Computer Networks COL 334/672

Switched Ethernet

Tarun Mangla

Sem 1, 2025-26

Switched Ethernet

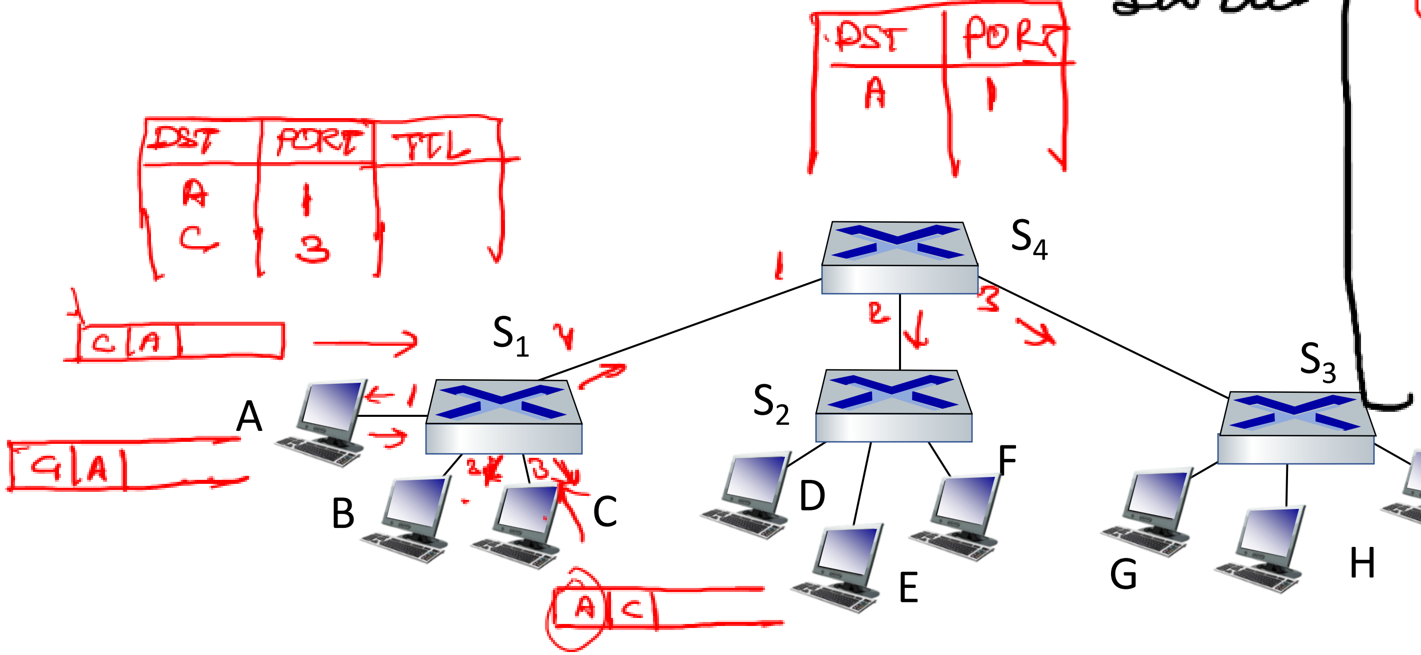
LLDP : Link layer discovery protocol

Challenges w/ self-learning switch

1. Scale
2. Moving hosts
   - ↳ A announces whenever it ~~joins~~ connect to a new switch
   - ↳ TTLs for inactive hosts

Forward

Routing



DST	PORT	TTL
A	1	
C	3	

DST	PORT
A	1

- Switched network reduces the collision domain
- L2 switches are typically self-learning

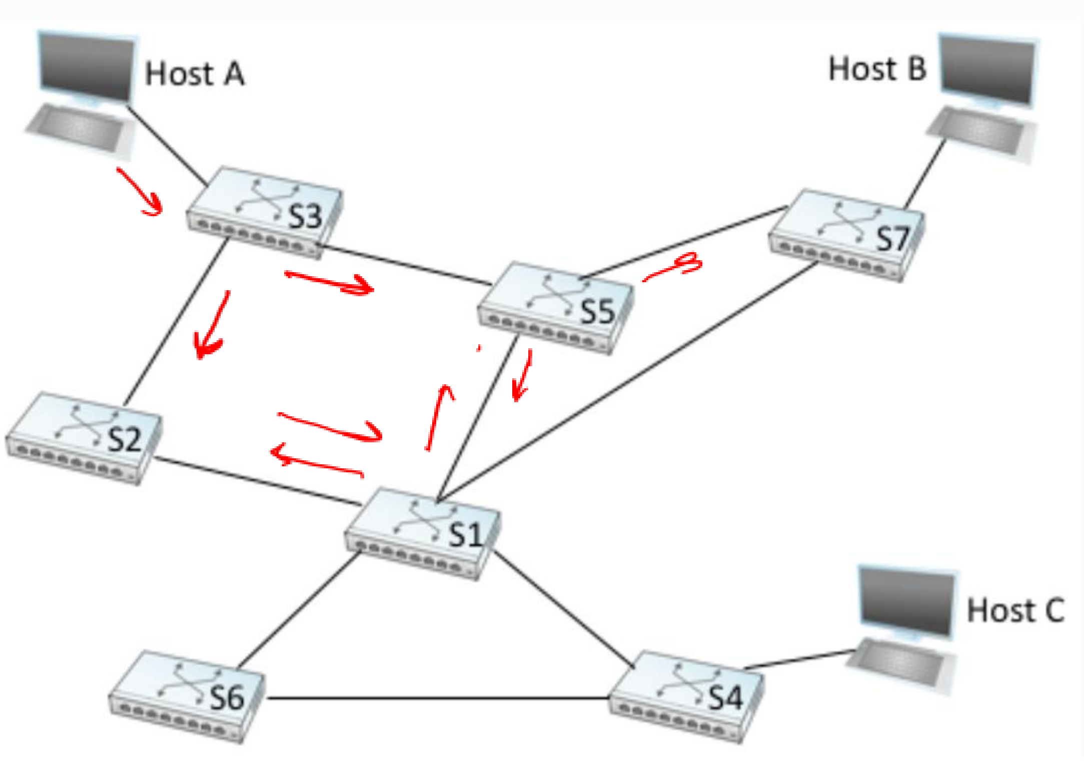
Designing distributed protocols is the key challenge

