**Network Layer: Data Plane**

**Networking Layer**

**Internetworking**: Routers forward packets from source to destination, crossing several networks along the way

The Network Layer transports segments from sending to receiving hosts.

* Network layer protocols are in every host, every router.
* Routers examine the header fields in all IP datagrams passing through it.

Two key network-layer functions:

1. **Forwarding**: Move packets from router’s input to appropriate router output.  
   (Getting through an interchange station)
2. **Routing**: Determine the route taken by packets from source to destination.  
   (Planning your trip from source 🡪 multiple stations 🡪 destination)

**Routing Algorithm** determines the end-to-end path through the network

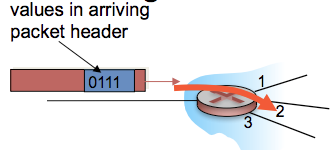
**Forwarding Table** then determines local forwarding at this router

* Packet arrives at Router
* Router uses forwarding table to determine which output link to forward the packet to **{ K=header val , V=output link }**

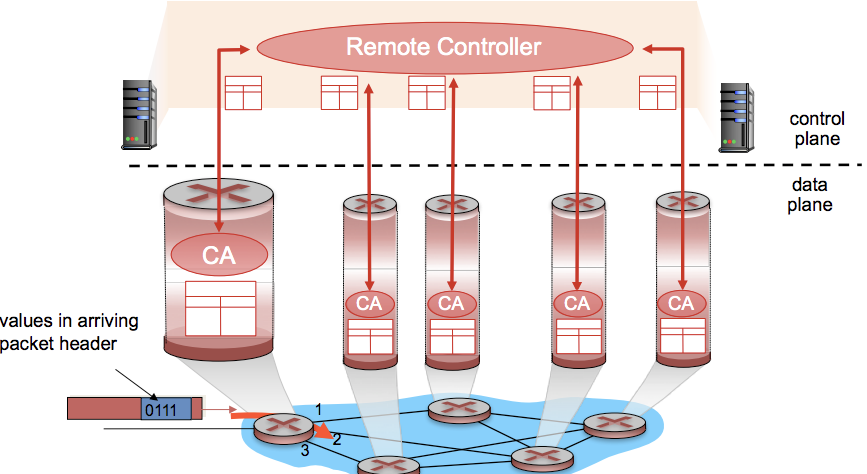
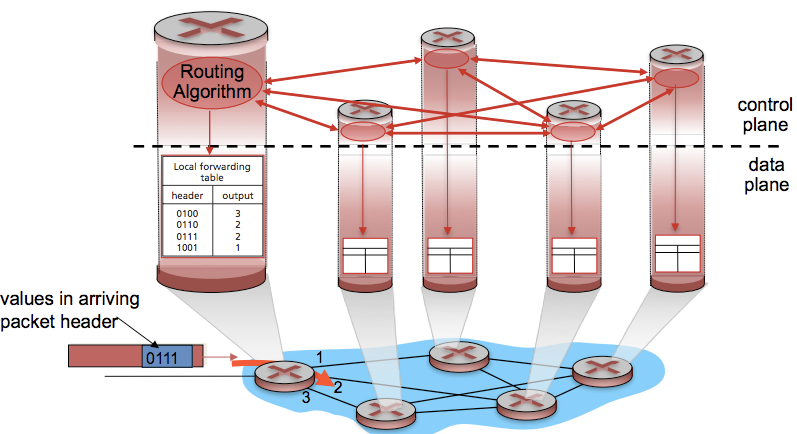
**Network Layer: Data vs Control Plane**

The **Control Plane** refers to the functions that determine how a packet is routed among routers in the end-to-end path.

The **Data Plane** refers to the functions that determine how packets are forwarded from a router input to its output port.



There are two Control Plane approaches:



**Per-Router Control Plane Logically Centralised Control Plane (SDN)**

**SDN: Software-Defined Networking**

Centralised servers.

A distinct controller interacts with local control agents (CA’s)

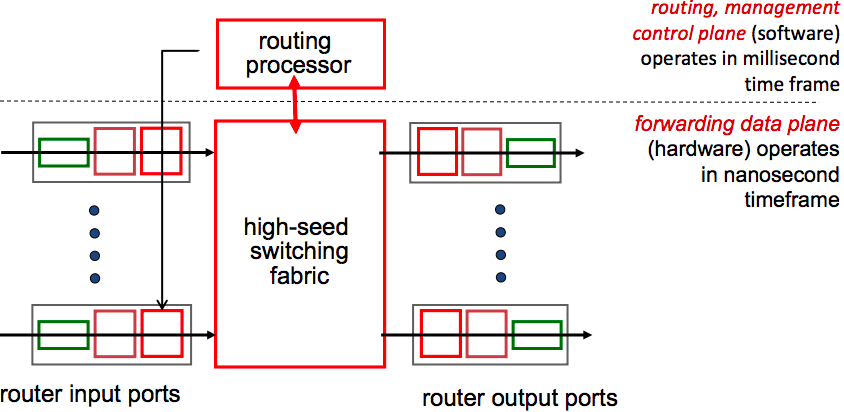
Individual Routing Algorithms in each and every router interact in the control plane.

