

Network Layer

Study-Ready Notes

Compiled by Andrew Photinakis

November 14, 2025

Contents

1	4.1.1 Forwarding and Routing: The Data and Control Planes	2
1.1	1. Concept Overview: The Two Core Functions	2
1.2	Forwarding: The "Data Plane"	2
1.3	Routing (Control Plane Function)	3
1.4	4.1.1.1 The Forwarding Table	3
1.5	4.1.1.2 Approaches to the Control Plane	4
1.6	4.1.1 Summary	4
2	4.1.2 Network Service Model	5
2.1	1. Concept Overview: The "Contract" of the Network Layer	5
2.2	2. Technical Mechanism: A "Menu" of Possible Services	5
2.3	3. Practical and Intuitive View: The Internet's <i>Actual</i> Service	5
2.4	4. Terminology Clarification: Switches vs. Routers	6
2.5	4.1.2 Summary	6
3	4.1 Overview of Network Layer	6
3.1	Core Role and Architectural Placement	6
3.2	Planes	7
3.2.1	Data Plane	7
3.2.2	Control Plane	7
4	4.2 What's Inside a Router?	7
4.1	Router Architecture	7
4.2	Template Example	8
5	4.3 The Internet Protocol (IP): IPv4, Addressing, IPv6, and More	8
5.1	Overview	8
5.1.1	IP Protocol	8

6	4.3.1 IPv4 Datagram Format	8
6.1	Concept Overview	8
6.2	Technical Mechanism: The Datagram Format	9
6.3	Template Example	11
7	4.3.2 IPv4 Addressing	11
7.1	Concept Overview: Interfaces and Subnets	11
7.2	Technical Mechanism: The IP Subnet	11
7.3	Concept: CIDR (Classless Inter-Domain Routing)	12
7.4	Obtaining and Managing IP Addresses	13
7.5	Template Example	14

Chapter 4: The Network Layer — Data Plane

1 4.1.1 Forwarding and Routing: The Data and Control Planes

Two key functions of the network layer, **forwarding** and **routing**, map directly to the **data plane** and **control plane**, respectively.

1.1 1. Concept Overview: The Two Core Functions

Welcome. In our last lecture, we completed our study of the transport layer, which provides a *logical* communication service between *processes* running on different hosts. Now, we move down the stack to examine the network layer, which provides the underlying *host-to-host* communication service that the transport layer relies on.

The network layer's primary role seems simple: to move packets, called **datagrams**, from a sending host to a receiving host. However, to accomplish this, the network layer must perform two distinct and critical functions: **forwarding** and **routing**.

1. Forwarding

- Is a *local* action.
- The process of taking a packet that has arrived on one of a router's input links and moving it to the appropriate output link.

2. Routing (Control Plane Function)

- Is a *network-wide* process that determines the end-to-end paths that packets take from a source host to a destination host.

1.2 Forwarding: The "Data Plane"

1. Definition

- Forwarding refers to the router-local action of transferring a packet from an input link interface to the appropriate output link interface.
- Forwarding is the primary function of the network layer's **data plane**, which includes per-router operations performed on a packet as it moves through the router.

2. Mechanism

- A packet arrives at a router's input link.
- Router examines one or more fields in the packet's header.
- Uses header values to index into its forwarding table.

- Value found in the forwarding table entry indicates the router's output link interface to which the packet should be forwarded.

3. Timescale & Implementation

- Forwarding is a fast (nanosecond) action that must be performed for every packet.
- Therefore, it is implemented in hardware.

4. Analogy

- Forwarding is like navigating a single highway interchange or roundabout: the driver makes a local, split-second decision based on signs.

1.3 Routing (Control Plane Function)

1. Definition

- Routing refers to the network-wide process of calculating and determining the end-to-end paths that datagrams follow from source to destination.
- It is the primary function of the **control plane**, which governs network-wide routing logic.

2. Mechanism

- Routing is accomplished by routing algorithms that calculate paths for datagrams.
- Routing decisions are installed in the forwarding tables of routers.

3. Timescale & Implementation

- Routing computations occur on much longer timescales (seconds to minutes).
- Typically implemented in software due to their complexity.

