Network Layer

Dr. Raja Vara Prasad,
IIIT Sri City, Chittoor

Network Layer

Functionalities

- forwarding
- routing
- connection setup

Services

- guaranteed delivery
- guaranteed delivery with bounded delay
- in-order packet delivery
- guaranteed maximum jitter

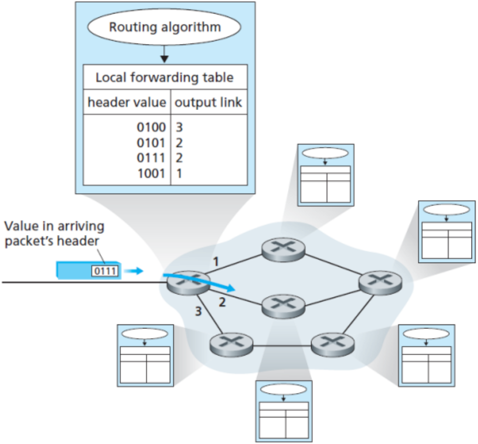
Topics:

- Inside a router
- Packet forwarding
- Internet Protocol
- Addressing and IPV4
- Network Address Translation (NAT)
- Datagram Fragmentation
- Internet Control Message protocol and IPV6

Functionalities

Forwarding

- transfer of a packet from an incoming link to an outgoing link within a *single* router
- Every router has a **forwarding table**.
- forwards a packet by examining the value of a field in the arriving packet's header
- header value to index into the router's forwarding table.
- value stored in the forwarding table entry for that header indicates the router's outgoing link interface to which that packet is to be forwarded.



Network Layer

Functionalities

Routing

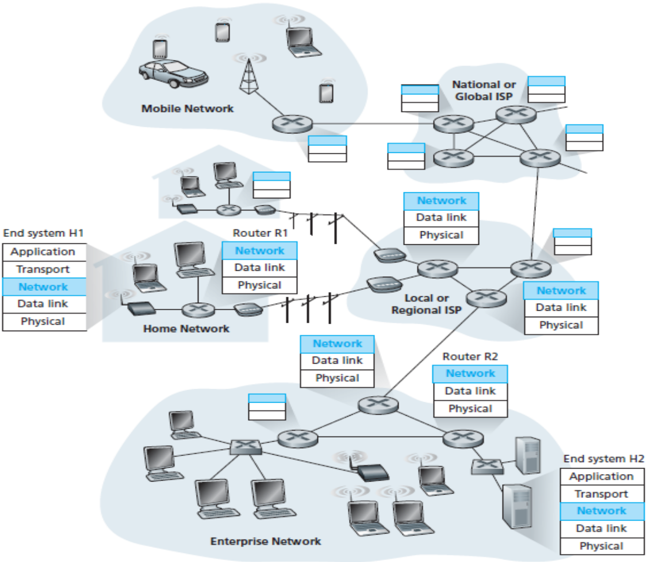
- involves *all* of a network's routers,
- collective interactions via routing protocols
- determine the paths that packets take on their trips from source to destination node
- must determine the route or path taken by packets ---from a sender to a receiver.
- algorithms to calculate these paths -- **routing algorithms**
- Routing algorithms : Centralized or Decentralized
- Human configuration without routing – response to changes in the network is slow
- Link layer or L2 switches: forwarding decision on values in the fields of the link layer
- Router or L3 switches: forwarding decision on the value in the networklayer
- Field.
- L3 require L2 services also

Network Layer

Routing:

- Primary role of the routers is to forward datagrams from input links to output links

- Routers do not run application- and transport-layer protocols



Connection setup

- Some network-layer architectures
- require the routers along the chosen path from source to destination to handshake with each other
- to set up state before network-layer data packets within a given source-to-destination connection can begin to flow

Network Services Models:

Transport layer at a sending host transmits a packet into the network

- can the transport layer rely on the network layer

to deliver the packet to the destination?

- When multiple packets are sent, will they be delivered to the transport layer in the receiving host in the order in which they were sent?

Network Services Models:

- Will the amount of time between the sending of two sequential packet transmissions be the same as the amount of time between their reception?
- Will the network provide any feedback about congestion in the network?
- What is the abstract view (properties) of the channel connecting the transport layer in the sending and receiving hosts?

