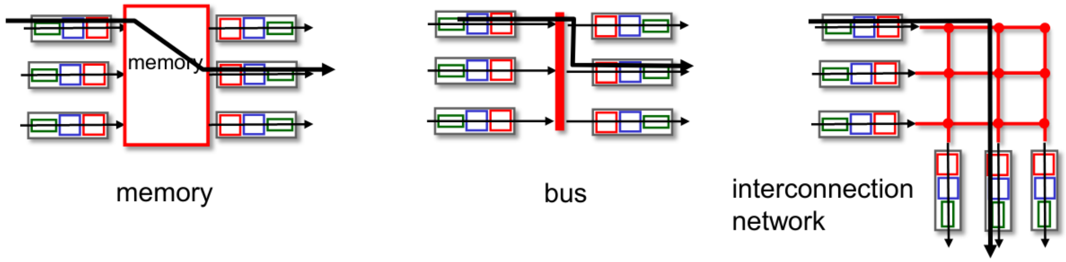
Destination Address Range	Link Interface
11001000 00010111 00010000 00000000 through 11001000 00010111 00010111 11111111	0
11001000 00010111 00011000 00000000 through 11001000 00010111 00011000 11111111	1
11001000 00010111 00011001 00000000 through 11001000 00010111 00011111 11111111	2
otherwise	3

- To determine which interface a an IP address should be mapped to, you see what address range has the longest prefix that matches the IP address of the packet that is being router.

Switching Fabrics

- **Switching fabrics** are responsible for transfering the **packet** from the **input link** to the appropriate **output link**.

- The **switching rate** is the rate at which **packets can be trasnferred** from **inputs** to **outputs**.
- There are **three main types of switching fabrics**:



- With **memory switching**, the packets are copied to the **system's memory**. This limits the switching rate to the **memories bandwidth**. This type of switching is directly under the control of the **CPU**.
- With **bus switching**, the packets are delivered from the **input port's memory**, to the **output port's memory** directly. The switching speed is limited to the **speed of the bus** (which is much faster than memory switching).
- With **interconnnection network switching**, is similar to the bus, except we can **transfer several packets in parallel**.
- We can have **seveal planes** of **interconnection network switching** that run in parallel to allow for scaling.

Input Port Functions

- There are three functions of an **input port**:
  1. Receives the bits.
  2. Interperates the bits.
  3. Forward the packet to the switching fabric.
- There are two types of **forwarding** in **decentralized switching**:
  1. **Destination-based forwarding** — Packets are forwarded based on their destination IP Address.
  2. **Generalized forwarding** — Packets are forwarded based on any set of header field values.
- If the **switching fabric** is **slower** than the **input ports conbined**, then **queueing may occur** at the input queues. The queueing may lead to **delays, and packet loss**.
- **Head-of-Line (HOL) blocking** is when the **packet** at the **front of the queue** prevents the **queue from moving** (due to congestion).

Output Port Functions

- There are three functions of an **output port**:
  1. Receive the packet from the switching fabric.
  2. Send the bits.
  3. Send a line termination.
- **Buffering** is required when **packets** arrive from the **switching fabric** faster than the **link's transmission rate**.

- The **drop policy** is a policy to determine **which packets to drop** if the **buffer is full**.
- **Congestion occurs when the output link is slower than the output ports**.

Determining Buffer Sizes

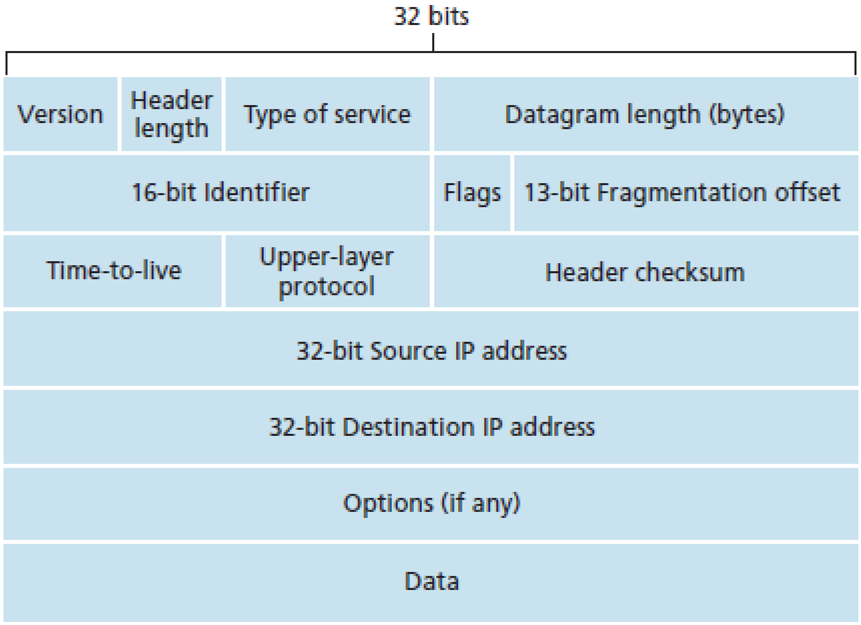
- The **RFC 3439** rule of thumb is that the **average buffer size** should be equal to RTT × LinkCapacity.
- There is such a thing as **too much buffering**; this can increase delays (particularly in home routers).
  - **Long RTTs** lead to **poor performance** in **real-time allpcations**.
- When the **buffer is full**, there are two ways to determine the **packets to drop** when more arrive:
  1. **Tail drop** involves dropping the arriving packets.
  2. **Priority drop** involves dropping packets on a priority basis.
-

