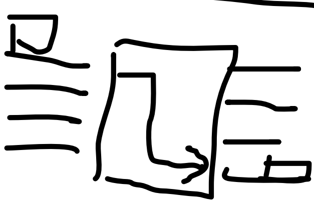
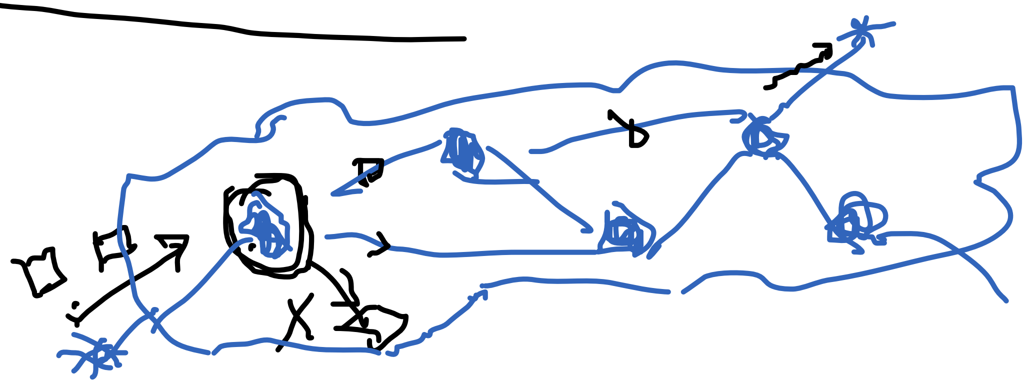
- Appl. Layer : design appl. logic & protocols.

SOCKETS

- Transport Layer: Reliable comm, connection,
(Congestion control).

+ Network Layer. : • forward pkt sends
• route pkt (dest.
• Addressing.





Routers

From. Delhi

postal sys:

Addr.

Addr : TO

Sanjay
DAIICT
Gandhinagar, GJ, India.

Hierarchical addr

Shimla.

Delhi

Jaipur

Ahd

Gandhinagar

Stage &

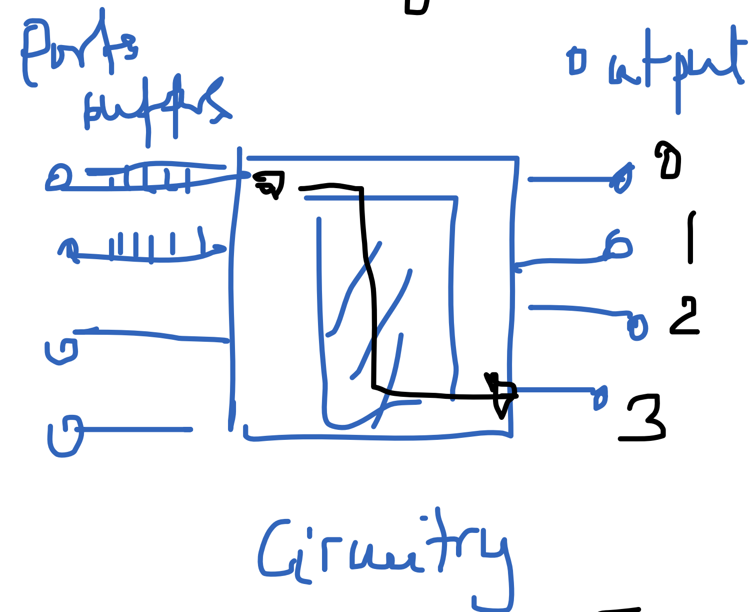
Forwarding

pkt at the input port is moved to output port after consulting a Table (Routing Table)

Dest. Addr	Output port #
X	K = 3
Y	K = 2

"Local" "Data"

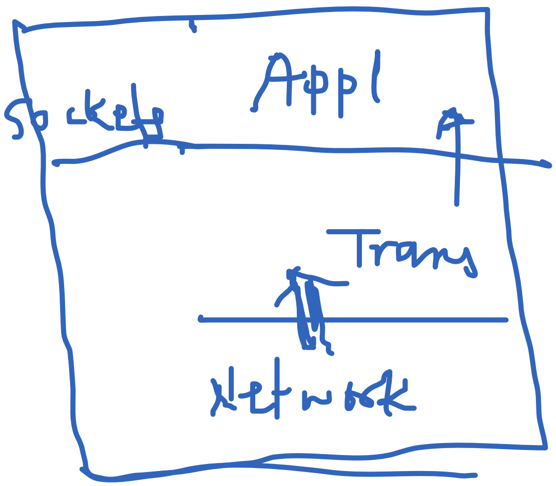
Routing Addressing



Create Rt. Table

"Global" & "Control"

