

Computer Networks

COL 334/672

Data Plane

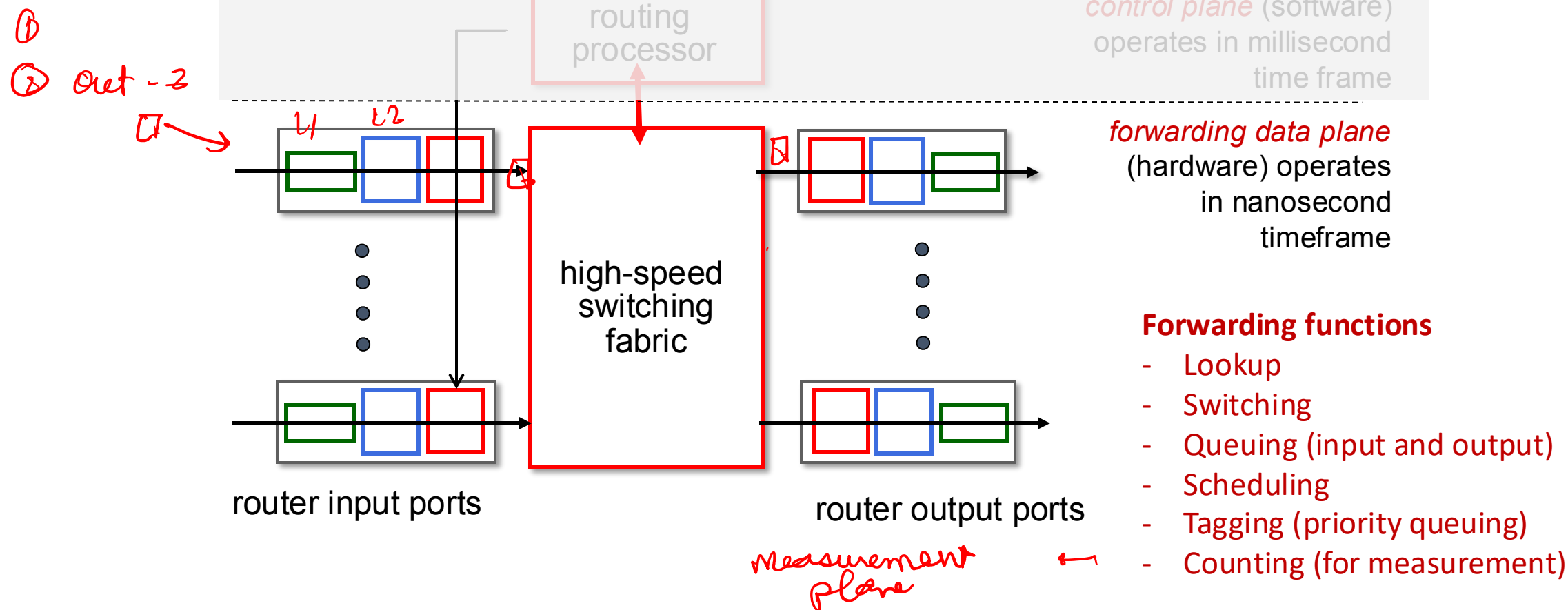
Tarun Mangla

Slides adapted from KR

Sem 1, 2024-25

Recap

Network Layer: Data Plane



Lookup

- **Problem statement:** Determine the output port based on destination IP lookup in the **forwarding table**
- How does the forwarding table look like?

Dst Addr Range	Output port
1100101xxxxx	0
1100101010xx	1
1100101011xx	2
otherwise	3

(Handwritten red annotations: A bracket on the left groups the first three rows. A bracket on the right groups the first three rows and the 'otherwise' row. A handwritten IP address 11001010000001 is shown next to the table.)

How do we do destination lookup in such a table?