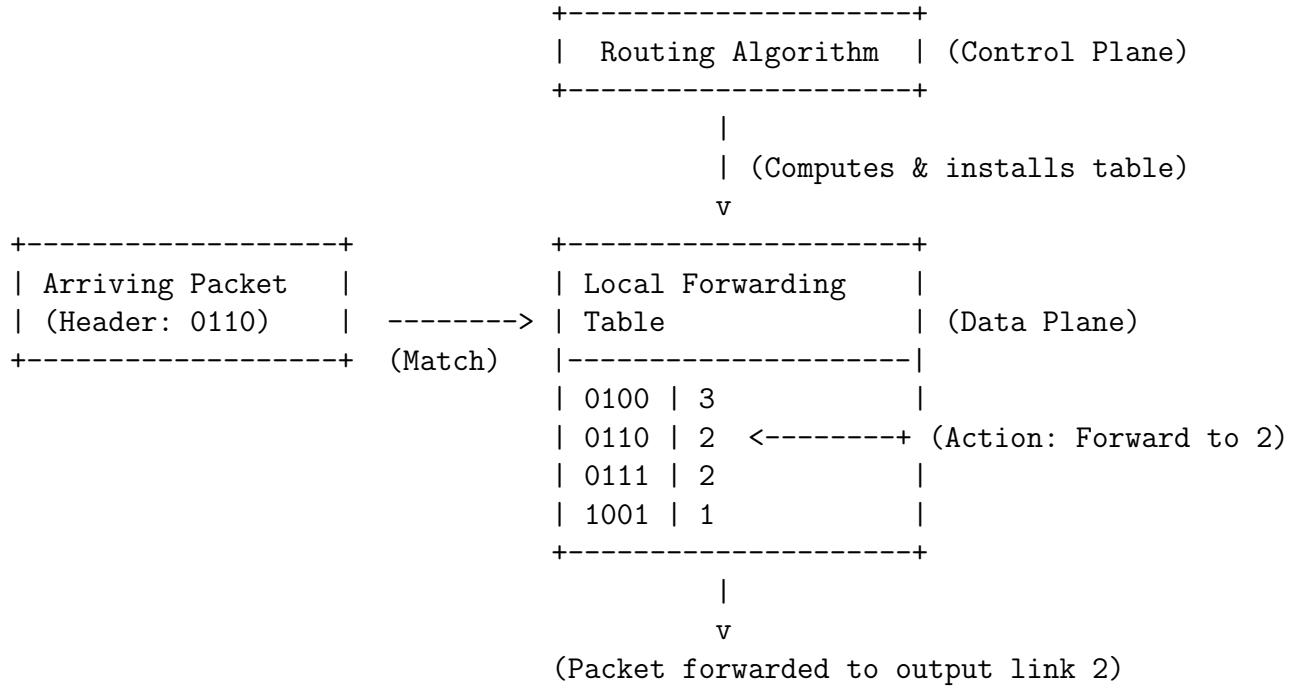
4. Analogy

- Routing is like planning an entire trip from Pennsylvania to Florida: the routing algorithm (driver) consults a map to select the best end-to-end path.
- Each router on the path forwards packets according to the plan.

1.4 4.1.1.1 The Forwarding Table

The routing (control plane) controls forwarding (data plane) via the **forwarding table**.

- **Function:** Router examines header fields of incoming packets, indexes into the forwarding table, and forwards the packet according to the table entry.
- **Origin:** Entries are computed and installed by the control plane. Changes in topology or costs trigger updates.



1.5 4.1.1.2 Approaches to the Control Plane

Approach 1: Traditional Per-Router Control

- Control plane runs in each router.
- Routing components communicate via routing protocols to compute forwarding tables.
- Distributed architecture.

Approach 2: Software-Defined Networking (SDN)

- Control plane is physically separated from data plane.
- A remote controller computes forwarding tables and installs them on all routers.
- Routers act as pure data-plane devices.
- Controller runs in software, typically on a reliable data center, and communicates with routers using protocols like OpenFlow.

1.6 4.1.1 Summary

- **Forwarding:** Local, fast, hardware-level data-plane function.
- **Routing:** Network-wide, slower, software-level control-plane function.
- **Data Plane:** Handles forwarding using the forwarding table.
- **Control Plane:** Computes paths and installs forwarding table entries.

- **Forwarding Table:** Links control plane and data plane.
- **Per-Router Control:** Traditional distributed routing model.
- **Logically Centralized Control (SDN):** Centralized controller computes forwarding tables.