(Layers). Sockets



services

lower layer provide services to upper layer

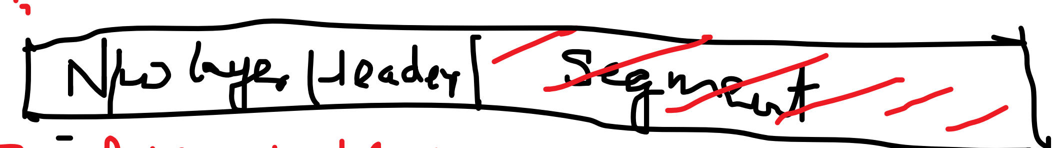
Interface

N/w Services:

given dest → "routing"

segments → "packets"

Trans layer msg = segment $\xrightarrow{\text{encap.}}$ packet



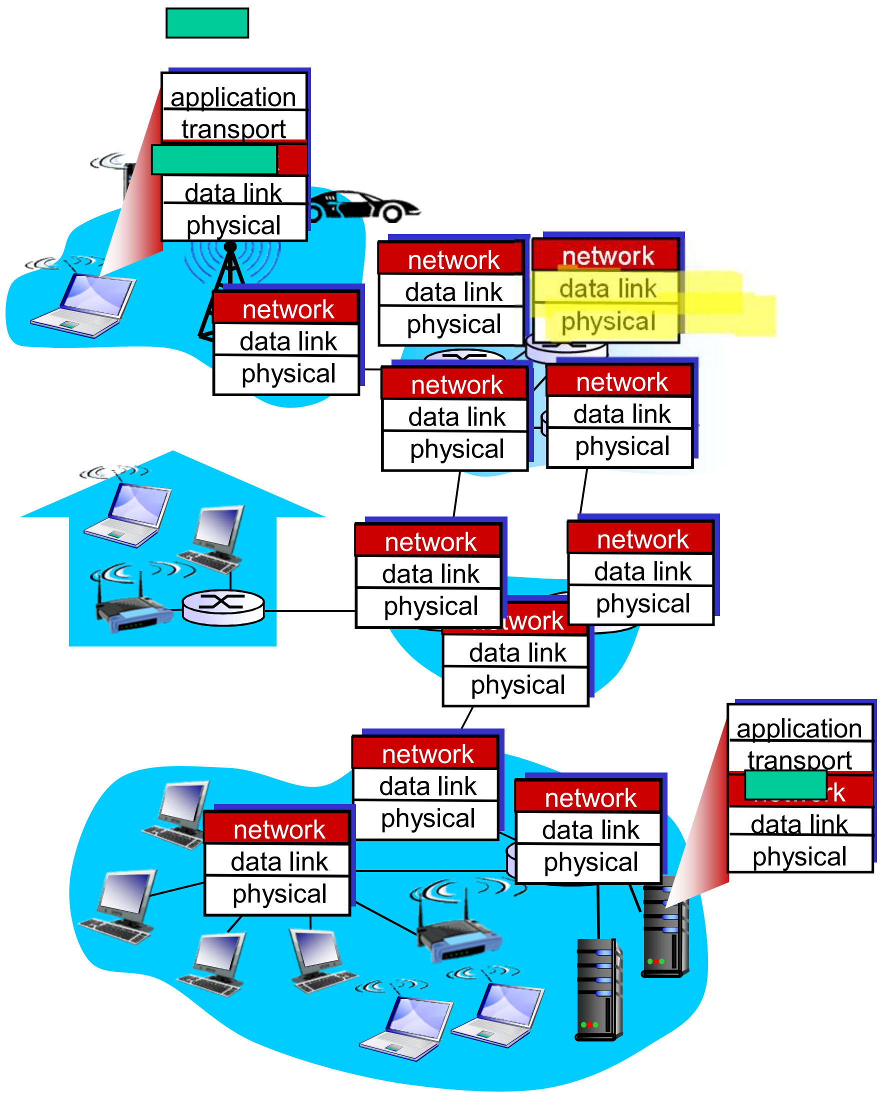
"Src Addr, Dest Addr, ... "

"packet"

N/w layer Msg.