Lookup: when looking for forwarding table entry for given destination address, use **longest** address prefix that matches destination address called **longest prefix matching**

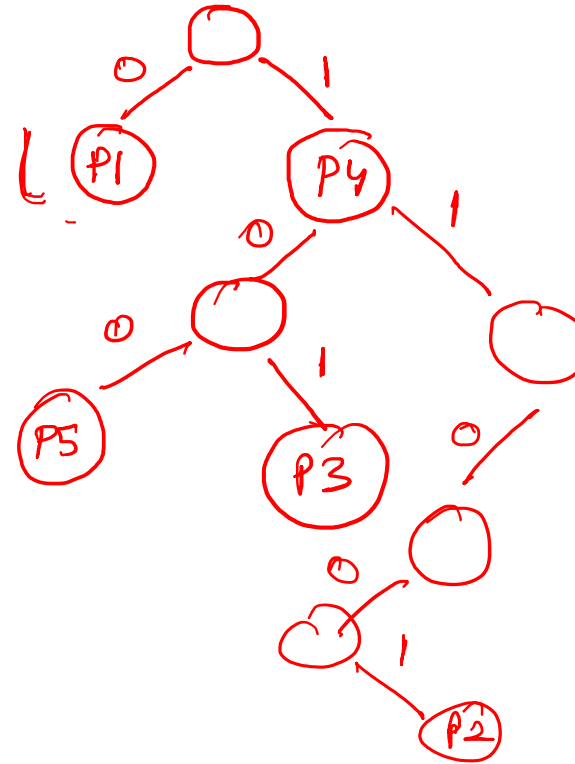
Lookup Techniques

- **Brute force:** Search through all forwarding table entries iteratively
 - Too slow!
- **Goal:** Minimize the number of memory lookups
- **Solution:** Use Trie or Prefix tree

Example: Unibit Tries

HW:
2-bit
trie

Pref	Output
0*	P1
1*	P4
101*	P3
100*	P5
11001*	P2



How many memory lookups?

Can we do better? *Multibit trie → Variable stride trie*

What is the trade-off?

Still better?

Content Addressable Memory

■ Content Addressable Memory (CAM):

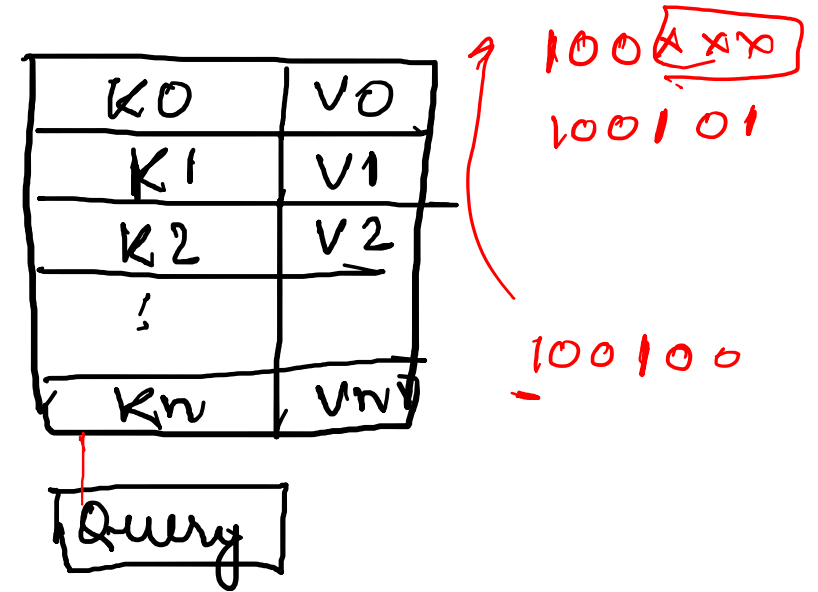
- Returns address of matched data
- $O(1)$ lookup

■ Binary CAM

- Search words consists of only 0s and 1s
- Would this suffice for longest prefix match?

■ Ternary CAM

- Allows a third matching state of X
- ➔ • Cisco Catalyst: ~1M routing table entries in TCAM
- Caution: TCAM is expensive, bulkier, power hungry. Used for high-end routers