2 4.1.2 Network Service Model

2.1 1. Concept Overview: The "Contract" of the Network Layer

The network service model defines the characteristics of end-to-end delivery of packets between sending and receiving hosts. It represents the "contract" between the network and transport layers:

- Will the packet be delivered?
- Will it arrive intact?
- Will it arrive in order?
- Will it arrive within a certain time?
- Is there a minimum throughput guarantee?

2.2 2. Technical Mechanism: A "Menu" of Possible Services

Potential services a network layer could offer:

- **Guaranteed Delivery:** Every packet eventually arrives.
- **Guaranteed Delivery with Bounded Delay:** Delivery occurs within a specified time.
- **In-Order Packet Delivery:** Packets arrive in sending order.
- **Guaranteed Minimal Bandwidth:** Network behaves like a dedicated link with a specified bit rate.
- **Security:** Encrypt all datagrams for confidentiality.

2.3 3. Practical and Intuitive View: The Internet's *Actual* Service

The Internet provides a single service: **best-effort service**:

- Delivery is not guaranteed; packets may be lost.
- In-order delivery is not guaranteed; packets may arrive out of order.

- Delay is not guaranteed; packets may experience long delays.
- Bandwidth is not guaranteed.

Design philosophy: *Keep the network core simple.* Reliability, ordering, and congestion control are pushed to the edges (transport/application layers).

2.4 4. Terminology Clarification: Switches vs. Routers

- **Packet Switch:** Generic device moving packets from input to output interfaces.
- **Link-Layer Switch:** Layer 2 device forwarding based on MAC addresses.
- **Router:** Layer 3 device forwarding based on network-layer addresses (IP addresses).

Focus of this chapter: **routers** (Layer 3 devices).

2.5 4.1.2 Summary

- **Network Service Model:** Defines network guarantees for transport layer.
- **Best-Effort Service:** Internet's minimalist model with no delivery, order, delay, or bandwidth guarantees.
- **Packet Switch, Link-Layer Switch, Router:** Key terminology distinctions.
- Minimalist model pushes intelligence to end systems.

3 4.1 Overview of Network Layer

3.1 Core Role and Architectural Placement

1. Network layer is third layer of the internet protocol stack
2. Resides between the transport layer and link layer
3. Fundamental role is to move network layer packets, called datagrams, from a sending host to a receiving host.
4. Host-to-Host Communication
 - While transport layer provides logical communication between processes, network layer provides logical communication between hosts
5. Universal Implementation
 - Transport and application layers run only on end systems (hosts)
 - Network layer runs in every host and router in network

- It's essential b/c routers must examine datagram headers to perform their forwarding function

6. Router Protocol Stack

- Routers are network-layer (Layer 3) devices
- Consist of "truncated" protocol stack, implementing physical, link, and network layers
- Do not implement transport or application layers
- Sole purpose is to forward datagrams, not run end-user applications

7. At Each Stage

(a) Sending Host

- Network layer takes segments from transport layer, encapsulates them into datagrams, and sends these datagrams to its nearby router

(b) Receiving Host

- Network layer receives datagrams from its nearby router, extracts transport-layer segments, and delivers them up to transport layer

(c) Routers

- Core function to examine header of an arriving datagram and forward it to appropriate output link
- Routers are 'Layer 3' devices
- Typically have a 'truncated' protocol stack, implementing up to network layer but not transport or application layers

3.2 Planes

3.2.1 Data Plane

- Performs per-router function of forwarding datagrams from a router's input link to its appropriate output link

3.2.2 Control Plane

- Performs network-wide logic that controls how datagrams are routed along an end-to-end path from source to destination host

4 4.2 What's Inside a Router?

4.1 Router Architecture

1. Primary function to transfer packets from incoming links to appropriate outgoing links

2. Task split among four interconnected components
 - (a) Input Ports (Hardware): Perform physical, link, and data-plane lookup for incoming packets

4.2 Template Example

- X
 1. Y

5 4.3 The Internet Protocol (IP): IPv4, Addressing, IPv6, and More

5.1 Overview

- Delves into foundational protocols and addressing schemes of the Internet Protocol (IP) that enable the Internet's network layer.
- IP Protocol provides essential, unreliable, best-effort delivery service that underpins all higher-layer communication
- IP Protocol is the only network-layer protocol used in the Internet

5.1.1 IP Protocol

- Is the only network-layer protocol in the internet
- Is the "narrow waist" of the hourglass, the single protocol that everything else - every transport protocol (TCP, UDP) and every application (HTTP, DNS) - must run on top of
- In turn must be able to run over any link-layer technology (Ethernet, WiFi, 4G, etc)

6 4.3.1 IPv4 Datagram Format

6.1 Concept Overview

Datagram: 1. The Internet's network-layer packet is known as a **datagram**.