Packet Scheduling

- There are several ways to **determine the way packets are scheduled**:
  1. **First Come First Serve (FCFS) Scheduling** schedules packets in the **order they arrive**.
  2. **Priority Scheduling** schedules packets **based on their classification** (classifications can be determined by header fields); all higher classified queues get send first.
  3. **Round Robin (RR) Scheduling** schedules packets **based on their classification**, but cycles between the different classification queues.
  4. **Weighted Fair Queueing (WFQ)** is a **round robin** scheduling algorithm, that assigns each class a weight, and the weight determines the amount of time is spent on each queue.

The Internet Protocol

Internet Protocol Datagrams



- The **maximum length** of a **datagram is 65536 bytes**. However the typical size is around **1500 bytes or less**.

IP Address

- An **IP Address** is a **32-bit identifier** associate with each **network host** or **router interface**.
  - An **interface** is a **connection** between a **host/router** and a **physical link**.
  - **Routers** typically have multiple interfaces, and **hosts** typically have one or two interfaces.

Subnetworks

Subnetworks (Subnets)

- A **subnetwork** is a **local partition** of an **Internet Protocol network**.
- **Subnets** are typically defined as a group of **device interfaces** that can **physically reach each other** without **passinging through an intervening router**.
- **Devices** in the **same subnet** have **common high order bits** in their **IP Addresses**, and the **low order bits differ between hosts**.
  - For example, the following addresses are an example of a subnet: 1.0.23.1, 1.0.23.5, 1.0.23.10
- A **subnet mask** is a **32-bit number** created by setting **host bits to 0**, and **network bits to 1**. All bits set to 0 can change, and the addresses will still be a part of the subnet.
  - There is a short hand for this, you put a slash and then the number of bits that are set to 1 starting from the left. For example /24 = 11111111.11111111.11111111.00000000

  - You can also express subnet masks as their decimal interpretation: 11111111.11111111.11111111.00000000 = 255.255.255.0

- There are **three main classes** of networks:
  1. **Class A** networks have a subnet mask of **255.0.0.0**.
  2. **Class B** networks have a subnet mask of **255.255.0.0**.
  3. **Class C** networks have a subnet mask of **255.255.255.0**.
- There is also a **Class D and Class E**, but they are reserved for special purposes, and research. They are not available for network hosts.

Gateways

- Each **inteface** on a **router** has a **unique IP Address**.
- **Routers** that provide an **interface** between **two distinct subnetworks** are known as a **gateway**.
  - Gateways are a router that interfaces multiple subnets, hence why each inteface gets a unique IP Address that must be within the subnet it is interfacing with.
  - When at least two gateways are directly connected, there is a new subnet between them.