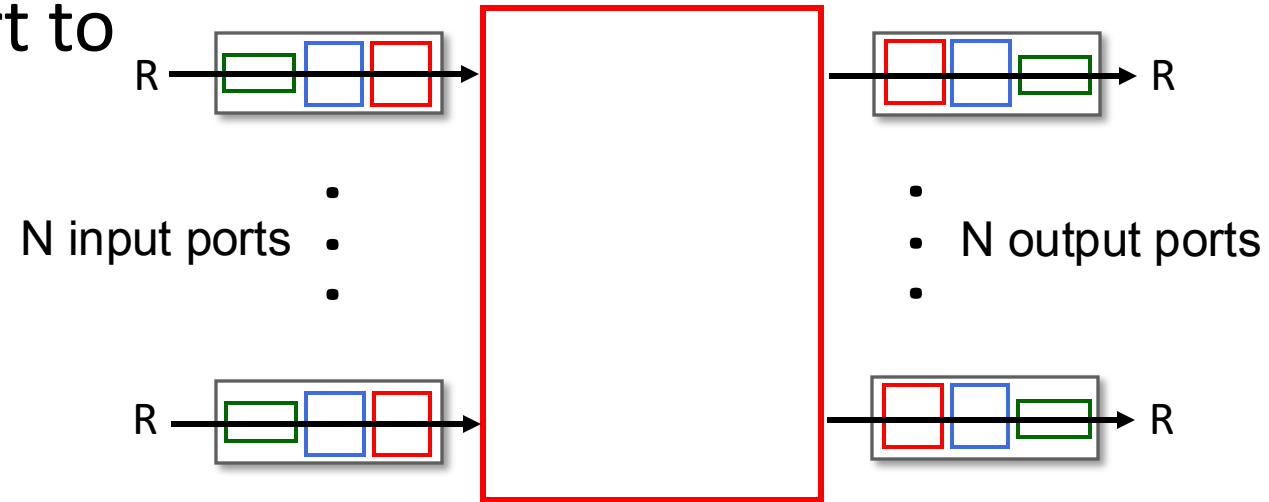


Data Plane Functions

- Prefix lookup
- **Switching**
- Queuing
- Scheduling

Switching

- Transfer packets from input port to appropriate output port
- **Switching rate:** rate at which packets can be transfer from inputs to outputs
- Often measured as multiple of input/output line rate
- What is the desired switching rate in this case?
- Switching rate depends on switching fabric and switching algorithm at the router

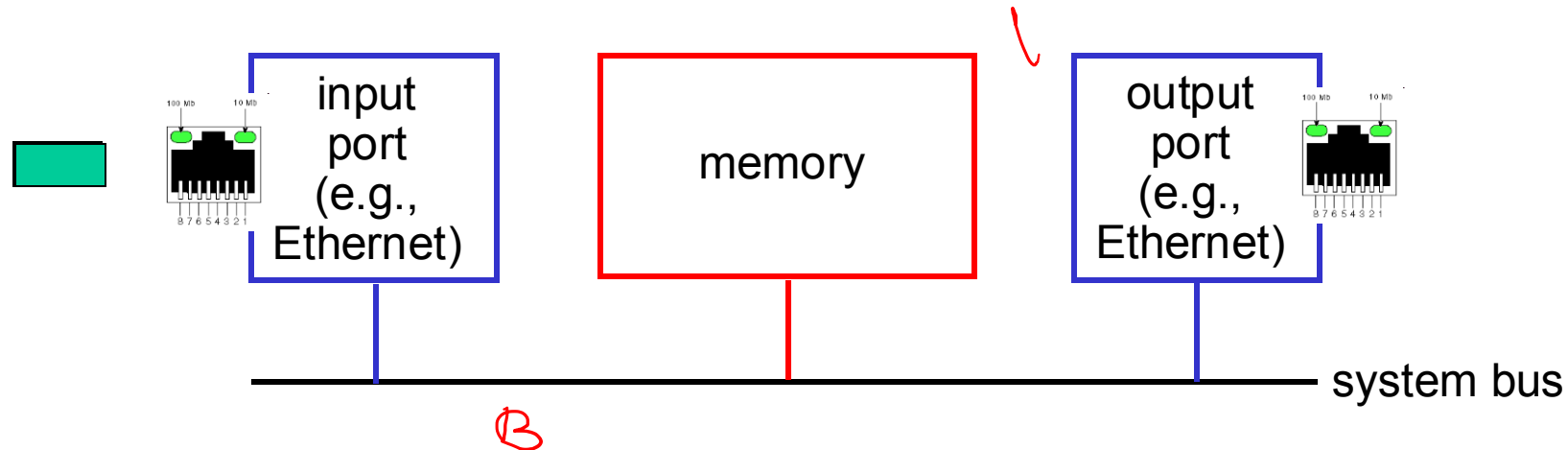


Switching via memory

first generation routers:

- traditional computers with switching under direct control of CPU
- packet copied to system's memory
- speed limited by memory bandwidth (2 bus crossings)
- What is the switching rate if the bus rate is B?