Network service model defines the characteristics of end-to-end transport of packets between sending and receiving end systems

- guaranteed delivery
- guaranteed delivery with bounded delay
- in-order packet delivery
- *Guaranteed minimal bandwidth*
- *Security services*

Network Layer

- Guaranteed delivery

  guarantees that the packet

- guaranteed delivery with bounded delay

  not only guarantees delivery of the packet, but delivery within a specified host-to-host delay bound

- in-order packet delivery

  guarantees that packets arrive at the destination in the order that they were sent

- *Guaranteed minimal bandwidth*

  sending host transmits bits at a rate below the specified bit rate,

  no packet is lost

  each packet arrives within a pre-specified host-to-host delay

- *Guaranteed maximum jitter*

  time between the transmission and reception of two successive packets at the sender and receiver

- *Security services*

  source host could encrypt the payloads of all datagrams

  destination host responsible for decrypting the payloads

Network Layer

Network Architecture	Service Model	Bandwidth Guarantee	No-Loss Guarantee	Ordering	Timing	Congestion Indication
Internet	Best Effort	None	None	Any order possible	Not maintained	None
ATM	CBR	Guaranteed constant rate	Yes	In order	Maintained	Congestion will not occur
ATM	ABR	Guaranteed minimum	None	In order	Not maintained	Congestion indication provided

CBR: real-time, constant bit rate audio and video traffic
virtual pipe for a dedicated fixed-bandwidth transmission

Available bit rate (ABR) ATM network service.

- a minimum transmission rate (MCR) is guaranteed
- If the network has enough free resources at a given time → higher rate than the MCR

Virtual-Circuit and Datagram Networks

Transport layer provides a choice between two services:
UDP or TCP

- Network layer → connectionless service or connection service
- Connection begins with handshaking between the source and destination
- connectionless service → no handshaking preliminaries.
- connection service at the network layer → **virtual-circuit** (**VC**) **networks**;
- A connectionless service at the N/W layer → **datagram networks**.

Virtual-Circuit

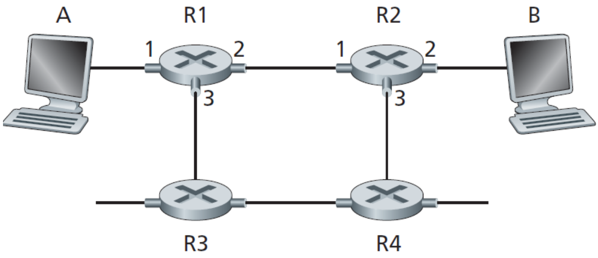
virtual-circuit networks, use connections at the network layer.
--- connections known as **virtual circuits (VCs)**

Ex: ATM and Frame Relay

A VC consists:
(1) a path (series of links and routers) between the source and destination hosts
(2) VC numbers, one number for each link along the path,
(3) entries in the forwarding table in each router along the path.

- packet will carry a VC number in its header
- a virtual circuit may have a different VC number on each link,
- each intervening router must replace the VC number of each traversing packet with a new VC number.
- The new VC number is obtained from the forwarding table.

Virtual-Circuit



Incoming Interface	Incoming VC #	Outgoing Interface	Outgoing VC #
1	12	2	22
2	63	1	18
3	7	2	17
1	97	3	87
...	...	...	...

Virtual-Circuit

- new VC across a router → entry is added to the forwarding table.
- whenever a VC terminates →entries in each table along its path are removed.

why a packet doesn't just keep the same VC number on each of the links along its route ?

- replacing the number from link to link reduces the length of the VC field in the packet header
- with multiple VC numbers, each link in the path can choose a VC number independently of the VC numbers chosen at other links along the path.
- If a common VC number were required for all links along the path, the routers would have to exchange and process a substantial number of messages to agree on a common VC number

Virtual-Circuit

- In a VC network, the network's routers must maintain **connection state information** for the ongoing connections.

three phases in a virtual circuit:

1. VC setup:

transport layer contacts the network layer, specifies the receiver's address and waits for the network to set up the VC

