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

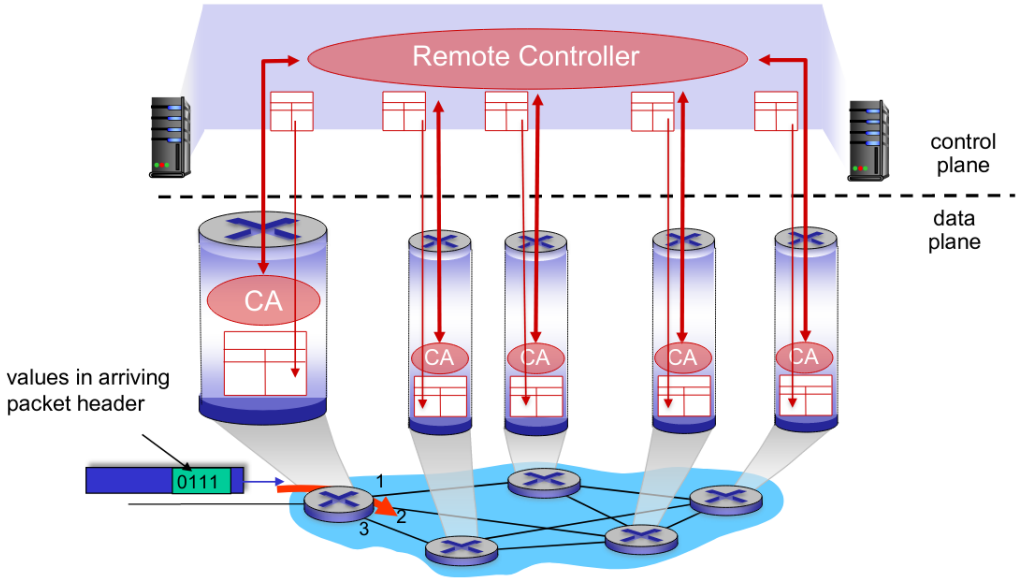
- One key network-layer function is **forwarding**, **forwarding** involves **moving packets** from a **rounter's input link** to the appropriate **output link**.
- Another key network-layer function is **routing**, **routing** involves **determining the route taken by packets** from the **source** to 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:

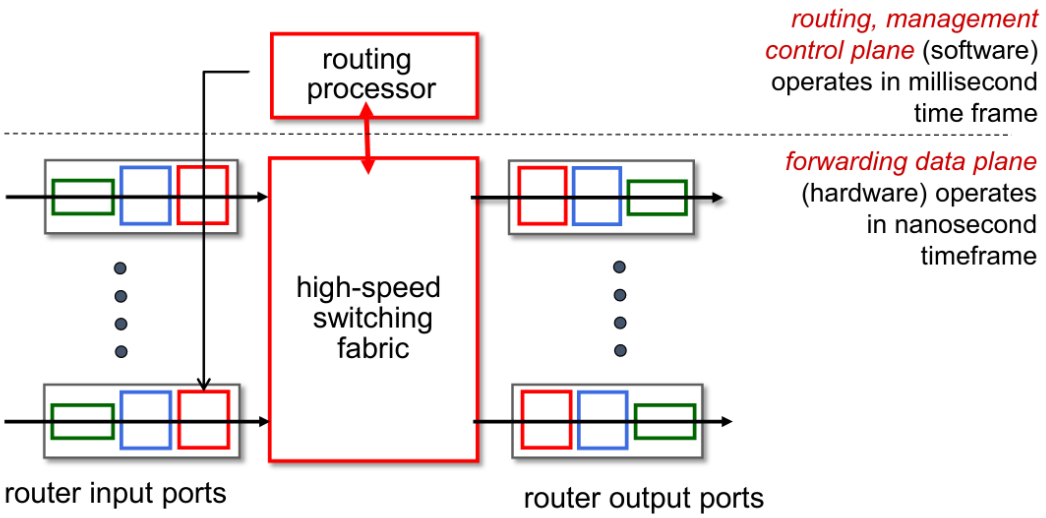
Network Architecture	Service Model	Quality of Service (QoS) Guarantees ?			
		Bandwidth	Loss	Order	Timing
Internet	best effort	none	no	no	no
ATM	Constant Bit Rate	Constant rate	yes	yes	yes
ATM	Available Bit Rate	Guaranteed min	no	yes	no
Internet	Intserv Guaranteed (RFC 1633)	yes	yes	yes	yes
Internet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