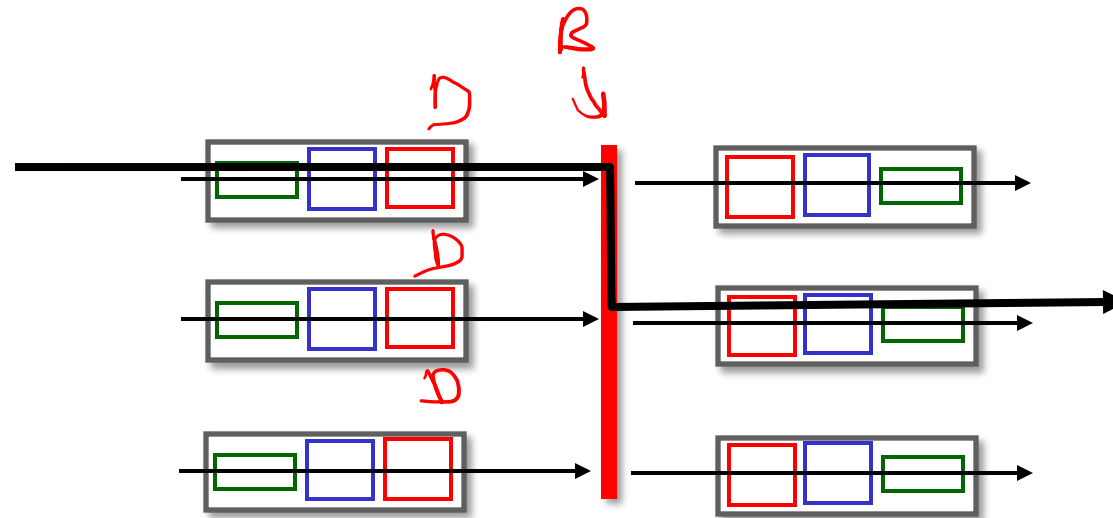
$B/2$



Switching via a bus

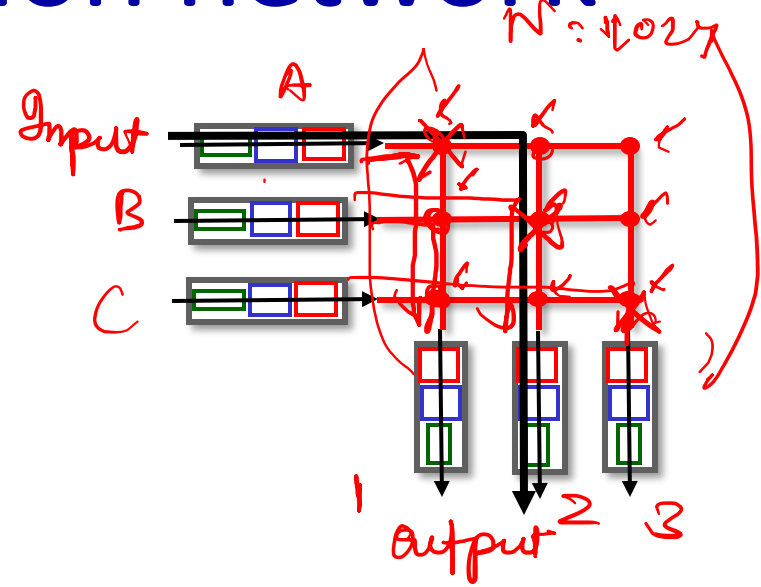
ASIC : Application Specific
Integrated Circuit

- datagram from input port memory to output port memory via a shared bus
- *bus contention*: switching speed limited by bus bandwidth
- 32 Gbps bus, Cisco 5600: sufficient speed for access routers