2. Fundamental to understanding IP's operations.
3. Format is defined in **RFC 791**.
4. IPv4 datagram has a variable-length header:
 - Typically 20 bytes long when no options are present.
 - Followed by a payload (data field).

5. Composed of two main parts:
- Header:** Contains all information a router needs to make forwarding decisions.
 - Data (Payload):** Contains the data being transported, typically a transport-layer segment (like TCP or UDP).

6.2 Technical Mechanism: The Datagram Format

- A textual representation of the IPv4 header:

0	1	2	3
0 1 2 3 4 5 6 7 8 9 0	1 2 3 4 5 6 7 8 9 0	1 2 3 4 5 6 7 8 9 0	1 2 3 4 5 6 7 8 9 0 1
+-----+-----+-----+-----+			
Version IHL Type of Service		Total Length	
+-----+-----+-----+-----+			
Identifier	Flags	Fragment Offset	
+-----+-----+-----+-----+			
Time-to-Live	Protocol	Header Checksum	
+-----+-----+-----+-----+			
Source IP Address			
+-----+-----+-----+-----+			
Destination IP Address			
+-----+-----+-----+-----+			
Options (if any) ...			
+-----+-----+-----+-----+			
Data (Payload)			
. . .			
+-----+-----+-----+-----+			

- Key Header Fields and Their Purpose:

- Version (4 bits):**
 - Specifies IP protocol version
 - For IPv4, this is always 4.
 - Router examines this field first to know how to parse the rest of the header (e.g. to know that it should expect an IPv4 header, not IPv6 header)
- IHL (Internet Header Length) (4 bits):**
 - Specifies length of IP header in 32-bit words
 - Field is necessary because 'Options' field can make header variable-length
 - Required to find start of a data/payload fields
 - Typical header length is 20 bytes (5 words)
- Type of Service (TOS) (8 bits):** Indicates priority and routing preferences.

- (a) Included to allow different types of IP datagrams (e.g. real-time vs non-real time) to be distinguished and serviced differently (e.g. with priority or low-delay)
 - (b) Bits are also used in Explicit Congestion Notification (ECN)
4. **Datagram Length (16 bits):** Length of the entire datagram (header + payload).
- (a) Total length of IP datagram (header + payload) in bytes
 - (b) Practical View:
 - i. A 16-bit field allows for theoretical maximum datagram size of 65,535 bytes
 - ii. However, datagrams are rarely larger than 1,500 bytes because they must fit within the Maximum Transmission Unit (MTU) of underlying link-layer to avoid fragmentation
5. **Identifier, Flags, Fragmentation Offset (32 bits total):** Identifies fragments of the original datagram.
- (a) Fields are all used for IP fragmentation
 - (b) If router needs to forward a datagram onto a link with an MTU smaller than datagram's size, it can "fragment" datagram into multiple smaller datagrams
 - (c) Such smaller datagrams are reassembled only at final destination
 - (d) This is a slow, complex process, and IPv6 gets rid of it entirely for routes
6. **Time-to-Live (TTL) (8 bits):** Limits the datagram's lifetime to prevent infinite loops.
- (a) Critical safety mechanism to ensure datagrams don't circulate forever in network (e.g. in a routing loop)
 - (b) TTL field is decremented by one at every router it passes through
 - (c) If router receives a datagram with a TTL of 1, it decrements it to 0, discards the datagram, and sends an ICMP error message back to source
7. **Protocol (8 bits):** Specifies the transport protocol in the payload (e.g., TCP, UDP).
- (a) Binds network layer to transport layer
 - (b) Used only at final destination host to indicate which transport-layer protocol should receive datagram's payload
 - i. 6 = TCP
 - ii. 17 = UDP
 - iii. 1 = ICMP (Internet Control Message Protocol)
8. **Header Checksum (16 bits):** Error-checking for the header.
- (a) Performs error detection on header only, not on payload
 - (b) Computed using 1s complement arithmetic over 2-byte units of the header
 - (c) Must be recomputed at every router because TTL field changes

- (d) Why only header?
- Transport layer (TCP/UDP) performs its own checksum on payload
 - TTL field changes at every route. Means header must be altered. Checksum must be recomputed and restored at every single router, which is time consuming
9. **Source and Destination IP Address (32 bits each):** Addresses of originating and final destination interfaces
- Fields contain 32-bit IP addresses of original src and final dest
 - Src address inserted by sending host
 - Dest address obtained from a DNS lookup
10. **Options (if any) (variable length):** Optional fields for control or security.
- Field allows IP header to be extended, but rarely used
 - Can be problematic because makes header length variable, complicating and slowing down router processing
11. **Data (Payload):** Actual transported data (e.g., TCP/UDP segment).