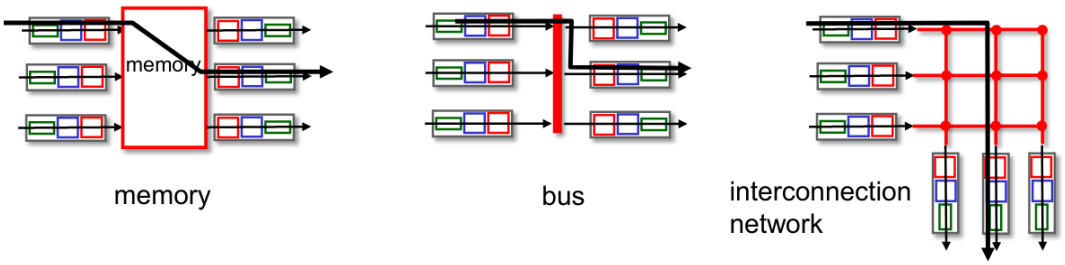
Destination Address Range	Link Interface
11001000 00010111 00010000 00000000 through 11001000 00010111 00010111 11111111	0
11001000 00010111 00011000 00000000 through 11001000 00010111 00011000 11111111	1
11001000 00010111 00011001 00000000 through 11001000 00010111 00011111 11111111	2
otherwise	3

  - 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

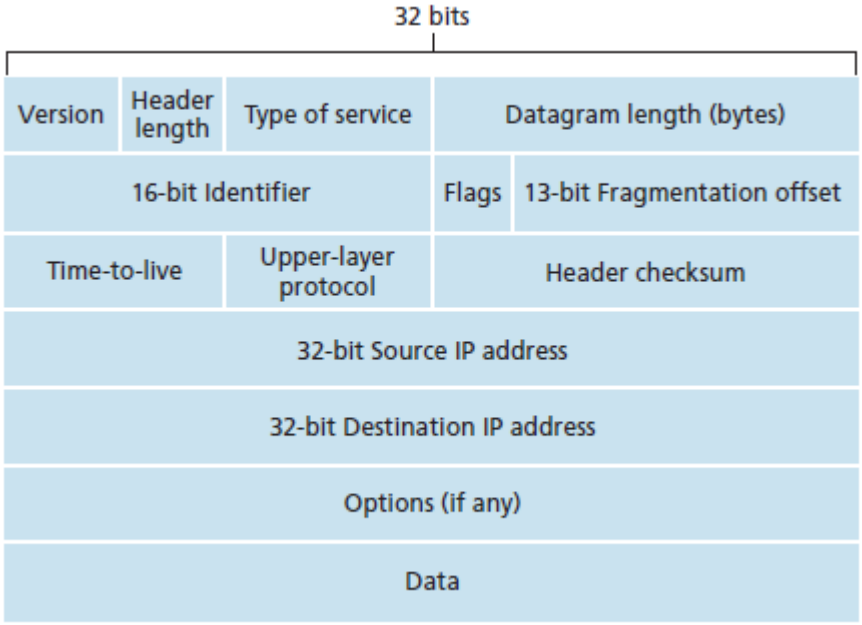
- The **RFC 3439** rule of thumb is that the **average buffer size** should be equal to RTT $\times$ LinkCapacity.
- There is such a thing as **too much buffering**; this can increase delays (particularly in home routers).
  - **Long RTTs** lead to **poor performance** in **real-time allpcations**.
- 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 (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.