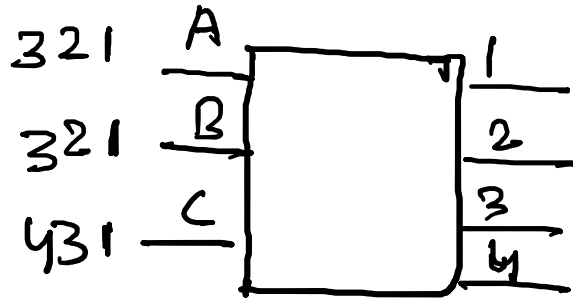
Switching via interconnection network

- Crossbar initially developed to connect processors in multiprocessor
- Need to develop efficient switching algorithms

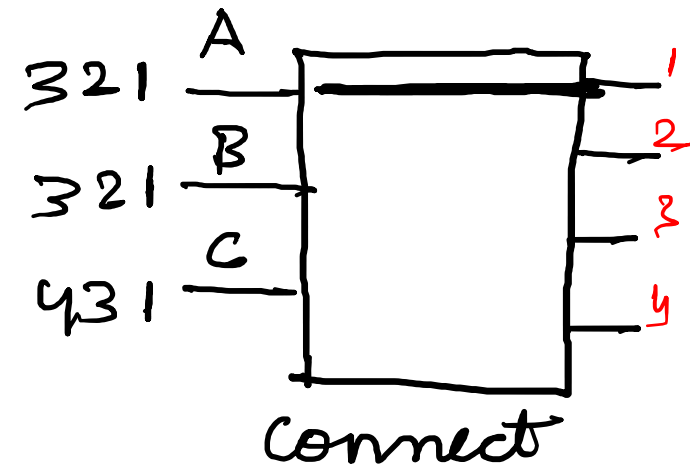
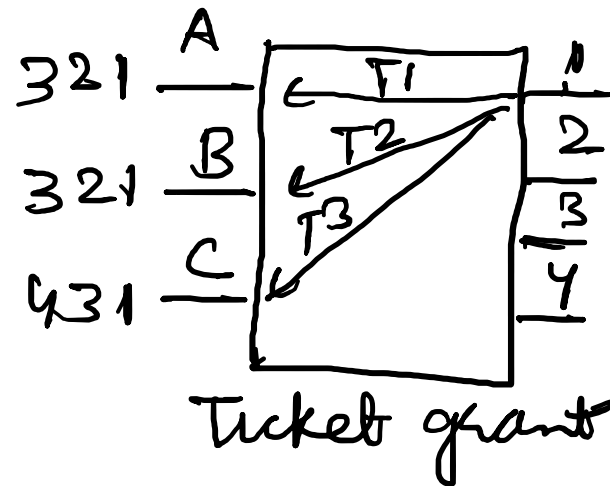
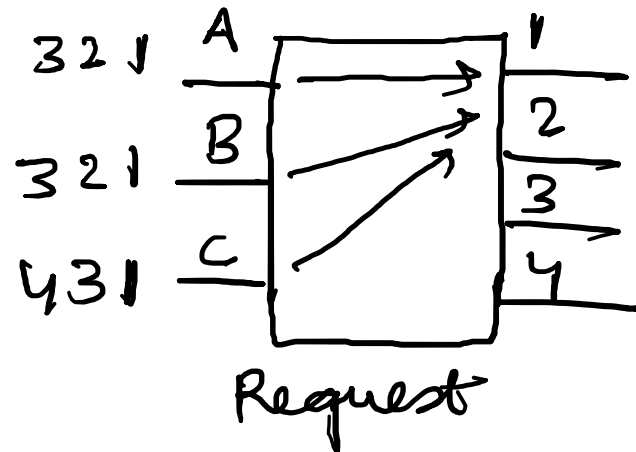


Switching algorithm

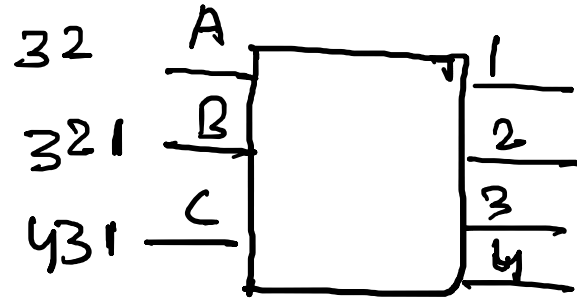
- **Take a ticket mechanism:** Use a simple ticket number mechanism for scheduling packets waiting at input port