IP Adressing — CIDR

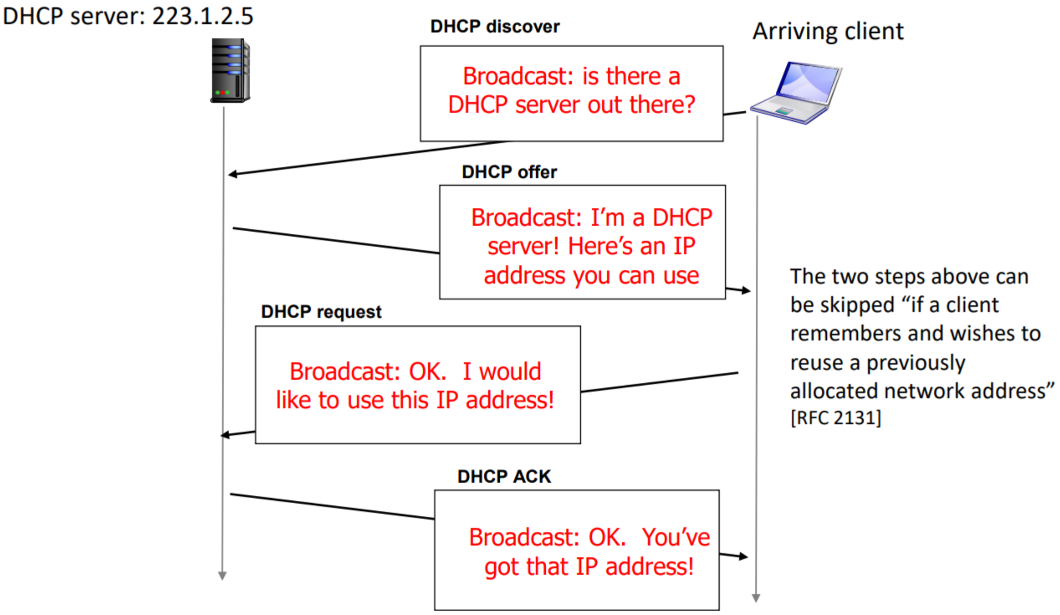
- **Classless InterDomain Routing (CIDR)** (aka supernetting) is a **method of assigning Internet Protocol (IP) adresses**.
- **CIDR** consists of the following address format: **a.b.c.d/x** where a, b, c, d are n-bit numbers numbers, and x is the number of bits in the subnet portion of the address.
  - An example would be **200.23.16.0/23**, the first 23 bits identify the subnetwork, and the last 9 bits identify the host within the subnetwork.
- There are to ways a **host** gets an **IP Address assigned**:
  1. **A Hard-coded address** that is assigned by a sysadmin in a configuration file.
  2. **The Dynamic Host Configuration Protocol (DHCP)** which allows devices the get IP Adress assignments dynamically from a server.
- DHCP is particularly useful for devices connecting to routers over **Wi-Fi**.

Dynamic Host Configuration Protocol

- The goal of **DHCP** is to allow **hosts** to **dynamically obtain IP Adresses** from a **network server** when it ”joins” the network.
- **DHCP ”leases”** out **IP Adresses** to devices on the **network** for a **specific period of time**. This allows for addresses to be **reused** if a device leaves the network.
- The process is **seamless** so hosts can easily come and go.
- The following is how a device **typically obtains an IP Address over DHCP**:
  1. The **host** broadcasts a **DHCP discover** message (only if the host does not know the address of the DHCP server).
  2. The **DHCP server** responds with a **DHCP offer** message (only if the host sends a discover message).

3. The **host** requests the **IP Address** that the **server** offered with a **DHCP request** message.
4. The **DHCP server** sends a **DHCP ack** message to the **address** that has been **leased to the host**.

- Typically, the **DHCP server** will be **co-located in the router**, serving all **subnets** the **router is attached to**.
- When **DHCP** messages are sent, the **server** typically uses port **67**, and the host typically uses port **68**.


DHCP server: 223.1.2.5
DHCP discover
Arriving client
Broadcast: is there a DHCP server out there?
DHCP offer
Broadcast: I'm a DHCP server! Here's an IP address you can use
The two steps above can be skipped "if a client remembers and wishes to reuse a previously allocated network address" [RFC 2131]
DHCP request
Broadcast: OK. I would like to use this IP address!
DHCP ACK
Broadcast: OK. You've got that IP address!

- **DHCP** can return more than just an **IP Address**, it can also return:
  1. The **address** of the **first-hop** router for a **client**.
  2. The **name** and **IP** of a **DNS server**.
  3. The **network mask** (including network vs host portion of an address).

Internet Service Providers

- The **Internet Corporation for Assigned Names and Numbers (ICANN)** is responsible for **assigning IP address blocks** to **Internet Service Providers (ISPs)**.
- The **ISPs** can then assign them to their clients.

Network Address Translation (NAT)

- All **devices** in a **local network** share **one public IPv4 address** as far as the outside world is concerned (the address of the first-hop router).
- **Datagrams** with a **source** or **destination** within this network have **10.0.0.0/24** or **192.168.0.0/24** as the **source and destination address** (usually).
- All **datagrams** that are **leaaving the local network** have the **same source NAT IP Address**, but different **source port numbers**.
- All **devices** in the **local network** have a **private 32-bit IP Address** that can only be used in the **local network**.