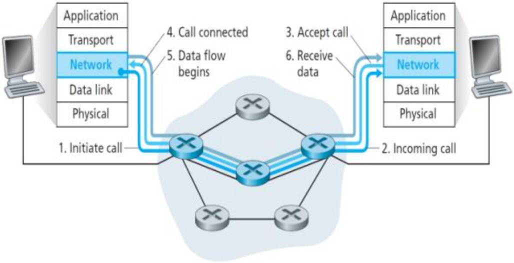
Network layer →

- determines the path,
- series of links and routers through which all packets of the VC will travel
- determines the VC number for each link along the path
- adds an entry in the forwarding table in each router along the path
- may reserve resources like bandwidth along the path of the VC

Virtual-Circuit

*2. **Data transfer:***

3. VC teardown: initiated when the sender (or receiver) informs the network layer of its desire to terminate the VC.



The network layer →

- inform the end system on the other side of the network of the call termination
- update the forwarding tables in each of the packet routers on the path to indicate that the VC no longer exists.
- **Signaling Messages:**

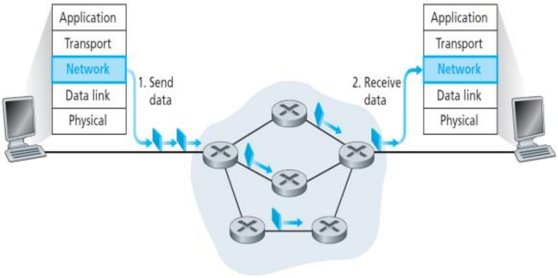
  messages that the end systems send into the network to initiate or terminate a VC

  messages passed between the routers to set up the VC
- Signaling Protocols

Datagram Network

- each time an end system wants to send a packet,
- it stamps the packet with the address of the destination end system and then pops the packet into the network
- passes through a series of routers
- routers uses the packet's destination address to forward the packet
- each router has a forwarding table that maps destination addresses to link interfaces
- router uses the packet's destination address to look up the appropriate output link interface in the forwarding table

- all destination addresses are 32 bits
- forwarding table would have one entry for every possible destination address
- more than 4 billion possible addresses



Datagram Network

- Router matches a prefix
- if there's a match, the router forwards the packet to a link associated with the match.
- Multiple matches: **longest prefix matching rule**

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

Prefix Match	Link Interface
11001000 00010111 00010	0
11001000 00010111 00011000	1
11001000 00010111 00011	2
otherwise	3

11001000 00010111 00010110 10100001

11001000 00010111 00011000 10101010

Datagram Network

Comparison of VC & Datagram: update intervals of forwarding tables

- time scale at which this forwarding state information changes is relatively slow: every 1-5 minutes
- In a VC network: forwarding table is modified: a new connection is set up or an existing connection is torn down. happen at a microsecond in a backbone router.

@ Datagram

- forwarding tables in datagram networks can be modified at any time,
- a series of packets sent from one end system to another may follow different paths through the network
- may arrive out of order

Datagram

Example of Datagram:

Internet is a datagram network

- more sophisticated end-system devices -- network-layer service model as simple as possible.
- in-order delivery, reliable data transfer, congestion control etc implemented at a higher layer, in the end systems.
- Internet network-layer service model makes minimal service guarantees, it imposes minimal requirements on the network layer
- makes it easier to interconnect networks that use very different link-layer technologies:

  ex: satellite, Ethernet, fiber, or radio with different transmission rates and loss characteristics
- to add a new service simply by attaching a host to the network and defining a new application-layer protocol has allowed new Internet applications in a short period of time.

Inside a Router

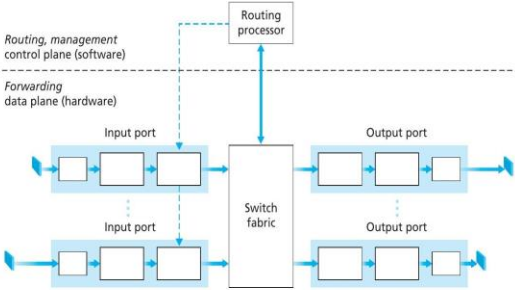


Figure 4.6 ♦ Router architecture

- Input port
- Switching fabric
- Output port
- Routing processor

Inside a Router

Input port:
performs the physical layer function
performs link-layer functions
lookup function is also performed
Control packets (for example, packets carrying routing protocol information) are forwarded from an input port to the routing processor

Switching fabric.
connects the router's input ports to its output ports.
is completely contained within the router: a network inside of a network router!