Network layer

- transport segment from sending to receiving host
- on sending side encapsulates segments into datagrams
- on receiving side, delivers segments to transport layer
- network layer protocols in *every* host, router
- router examines header fields in all IP datagrams passing through it



Two key network-layer functions

network-layer functions:

- *forwarding:* move packets from router's input to appropriate router output
- *routing:* determine route taken by packets from source to destination
  - *routing algorithms*

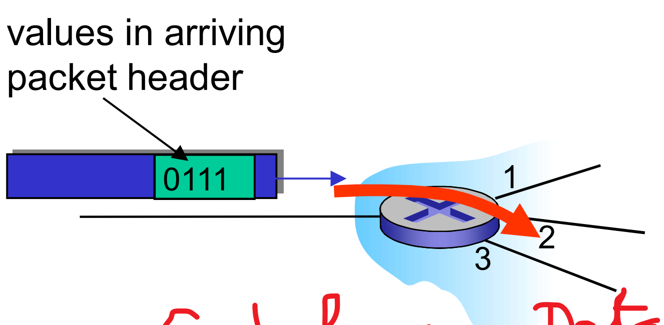
analogy: taking a trip

- *forwarding:* process of getting through single interchange
- *routing:* process of planning trip from source to destination

Network layer: data plane, control plane

Data plane

- local, per-router function
- determines how datagram arriving on router input port is forwarded to router output port
- forwarding function



Control plane

- network-wide logic
- determines how datagram is routed among routers along end-end path from source host to destination host
- two control-plane approaches:
  - *traditional routing algorithms:* implemented in routers
  - *software-defined networking (SDN)*: implemented in (remote) servers

Router: Control | Data

Fun.

