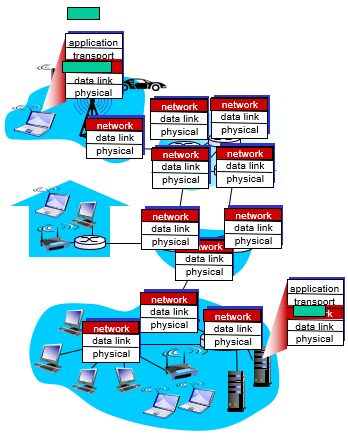
# Network Layer

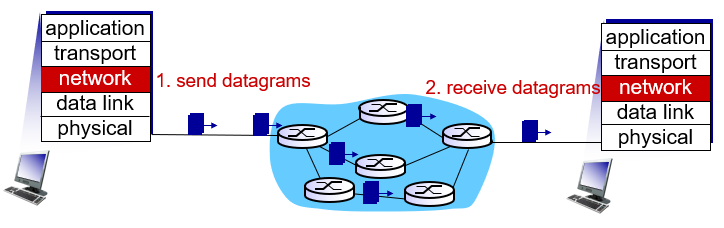
## Network Layer: Goals

* Understand principles behind network layer services
  + Forwarding vs routing
  + Addressing
  + How a router works
  + Routing (path selection)
  + Helpful resources
    - <https://zhuanlan.zhihu.com/p/366194846>
    - <https://zhuanlan.zhihu.com/p/367382743>

## Network Layer

* Role: transports segments from sending to receiving host; provides logical connection between hosts, not processes
  + Sender’s side: encapsulates segments into IP datagrams, passes to link layer
  + Receiver’s side: receives datagrams, extracts transport-layer segments, and delivers segments up to transport layer
* Network layer protocols are in *every* network device: hosts and routers
* Router examines header fields in *all* IP datagrams passing through it

## Datagram networks

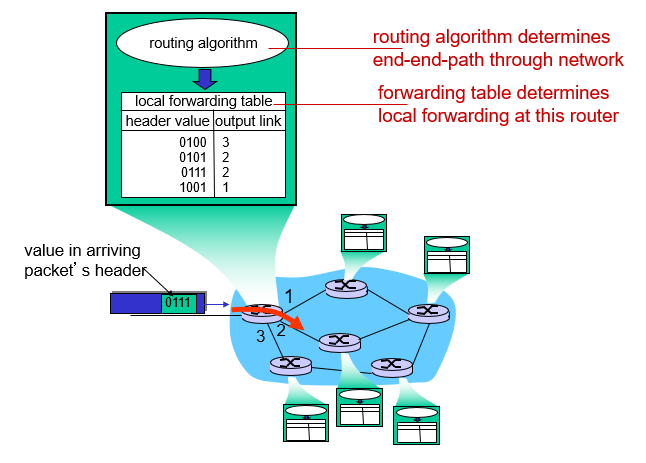
* The most common network model (used on the Internet).
* No call setup at network layer
* Routers: no state about end-to-end connection
  + No network-level concept of connection
* Packets forwarded using destination host address

## Two key network-layer functions

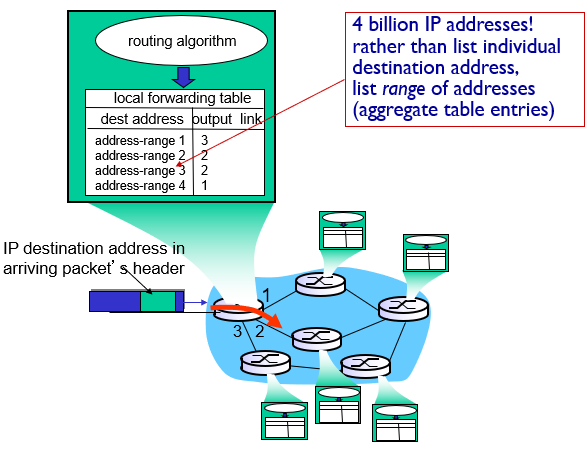
* **Forwarding** (in data plane): local per-router action of transferring packets from a router’s input link to the correct router output link
  + takes place at very short timescales (usually a few nanoseconds)
  + is implemented in hardware either using a router vendor’s own hardware designs, or using purchased merchant-silicon chips (e.g. Intel, Broadcom)
  + trip analogy: process of getting through single interchange
* **Routing** (in control plane): network-wide process to determine route (end-to-end path) taken by packets from source to destination via routing algorithms/protocols (RIP, OSPF, BGP)
  + takes place on much longer timescales (seconds)
  + is implemented in software
  + trip analogy: process of planning trip from source to destination
* Why we need forwarding and routing functions:
  + Human configurations would be more error-prone and slower to respond to changes in network topology than a routing protocol