Output ports:
stores packets received from the switching fabric
transmits these packets on the outgoing link by performing the necessary link-layer and physical-layer functions

Routing processor.
- executes the routing protocols
- maintains routing tables and attached link state information,
- computes the forwarding table for the router.
- performs the network management functions

Inside a Router

Router forwarding plane:
router's input ports, output ports, and switching fabric together → forwarding function → always implemented in hardware: **router forwarding plane**

Ex:

- a 10 Gbps input link and a 64-byte IP datagram,
- the input port has only **51.2 ns** to process the datagram before another datagram may arrive.
- If *N* ports are combined on a line card (as is often done in practice), the datagram-processing **pipeline must operate *N* times faster**
- far too fast for software implementation

Router's control functions

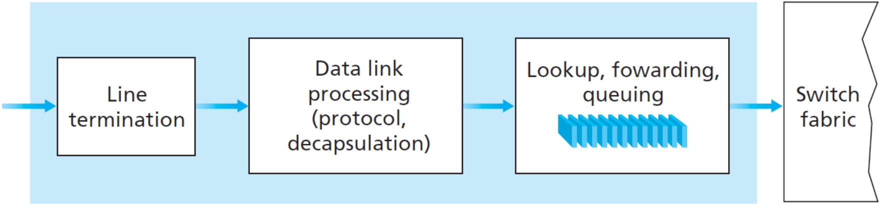
- executing the routing protocols, responding to attached links that go up or down, and performing management functions
- operate at the **millisecond** or second timescale
- **router control plane** functions are usually implemented in **software** and execute on the **routing processor**

Inside a Router

1. Input processing
2. Switching
3. Output processing

1. Input processing:

- input port's line termination function and link-layer processing implement the physical and link layers for that individual input link
- lookup performed → Central to the router's operation → to look up the output port → Arriving packet will be forwarded via the switching fabric.



Inside a Router

1. **Input processing:**

- a shadow copy typically stored at each input port
- forwarding table is copied from the routing processor to the line cards
- forwarding decisions → locally, at each input port --> without invoking the centralized routing processor on a per-packet basis →avoiding a centralized processing bottleneck.
- search through the forwarding table → for the longest prefix match
- at Gigabit transmission rates → lookup → nanoseconds
- techniques beyond a simple linear search through a large table
- Special attention: memory access times → embedded on-chip DRAM, faster SRAM, Ternary Content Address Memories (TCAMs) using the fabric

Inside a Router

1. Input processing:

- a packet may be temporarily blocked from entering the switching fabric if packets from other input ports are currently using the fabric
- will be queued at the input port and then scheduled to cross the fabric at a later point in time

Other important aspects of Input processing:

(1) physical- and link-layer processing must occur, as discussed above;

(2) the packet’s version number, checksum and time-to-live field must be checked and the latter two fields rewritten;

(3) counters used for network management (number of IP datagrams received) must be updated.

Inside a Router

Switching:
through this fabric that the packets are actually switched from an input port to an output port.

Three types of switching:
- *Switching via memory*
- *Switching via a bus.*
- *Switching via an interconnection network*

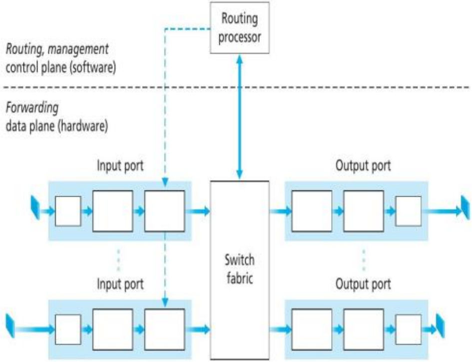


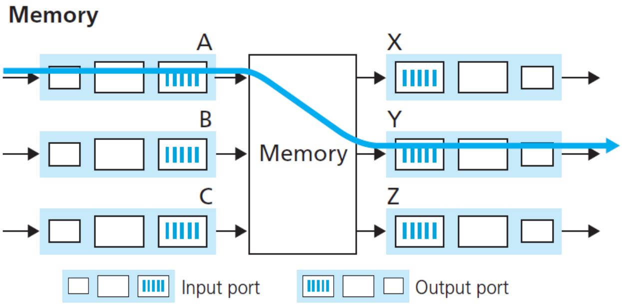
Figure 4.6 ♦ Router architecture

Inside a Router: Switching

Switching via memory

- earliest routers – switching → under direct control of CPU
- *Input port signals the arrival of a packet → routing processor →Interrupt*
- *Processor completes lookup → copies packet → output buffer*
- *Present day routers →* processing on a line card
- shared-memory multiprocessors