Per-router control plane

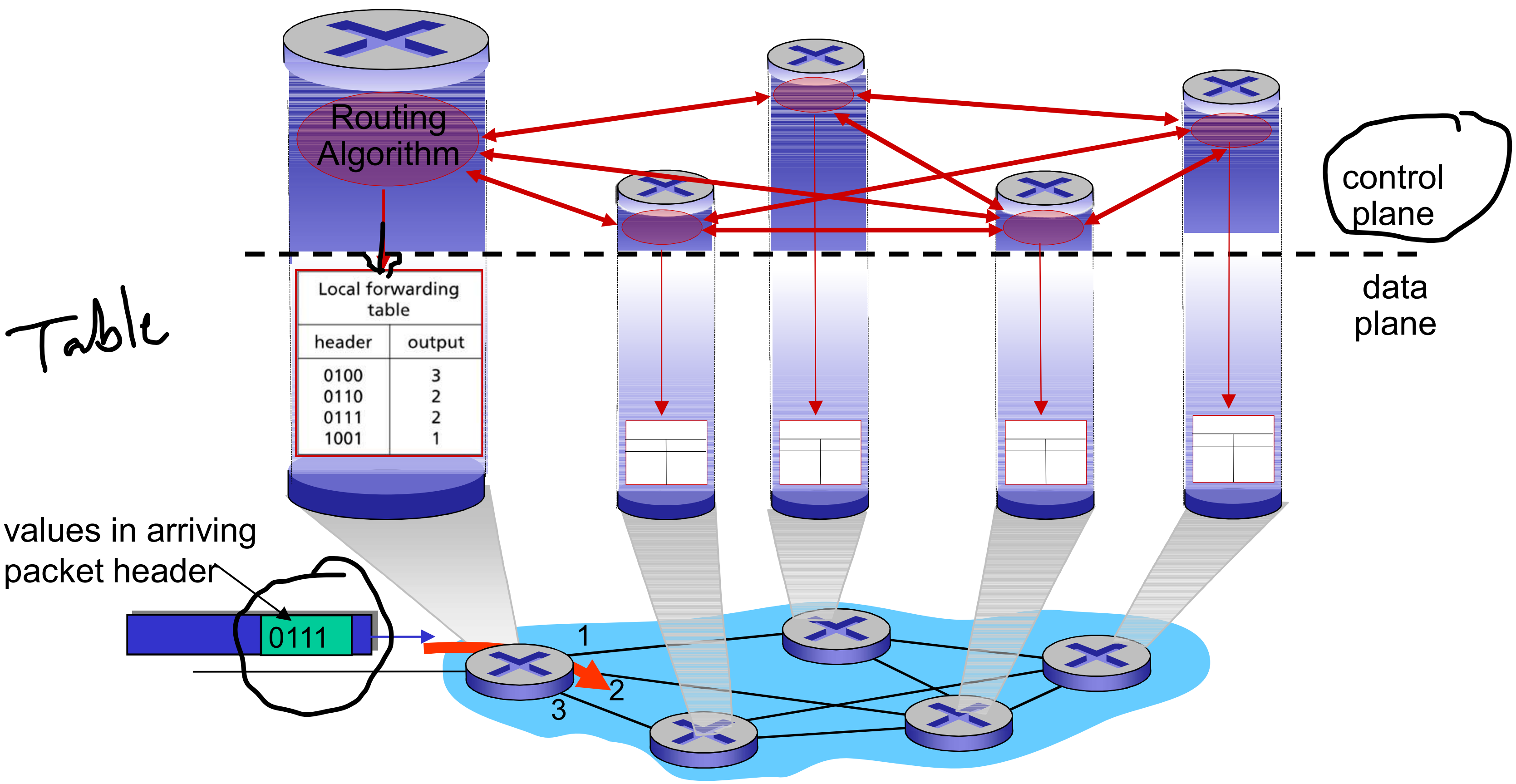
Layers

Control | Data

plane

Individual routing algorithm components *in each and every router* interact in the control plane