## Interplay between routing and forwarding

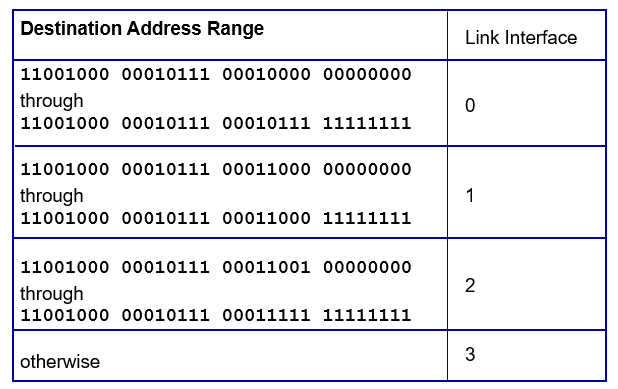
* The individual routing algorithm is implemented in *each* and *every* router
  + routers examine header fields in all IP datagrams passing through it
  + using these header values to index to its forwarding table
  + moves datagrams from input ports to output ports to transfer datagrams along end-to-end path (i.e. route)



## Datagram forwarding table



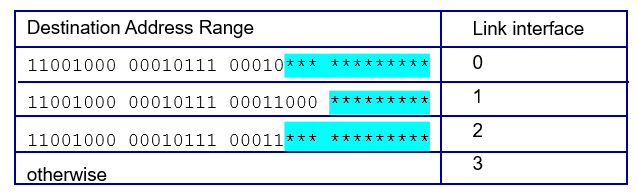
## Datagram forwarding table: example



* Q: but what happens if ranges don’t divide up so nicely?
* A: use longest prefix matching

## Longest Prefix Matching

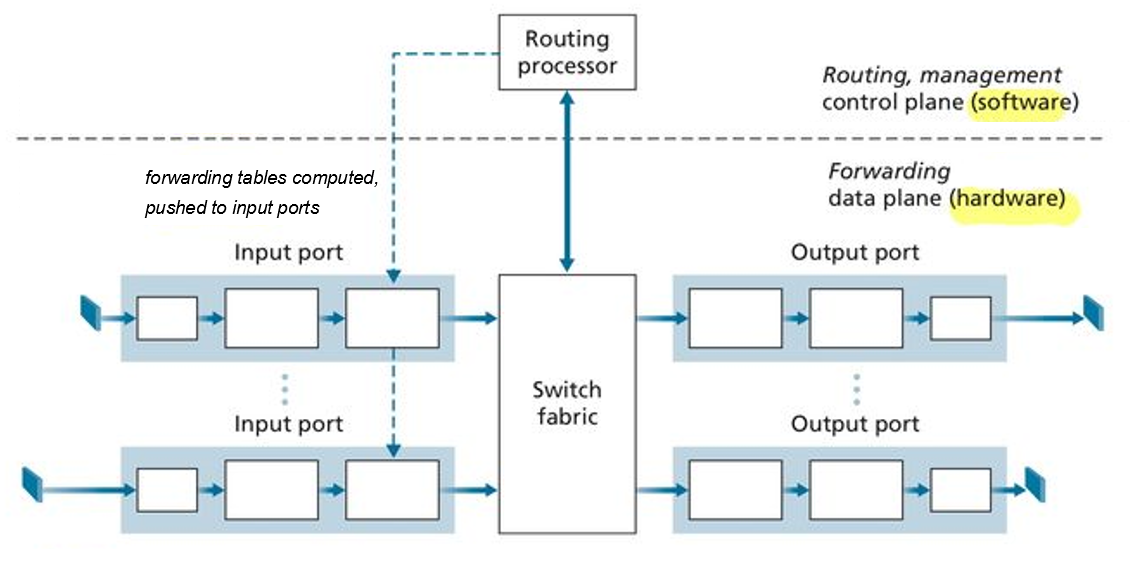
* When looking for forwarding table entry for given destination address, use the *longest* address prefix that matches destination address (i.e. which one matches more?).