Ex: Cisco’s Catalyst 8500 series switches

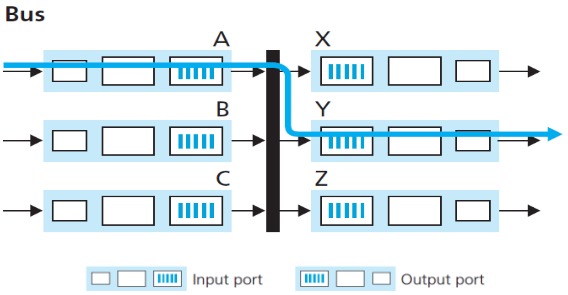


Inside a Router: Switching:

Switching via a bus:

- an input port transfers a packet directly to the output port over a shared bus, without intervention by the routing processor
- input port pre-pend a switch-internal label → indicating the local output port → transmitting the packet onto the bus
- received by all output ports, but only the port that matches the label

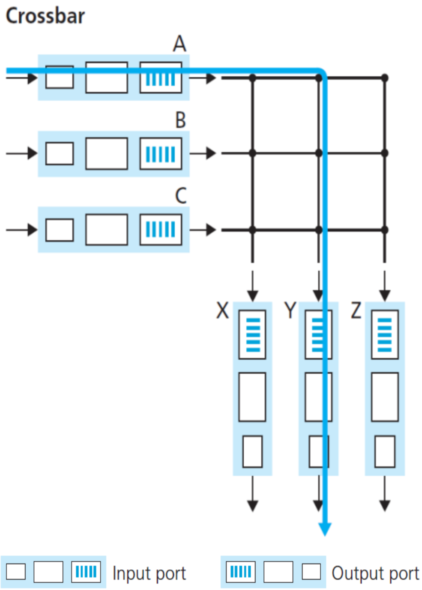
- Even multiple packets at input ports → one packet on bus
- switching speed of the router is limited to the bus speed



Inside a Router: Switching

Switching via an interconnection network:

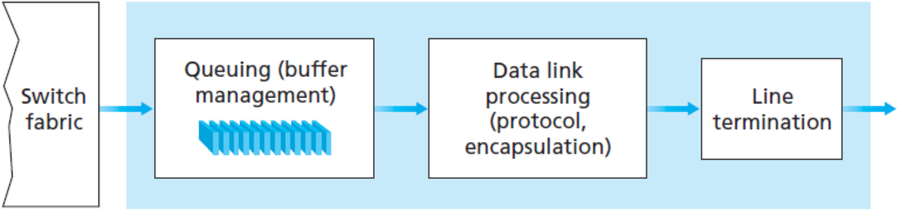
- more sophisticated interconnection network
- crossbar switch is an interconnection network consisting of $2N$ buses that connect N input ports to N output ports
- Each vertical bus intersects each horizontal bus at a crosspoint → can be opened or closed at any time by the switch fabric controller
- crossbar networks are capable of forwarding multiple packets in parallel
- if two packets from two different input ports → to the same output port → one will have to wait at the input



Inside a Router

Output processing:

- takes stored packets in the output port's memory → transmits them over the output link
- selecting and de-queueing packets for transmission
- performing the needed link layer and physical-layer transmission functions



Inside a Router: Output processing

Queueing at input and output ports:

- packet queues may form at both the input ports *and* the output ports
- extent of queueing depend on
  → the traffic load → the relative speed of the switching fabric,
  → the line speed
- queues grow large → router's memory exhaust → **packet loss** will occur when no memory is available to store arriving packets
- an identical input and output transmission rate of R_{line} packets/sec
- R_{switch} rate at which packets can be moved from input to output port
- if $R_{switch} = N^* R_{line}$ negligible queuing will occur at input ports
- If all packets at N input ports are destined to same output port ?
- output port can transmit only a single packet in a unit of time
- *N* arriving packets will have to queue for transmission over the outgoing link.
- number of queued packets can grow large enough → exhaust available memory at the output port → packets are dropped