Table



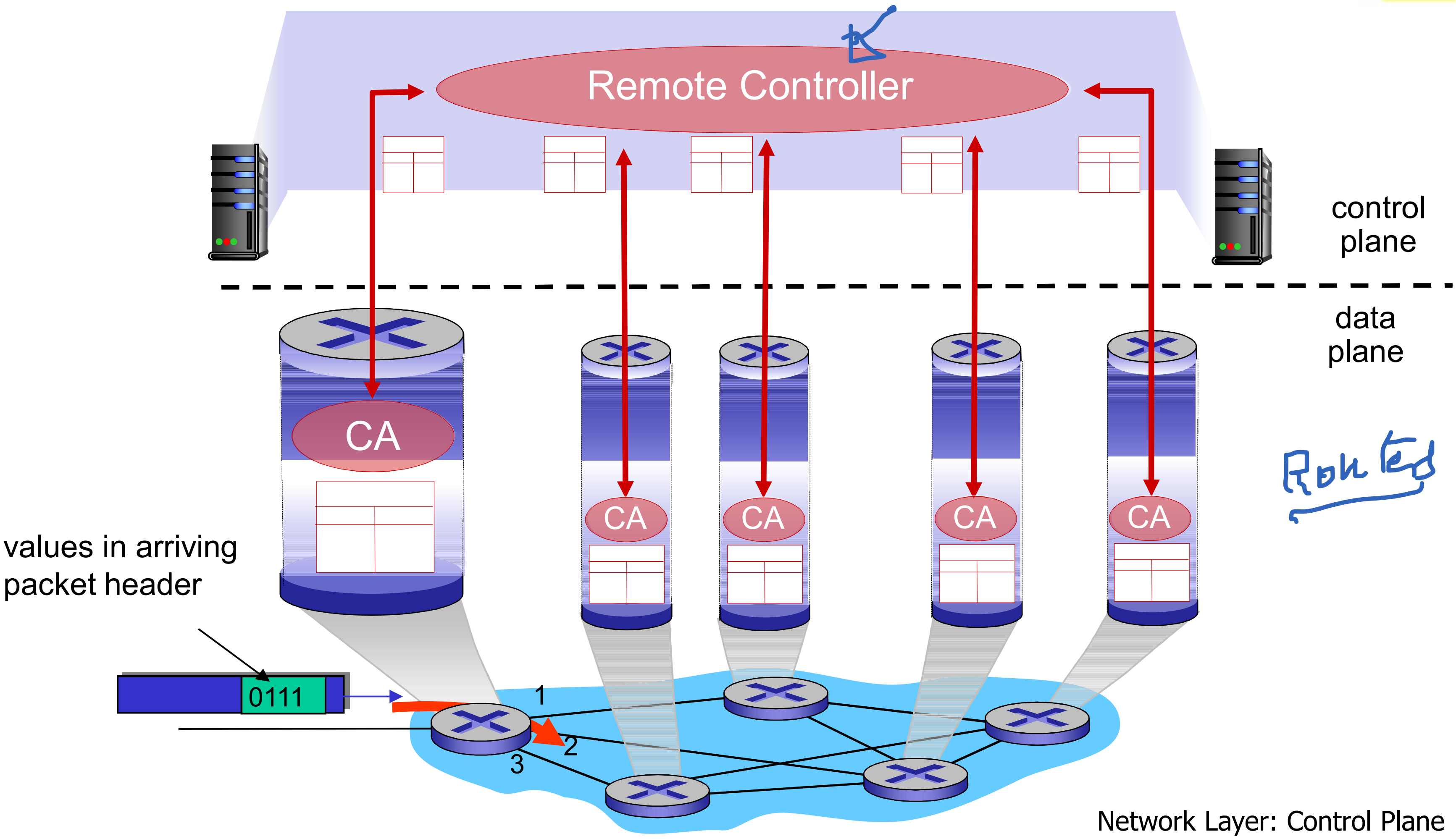
Logically centralized control plane

$\leftarrow$ SDN

Routing Prot

Distributed

A distinct (typically remote) controller interacts with local control agents (CAs)



Network service model

Q: What *service model* for "channel" transporting datagrams from sender to receiver?

example services for individual datagrams:

- guaranteed delivery
- guaranteed delivery with less than 40 msec delay

example services for a flow of datagrams:

- in-order datagram delivery
- guaranteed minimum bandwidth to flow
- restrictions on changes in inter-packet spacing

Network layer service models:

Network Architecture	Service Model	Guarantees ? Bandwidth	Loss	Order	Timing	Congestion feedback
Internet	best effort	none	no	no	no	no (inferred via loss)
ATM	CBR	constant rate	yes	yes	yes	no congestion
ATM	VBR	guaranteed rate	yes	yes	yes	no congestion
ATM	ABR	guaranteed minimum	no	yes	no	yes
ATM	UBR	none	no	yes	no	no

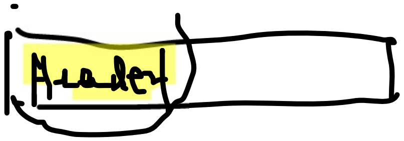
Chapter 4: outline

Forwarding.

Router Architecture

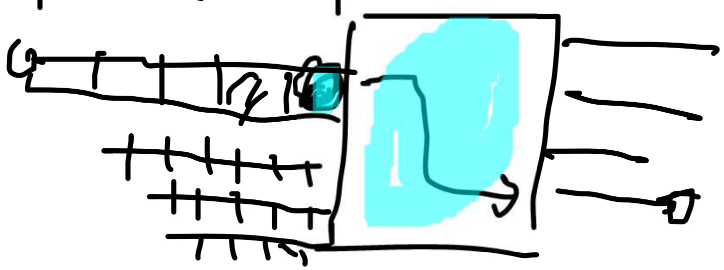
Routing Table exists

pkt:



dest. Addr	Output port #
x	#
y	#

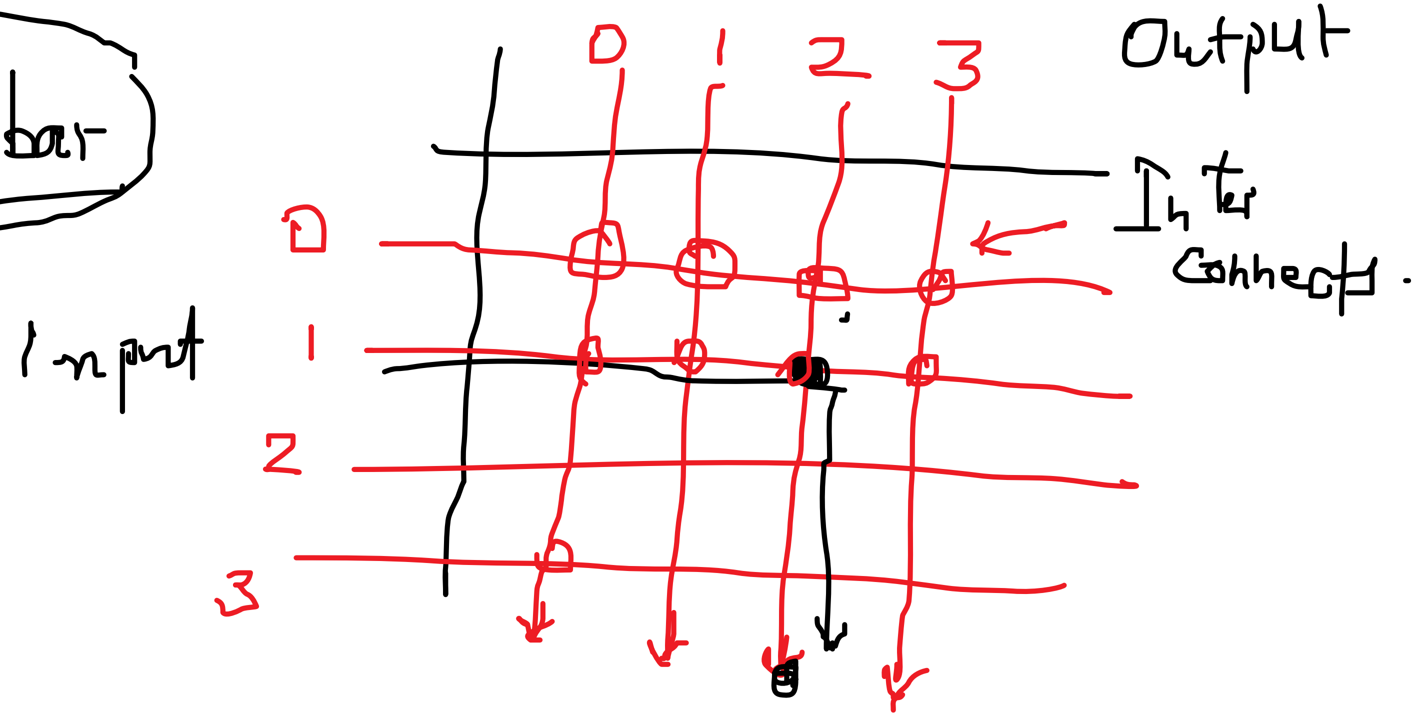
input queue/buff



Scheduler: decides which pkt needs to be forwarder

FCFS:

Cross bar



Interconnects · programmable gates (on/off)

open / closed

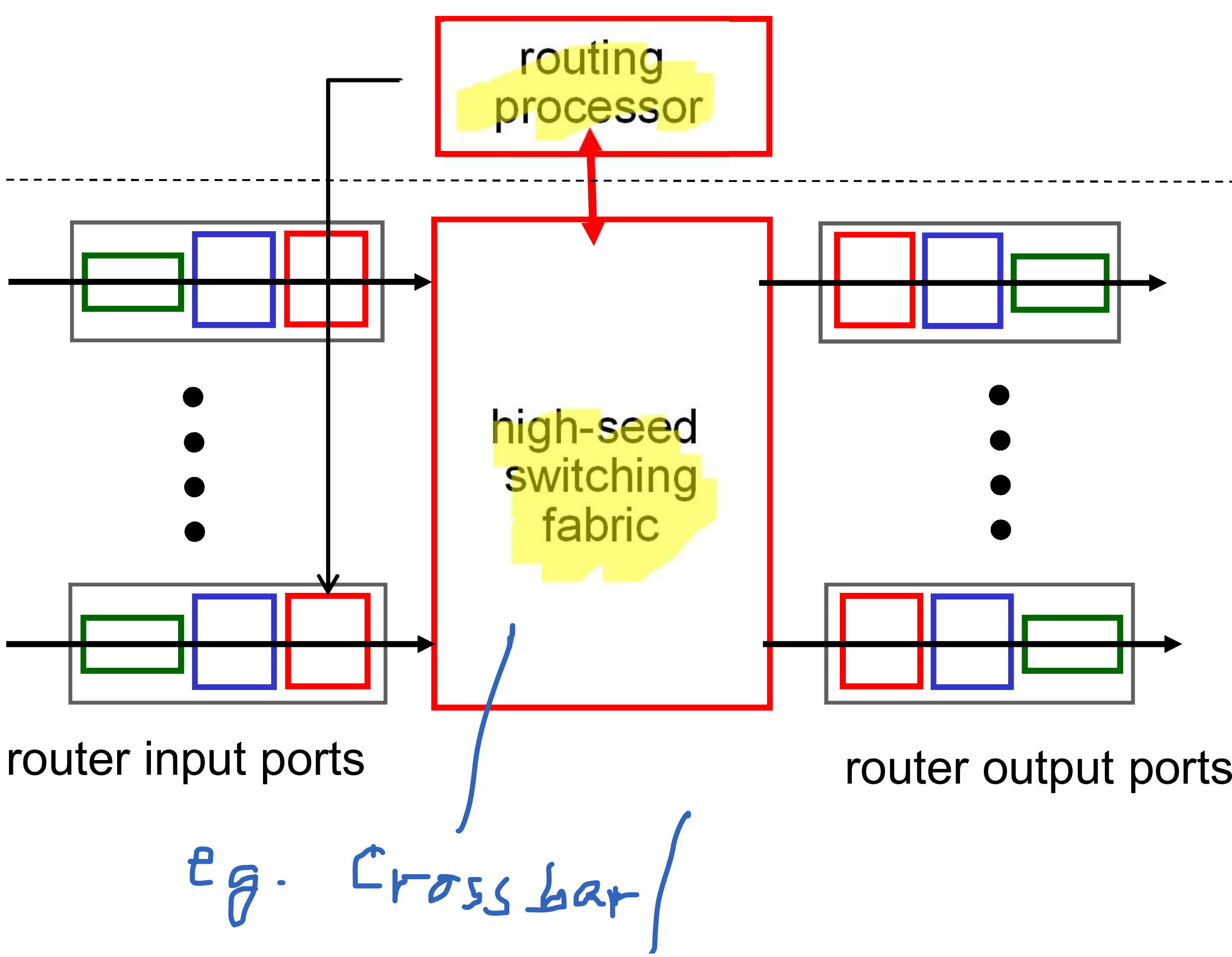
pkt at (in = 1, out=2).