**Network Layer: Service Models**

The N.L service model defines characteristics of the transport of data between one “edge” of the network to the other.

Example services include: *Guaranteed Delivery, Guaranteed Minimum Bandwidth to Flow, In-Order Deliveries etc.*

**Router Architecture Overview**



**Input Port Functions**

**Line Termination**

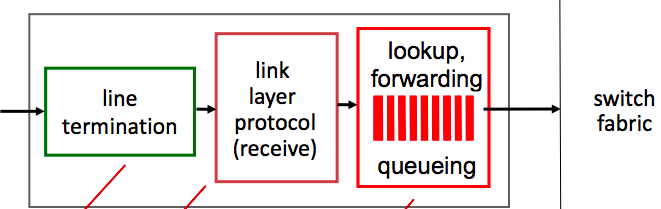
* Physical layer: bit-level reception

**Link Layer Protocol (Receive)**

* Data Link Layer e.g. Ethernet

**Lookup, Forwarding, Queuing**

* Decentralised Switching



**Decentralised Switching**

* Lookup the output port using header field values and forwarding table in input port memory. (*Match Plus Action*)
* Goal: Finish input port processing at line speed
* Queuing: if datagrams arrive faster than forwarding rate into switch fabric
* Two types of forwarding: Destination-Based (based on ONLY IP address) vs. Generalised (based on header fields)

**(1) Destination-based forwarding**: forward based only on destination IP address (traditional)

Destination-Based Forwarding Table

|  |  |
| --- | --- |
| **Destination IP Address Range** | Link Interface |
| 11001000 00010111 00010000 00000000 TO  11001000 00010111 00010111 11111111 | 0 |
| 11001000 00010111 00011000 00000000 TO  11001000 00010111 00011000 11111111 | 1 |
| 11001000 00010111 00011001 00000000 TO  11001000 00010111 00011111 11111111 | 2 |
| Otherwise | 3 |

**(2) Generalised forwarding**: forward base on any set of header field value

* Use **Longest Prefix Matching**: For a given DA, use the longest address prefix that matches the destination address.
* STEP 1: Find the IP ranges / entries in the forwarding table which matches with the Destination Address IP.
* STEP 2: Choose the IP range / entry with the longest prefix.

Longest Prefix Matching

|  |  |
| --- | --- |
| **Destination IP Address Range** | Link Interface |
| 192.168.32.0 / 26 | 0 |
| 192.168.32.0 / 24 | 1 |
| 192.120.32.0 / 19 | 2 |
| Otherwise | 3 |

**Question: Which Link Interface would a packet with destination IP 192.168.32.1 go to? [ IMPORTANT TO STUDY ]**

STEP 1: Convert each Destination IP Address Range to binary

|  |  |
| --- | --- |
| **Destination IP Address Range** | Link Interface |
| 192 168 32 0  11000000 10101000 00100000 00000000 / 26 | 0 |
| 192 168 32 0  11000000 10101000 00100000 00000000 / 24 | 1 |
| 192 120 32 0  11000000 01111000 00100000 00000000 / 19 | 2 |
| Otherwise | 3 |

STEP 2: Get the prefix

|  |  |
| --- | --- |
| 11000000 10101000 00100000 00\*\*\*\*\*\* 26 binary digits | 0 |
| 11000000 10101000 00100000 \*\*\*\*\*\*\*\* 24 binary digits | 1 |
| 11000000 01111000 001\*\*\*\*\* \*\*\*\*\*\*\*\* 19 binary digits | 2 |
| Otherwise | 3 |

STEP 3: Convert Destination IP Address to binary

192.168.32.1 = 11000000 10101000 00100000 00000001

STEP 4: Find matching entries.

11000000 10101000 00100000 00000001 matches with Link Interfaces [0] and [1]

STEP 5: If more than one match, choose one with the longest prefix

Link Interface [0] has the longest prefix of 26. Forward the packet to Link Interface [0].