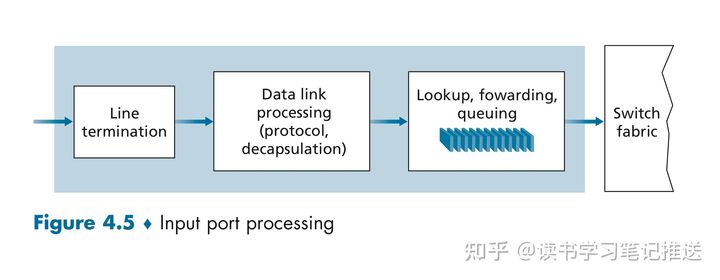
* Examples:
  + DA: 11001000 00010111 00010110 10100001
    - Which interface? Link 0, since it matches most with that range
  + DA: 11001000 00010111 00011000 10101010
    - Which interface? Link 1, matches most with that range

## Router architecture overview

* **[Input Ports]** perform several key functions:
  + physical layer function of terminating an incoming physical link at a router
  + link-layer function to interoperate with the link layer at the other side of the incoming link
  + look-up function using the forwarding table, which determines router output port to which an arriving packet will be forwarded via the switching fabric
* **[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 receiving from the switching fabric and transmits these packets on the outgoing link by performing the necessary link-layer and physical-layer functions
* **[Routing Processer]** performs control-plane functions
  + executes the routing protocols
  + maintains routing tables and attached link state info
  + computes the forwarding table for router
  + performs the network management functions



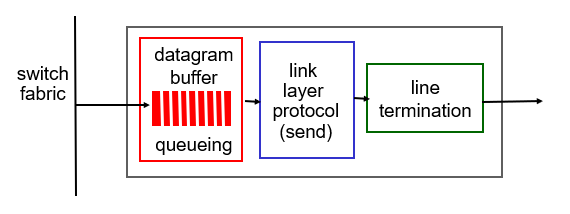
## Input Ports



* **[Line Termination]** is a physical layer function for receiving the bit level transmissions over the physical medium (e.g. copper, fiber, wireless)
* **[Link-Layer Function]** assembles bits into link-layer frames (e.g. Ethernet frames)
* **[Lookup, Forwarding, Queuing]** are network-layer functions,
  + *Lookup and Forwarding*: routers determine appropriate Output Port to which the arriving packet will be forwarded via the switching fabric by using header field values and forwarding table in Input Port memory (match plus action")
    - Destination-based Forwarding: based ONLY on destination IP address (traditional)
    - Generalized Forwarding: based on any set of header field values (i.e. source IP address, header length, seq #...)
  + *Decentralized Switching*: The forwarding table is copied from the Routing Processor to the Input Line Cards over a separate bus
    - Goal: Forwarding decisions can be made locally, complete *Input Port Processing* at *line speed* to avoid a *centralized processing bottleneck*
  + *Queueing*: needed if datagrams arrive faster than forwarding rate into switch fabric