close (1, 2) gate

open all other

Router architecture overview

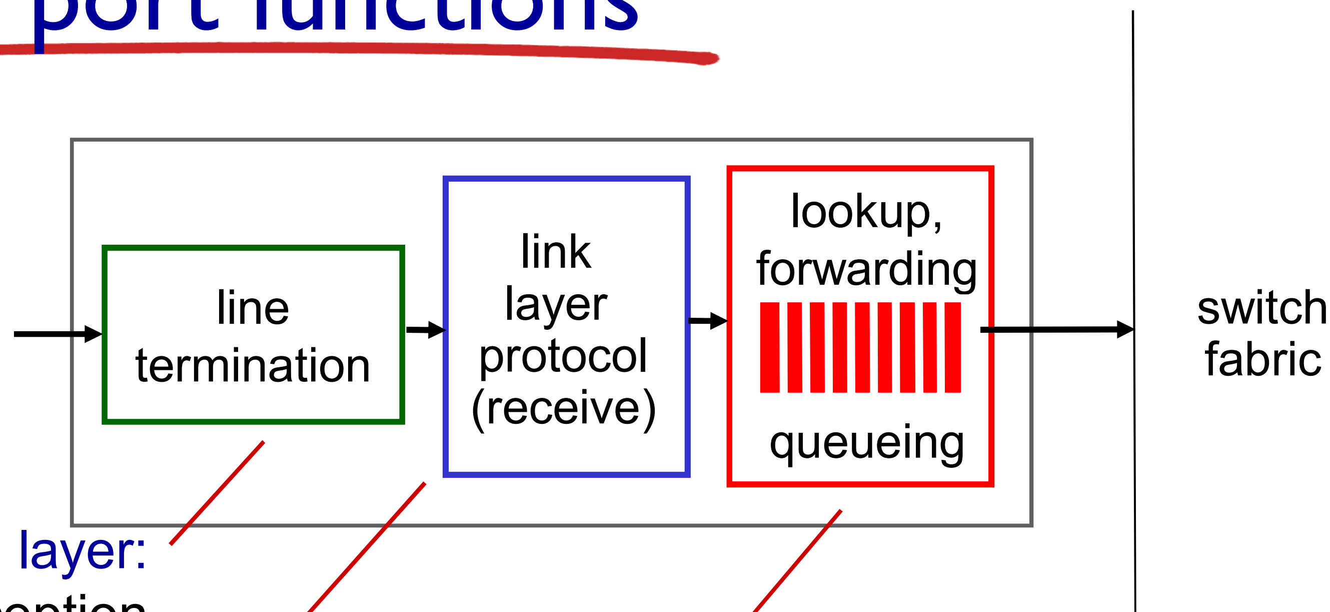
- high-level view of generic router architecture:



routing, management control plane (software) operates in millisecond time frame

forwarding data plane (hardware) operttes in nanosecond timeframe

Input port functions



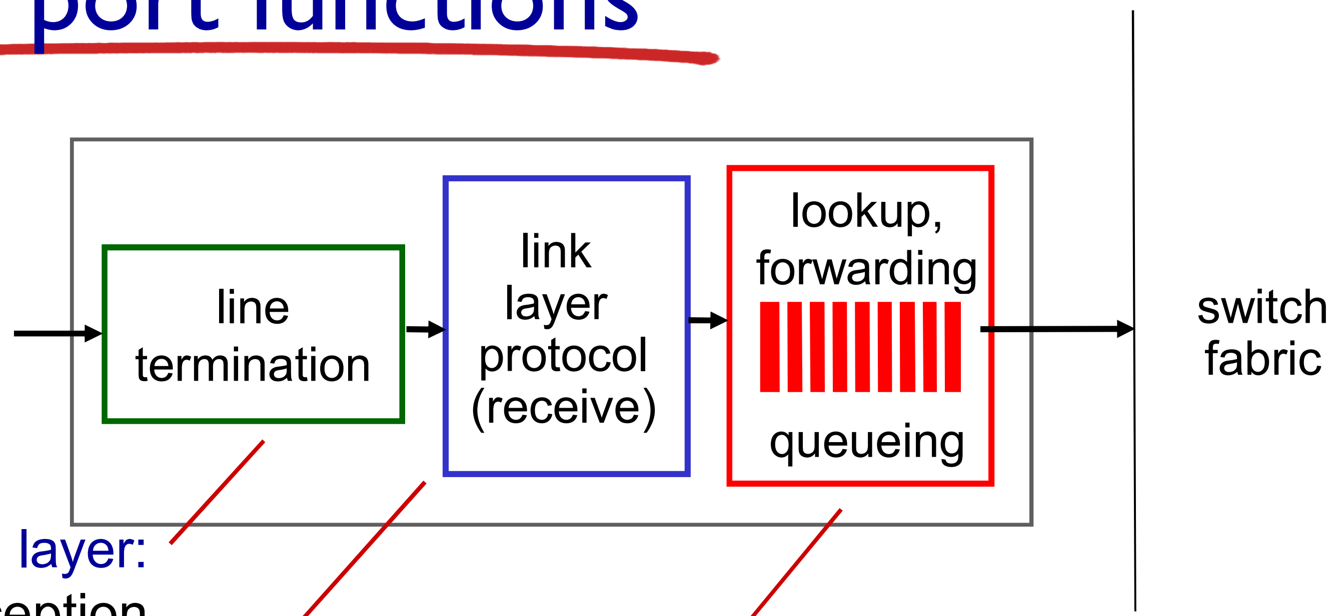
physical layer:
bit-level reception

data link layer:
e.g., Ethernet
see chapter 5

decentralized switching:

- using header field values, lookup output port using forwarding table in input port memory *("match plus action")*
- goal: complete input port processing at 'line speed'
- queuing: if datagrams arrive faster than forwarding rate into switch fabric

Input port functions



physical layer:
bit-level reception

data link layer:
e.g., Ethernet
see chapter 5

decentralized switching:

- using header field values, lookup output port using forwarding table in input port memory *("match plus action")*
- *destination-based forwarding:* forward based only on destination IP address (traditional)
- *generalized forwarding:* forward based on any set of header field values

1 byte 202 105 . 10 . 210 " 32 bit

Destination-based forwarding

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

Q: but what happens if ranges don't divide up so nicely?

202 . 105 . * . * → 1

Failure.

Routing Table design: Global: All routers talk to each other

↳ routes along the ways is working.

↳

→ periodically check that your neigh. routers are alive.

Longest prefix matching

longest prefix matching

when looking for forwarding table entry for given destination address, use *longest* address prefix that matches destination address.

Destination Address Range	Link interface
11001000 00010111 00010*** *********	0
11001000 00010111 00011000 *********	1
11001000 00010111 00011*** *********	2
otherwise	3

examples:

DA: 11001000 00010111 00010110 10100001 which interface?

DA: 11001000 00010111 00011000 10101010 which interface?

Longest prefix matching

- we'll see *why* longest prefix matching is used shortly, when we study addressing
- longest prefix matching: often performed using ternary content addressable memories (TCAMs)
  - *content addressable:* present address to TCAM: retrieve address in one clock cycle, regardless of table size
  - Cisco Catalyst: can up ~1M routing table entries in TCAM