Round 1



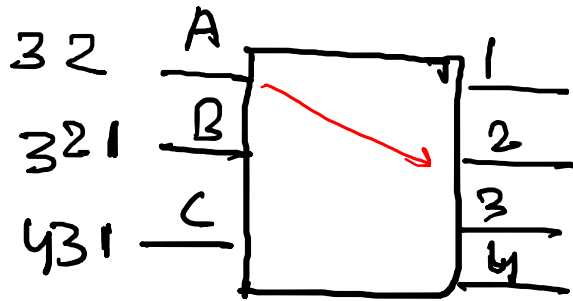
Take a Ticket Mechanism



time (in packet times)

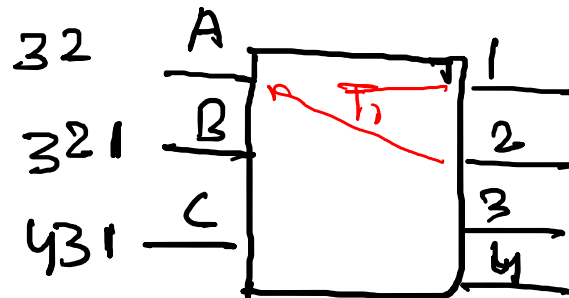
1	A	B	C			
2		A	B			
3			A	B	C	
4						C

Round 2

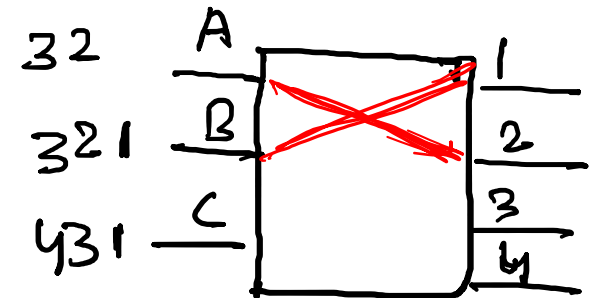


Request

Can we do better?