6.3 Template Example

- X
 - 1. Y

7 4.3.2 IPv4 Addressing

7.1 Concept Overview: Interfaces and Subnets

- IP Addresses and Interfaces
 1. IP addr is 32 bits long
 2. IP addr associated with a router or host interface, not device itself
 3. Router has multiple interfaces (one per link), and thus multiple IP addrs
 4. Hosts typically have one interface (and one IP address) connecting them to network
 5. Addresses are written in dotted-decimal notation

7.2 Technical Mechanism: The IP Subnet

- IP addresses not assigned randomly
- Portion of address is determined by network interface connected to
- /24 notation is subnet mask

1. Indicates that leftmost 24 bits define subnet, and remaining $(32-24) = 8$ bits identify specific interfaces on that subnet
- Recipe for Identifying Subnets
 1. Detach every interface from its host or source
 2. This creates "islands" of isolated networks
 3. Each island is a subnet
- Subnets
 1. IP subnet is a network that connects multiple host interfaces and router interfaces,
 2. Forms isolated network without intervening routers
- Subnet Addressing
 1. Interfaces on given subnet share same high-order bits of their IP addresses
 2. Notation (CIDR)
 - (a) A subnet address is denoted by form a.b.c.d/x, where /x indicates number of high-order bits that constitute network portion of address

7.3 Concept: CIDR (Classless Inter-Domain Routing)

1. Is a flexible IP address assignment strategy that replaced older class-based system
2. Introduced /x notation, where x specifies how many bits of the address form network prefix
- Mechanism
 1. CIDR generalizes traditional subnet and host division
 - (a) In address a.b.c.d/x
 - (b) x most significant bits are prefix
 - (c) Remaining $32 - x$ bits are host portion
 2. Allows networks to have variable-length prefixes, making address allocation more flexible
- Scalability via Address Aggregation
 1. Organizations are assigned contiguous blocks of IP addresses with a common prefix
 2. Routers outside of org only need one entry in their forwarding tables.
 - (a) "To reach any address starting with 200.23.16.0/20, forward to that org's ISP"
 3. Ability to use single prefix to advertise many networks = address aggregation (or route aggregation)
 4. Essential for keeping global routing tables small and manageable

7.4 Obtaining and Managing IP Addresses

1. Getting a block of addresses (for an ISP or large org)
 - (a) Global authority is ICANN (Internet Corporation for Assigned Names and Numbers)
 - (b) ICANN allocates address blocks to Regional Internet Registries (RIRs)
 - (c) An ISP (like Comcast, or Verizon) gets its address blocks from its RIR
 - (d) An org gets its address blocks from its ISP
 - (e) Hierarchical allocation allows for route aggregation
2. Getting a host address (DHCP)
 - (a) Once org has blocks of addresses (e.g. 68.85.2.0/24) it needs to assign individual addresses to its hosts
 - (b) Done automatically by Dynamic Host Configuration Protocol (DHCP)
 - (c) DHCP - is a plug and play protocol. When your laptop connects to a network, uses DHCP to automatically get
 - i. Its IP address (e.g. 68.85.2.101)
 - ii. Its subnet mask (e.g. /24)
 - iii. IP address of default gateway (first hop router)
 - iv. IP address of local DNS server
 - (d) How DHCP works?
 - i. DHCP Discover
 - A. New host client sends broadcast message (Dest IP: 255.255.255.255, Source IP: 0.0.0.0)
 - B. Asks "is there a DHCP server out there?"
 - ii. DHCP Offer
 - A. A DHCP server (typically on the router) receives the discover and replies with offer message
 - B. Includes proposing an IP address, lease time, etc.
 - iii. DHCP Request
 - A. Client formally requests offered address by sending a broadcast "request" message
 - B. This tells any other servers that sent offers that they weren't chosen
 - iv. DHCP ACK
 - A. Server confirms allocation with a "DHCP ACK" message
 - B. Client can now use IP address

7.5 Template Example

- X
 - 1. Y