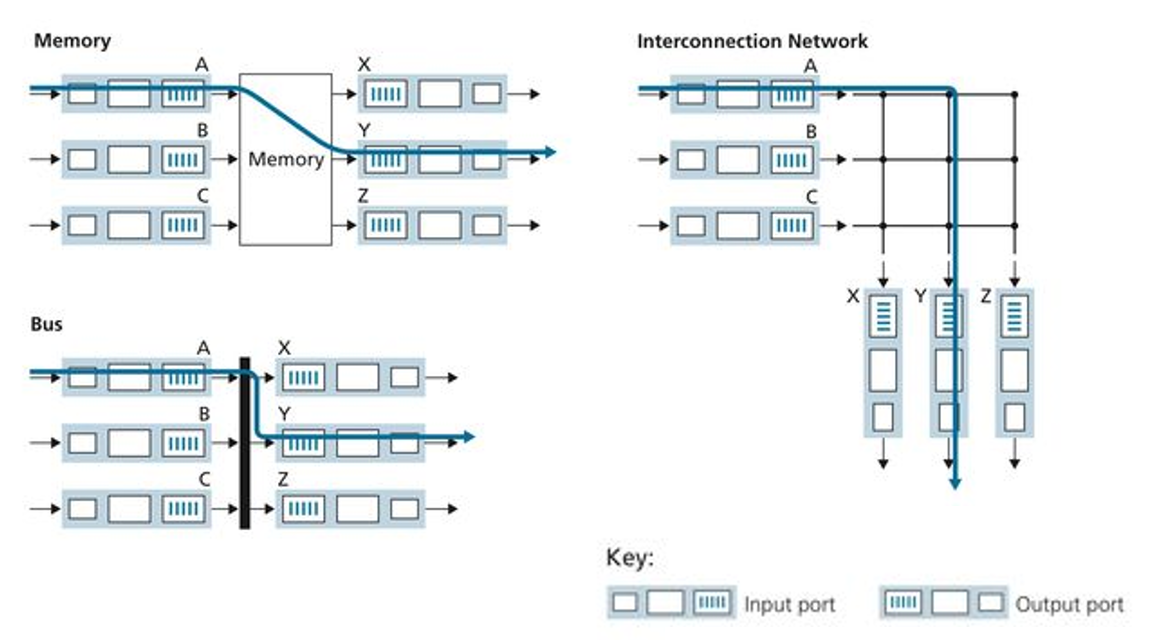
## Output Ports

* *Buffering* required when datagrams arrive from fabric faster than transmission rate



## Switching Fabric

* Switching fabrics forward packets from an Input Link to an appropriate Output Link
* Switching rate: rate at which packets can be transfer from inputs to outputs
  + often measured as multiple of input/output line rate
  + N input ports: switching rate should preferably be N × ‘line rate’ or higher
* Three types of switching fabrics:



## Switching via Memory

* The simplest, earliest routers were traditional computers with switching under direct control of the CPU (routing processor)
* Input and Output ports functioned as traditional I/O devices in a traditional Operating System:
  + an input port with an arriving packet first *signaled* the routing processor via an interrupt
  + then the packet was copied from the input port into processor memory
* The routing processor extracted the destination address from the header,
  + looked up the appropriate output port in the forwarding table
  + and copied the packet to the output port's buffers
* Speed is limited by memory bandwidth…
  + if the memory bandwidth is a maximum of B packets/sec can be written into or read from,
  + then the overall forwarding throughput must be less than B/2 since ONLY ONE memory read/write can be done at a time over the shared system bus

## Switching via a Bus

* An input port transfers a packet directly to the output port via a shared bus
* Only ONE packet can cross the bus at a time, so the switching speed of the router is limited to the bus speed
* Roundabout Analogy: the roundabout could only contain one car at a time.
* Nonetheless, switching via a bus is often sufficient for routers that operate in small local area and enterprise networks

## Switching via an Interconnection Network

* A 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*, which can be opened or closed at any time by the switch fabric controller
* A crossbar switch is Non-Blocking
  + a packet being forwarded to an output port will NOT be blocked from reaching that output port as long as NO other packet is currently being forwarded to that output port
  + however if two packets from different input ports are destined to the same output port, one will have to wait at the input since ONLY ONE packet can be sent over any given bus at a time
* More sophisticated interconnection networks use multiple stages of switching elements to allow packets from different input ports to proceed towards the same output port at the same time through the multi-stage switching fabric.
* A router's switching capacity can also be *scaled* by running multiple switching fabrics in parallel
  + an input port breaks a packet into K smaller chunks
  + sends ("sprays") the chunks though K of N switching fabrics to the selected output port
  + which reassembles the K chunks back into the original packet

## Queuing

* As the queues grow large, the router's memory can eventually be *exhausted* and *packet loss* will occur when NO memory is available to store arriving packets.

## Input Port Queuing

* If switch fabric slower than input line speeds (or slower than input ports combined)…
  + queuing may occur at input queues
  + queuing delay and loss due to input buffer overflow
* Head-of-the-line (HOL) Blocking: Queued datagram at front of queue prevents others in queue from moving forward

## Output Port Queuing

* Buffering required when datagrams arrive from fabric faster than link transmission rate.
  + queueing delay and loss due to output port buffer overflow
  + *Drop Policy*: which datagrams to drop if NO free buffers?
    - tail drop: drop arriving packet
    - priority: drop/remove on priority basis
* Marking Policy: which packets to mark to signal congestion
  + ECN: Explicit Congestion Notification (Recall Sec 3.7.2)
  + RED: Random Early Detection
* *Scheduling Discipline* chooses among queued datagrams for transmission