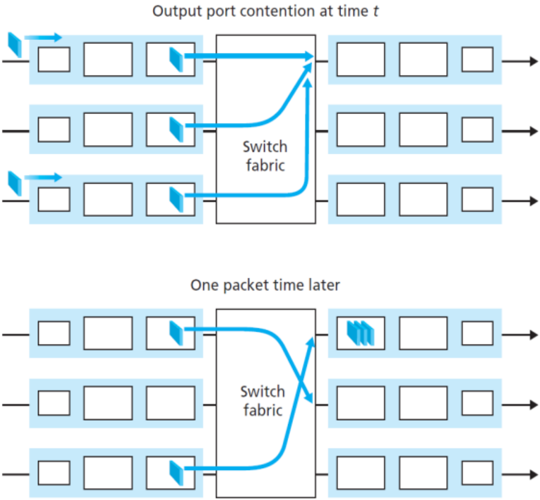
Inside a Router: Output processing

Queueing at input and output ports:

- packet queues may form at both the input ports *and* the output ports
- extent of queueing depend on
  →the traffic load → the relative speed of the switching fabric,
  →the line speed
- queues grow large → router's memory exhaust → **packet loss** will occur when no memory is available to store arriving packets
- an identical input and output transmission rate of R_{line} packets/sec
- R_{switch} rate at which packets can be moved from input to output port
- if $R_{switch} = N^* R_{line}$ negligible queuing will occur at input ports
- If all packets at N input ports are destined to same output port ?
- output port can transmit only a single packet in a unit of time
- *N* arriving packets will have to queue for transmission over the outgoing link.
- number of queued packets can grow large enough → exhaust available memory at the output port → packets are dropped

Inside a Router: Output processing:

- **packet scheduler** at the output port must choose one packet among those queued for transmission
- first-come-first-served (FCFS)
- weighted fair queuing (WFQ) → shares the outgoing link fairly among the different end-to-end connections that have packets queued for transmission
- no enough memory to buffer an incoming packet either drop the arriving packet or
  remove one or more already-queued packets
- **Random Early Detection -** probabilistic marking/dropping functions



Inside a Router: Output processing:

what if the Switch fabric is not fast enough: → packet queuing at the input ports

Assume:

(1) all link speeds are identical

(2) one packet can be transferred from any one input port to a given output port in the same amount of time it takes for a packet to be received on an input link

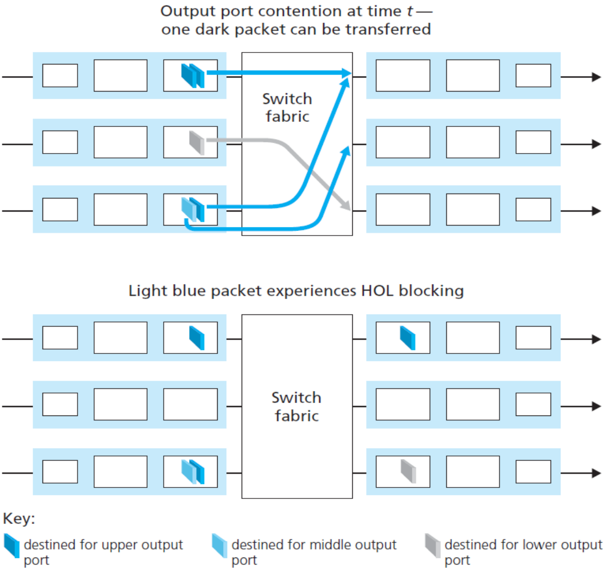
(3) packets are moved from a given input queue to their desired output queue in an FCFS manner

→Multiple packets can be transferred in parallel, as long as their output ports are different

→ if two packets of two input queues are destined for the same output queue one of the packets will be blocked and must wait at the input queue.

→ **head-of-the-line** (**HOL**) **blocking** → queue will grow to unbounded length

Inside a Router: Output processing:



Inside a Router: Routing plane

- fully resides and executes in a routing processor within the router
- network-wide routing control plane → decentralized

→ with different pieces executing at different routers and interacting by sending control messages to each other

New router control plane architectures

→ part of the control plane is implemented in the routers along with the data plane
→ part of the control plane can be implemented externally to the router
→ A well-defined API dictates how these two parts interact and communicate with each other

Software Defined Networking (SDN)

- separating the software control plane from the hardware data plane
- allowing different customized control planes to operate over fast hardware data planes