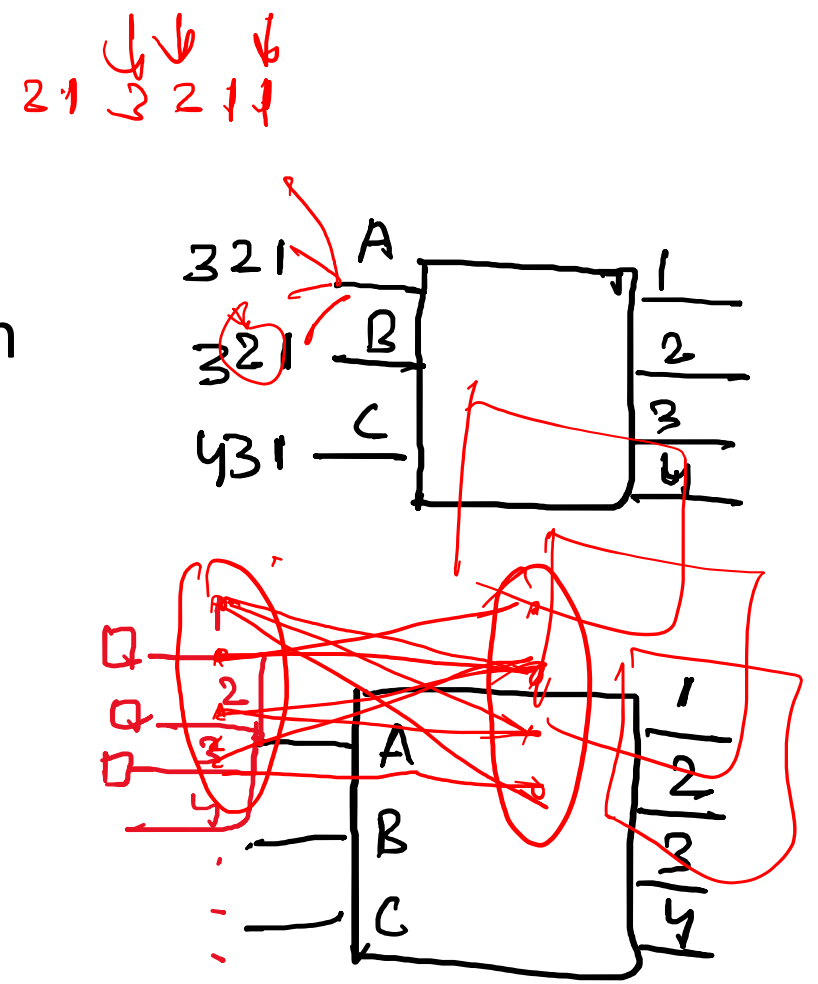
grant



connect

Input port queuing

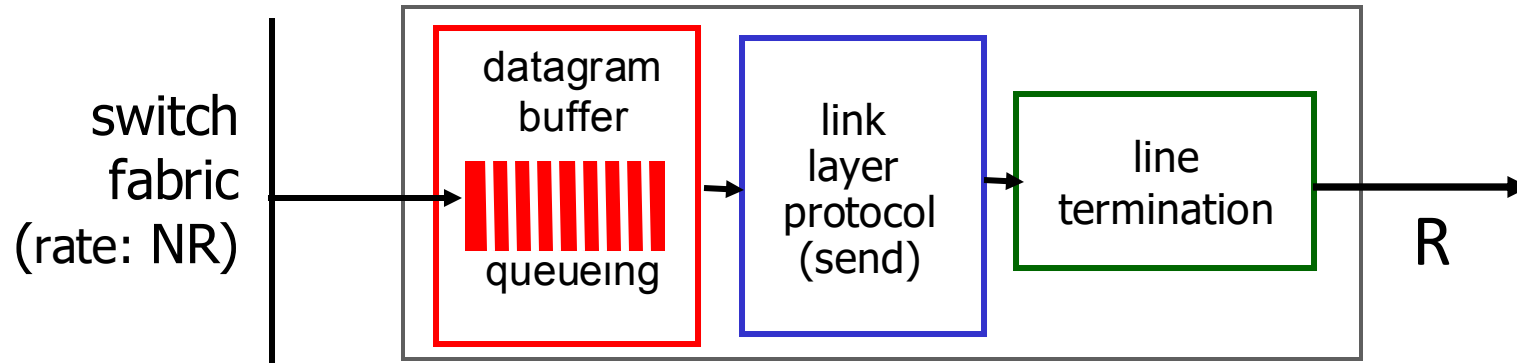
- **Head-of-the-Line (HOL) blocking:** queued datagram at front of queue prevents others in queue from moving forward
- Use **virtual output queuing**
- This reduces to **maximal bipartite matching** problem
- Known algorithms to solve it optimally
 - However, too slow!
 - Approximation algorithms: E.g., Parallel Iterative Match , iSLIP
- Can we still do better?
 - Hardware optimization: use multiple switching planes in parallel



Data Plane Functions

- Prefix lookup
- Switching
- **Queuing**
- Scheduling

Output port queuing



- *Buffering* required when datagrams arrive from fabric faster than link transmission rate, otherwise datagrams will get lost

How much buffering?

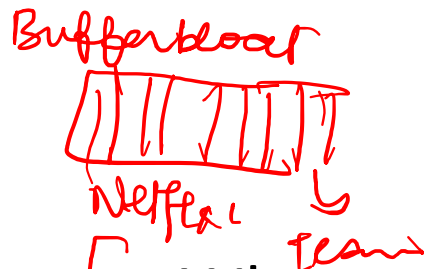
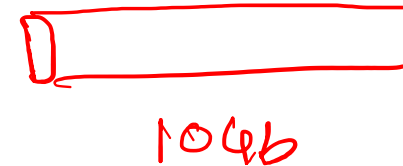
Bandwidth Delay Product

- RFC 3439 rule of thumb: average buffering equal to “typical” RTT (say 250 msec) times link capacity C
 - e.g., C = 10 Gbps link: 2.5 Gbit buffer

for a larger value of N

- more recent recommendation: with N flows, buffering equal to

$$\frac{RTT \cdot C}{\sqrt{N}}$$



- Why not simply use large buffers?
- but *too* much buffering can increase delays (particularly in home routers)
 - long RTTs: poor performance for real-time apps, sluggish TCP response