- The advantage that is provided by allowing **one public IP Address** to map to **several private IP Addresses** is:

  1. The ISP only needs to assigne one address to the customer for all devices on the local network.
  2. You can change the address of hosts in a local network without notifying the outside world.
  3. You can change ISP providers without changing addresses of devices in the local network.
  4. Enhanced security; devices inside the local network are not directly addressable or visible by the outside world, instead they are proxied.

- In 2011, ICANN allocated the last remaining available IPv4 addresses to an ISP, public and private IP Adresses allow us to remain on IPv4 for longer, because the same addresses are reused in several local networks.

NAT Implementation

- The NAT router must transparently do the following:

  1. **Outgoing Datagrams** — The source IP and port of every datagram must be replaced by the NAT IP and a new port which identifies the host in the local network. Remote hosts will respond using the new NAT IP address and port. The mapping will be placed in a NAT table.
  2. **Incoming Datagrams** — The destination IP and port of every datagram must be replaced with the with the original local IP and port, which were previously stored in a NAT table.

The NAT Controversy

- Nat has been controversial:

  - **Routers** should only process **up to layer 3** (ports are layer 4).
  - The **address shortage** should be solved by **IPv6**, not **NAT**.
  - It **Violates** the **end-to-end argument** (port # manipulation).
  - It doesnt allow **clients** to **connect** to **servers behind NAT**.

- Even with all of th controversy, NAT is here to stay. It is extensively used in home and institutional networks, and 4G/5G cellular networks.

Middlebox Functions

- A **Middlebox** is an **intermediary box** that performs **functions apart from the noraml, standard functions** of an **IP router** on the **data path** between a **source host** and a **destination host**.

- Some exampls of middleboxes are firewalls, NAT, load balancers, CDN cache, etc.

- Initially, **middleboxes were proprietary** hardware solutions.

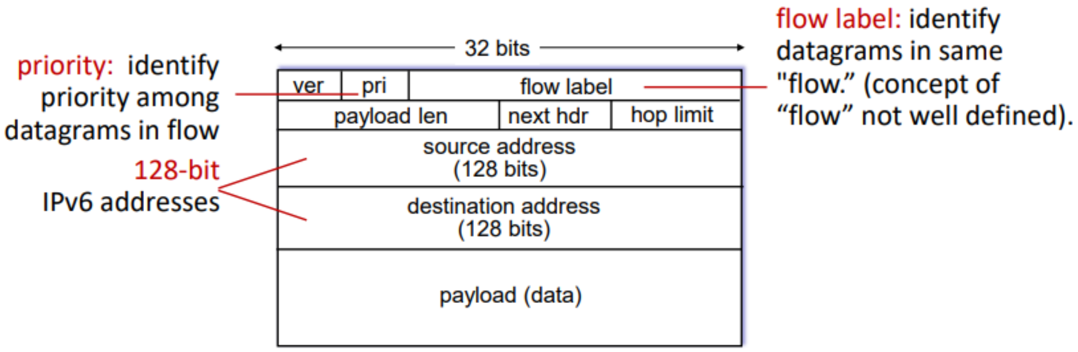
- Over time, things moved towards **open solutions** with **programmable local actions** via match+action.

- **Software-Defined Networking (SDN)** allows for **logically centralized control** and **configuration managment**, often in the **"cloud"**.

Internet Protocol version 6 (IPv6)

IPv6 Format

- The **initial motivation** was the fact that **IPv4 addresses** would be (at the time they were not completely allocated, they are now) **completely allocated**.
- Unlike IPv4, the IPv6 format does not have **a checksum, options, or fragmentation**.
- A major problem is that not all routers can be upgraded simultaneously.
- To allow **routers** that **only support IPv4, to handle IPv6 datagrams**, we can use **tunneling**: The **IPv6 datagram** is carried as a **payload** in an **IPv4 datagram**. This is essentially, a packet within a packet.
- Tunneling is used extensively in other contexts such as 4G/5G.


32 bits
ver
pri
flow label
payload len
next hdr
hop limit
source address (128 bits)
destination address (128 bits)
payload (data)
priority: identify priority among datagrams in flow
128-bit IPv6 addresses
flow label: identify datagrams in same "flow." (concept of "flow" not well defined).