What happens in case of a loop?

- Why would switched Ethernet have loop?
  - Redundancy
- Packet loops are a waste of network bandwidth
- How do we prevent packets from looping?

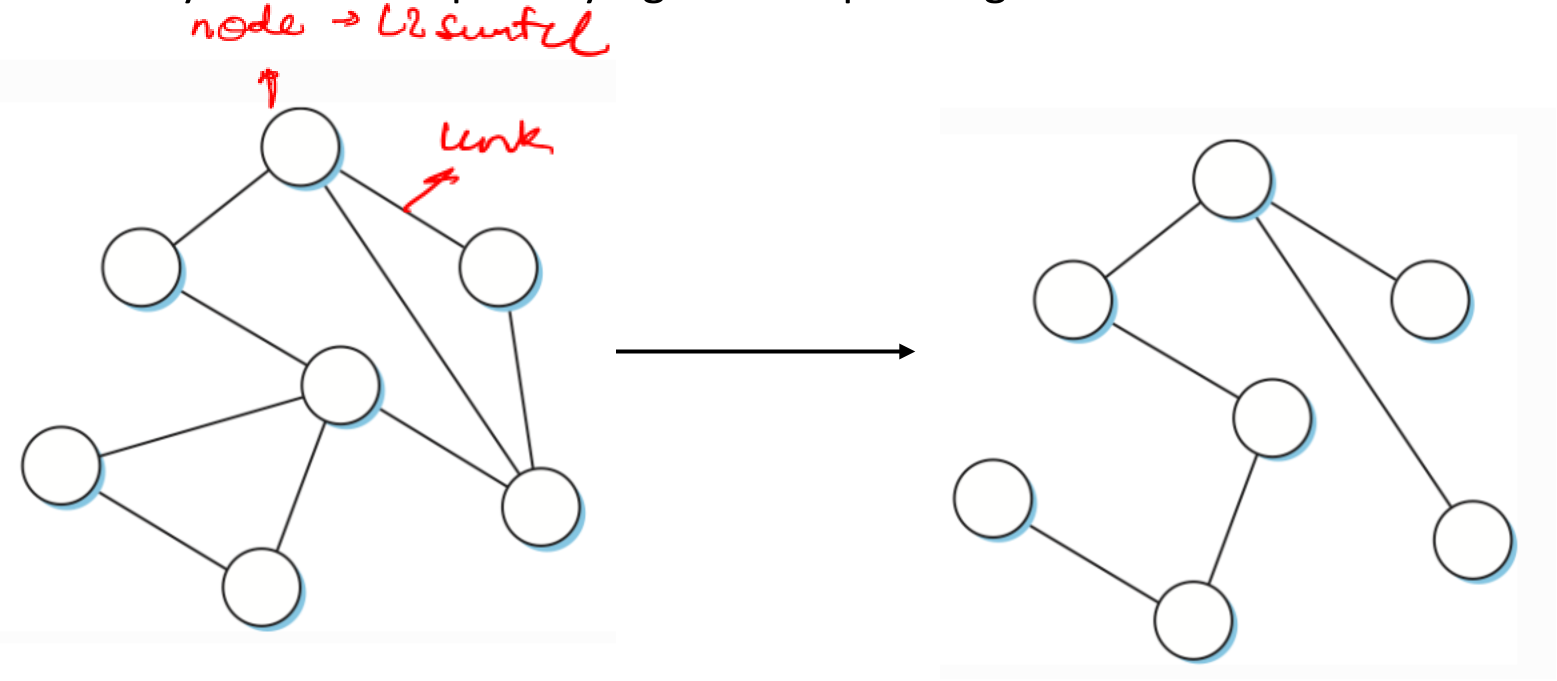
Sol #1: Use seq # & drop a packet w/ same seq #

Does not scale



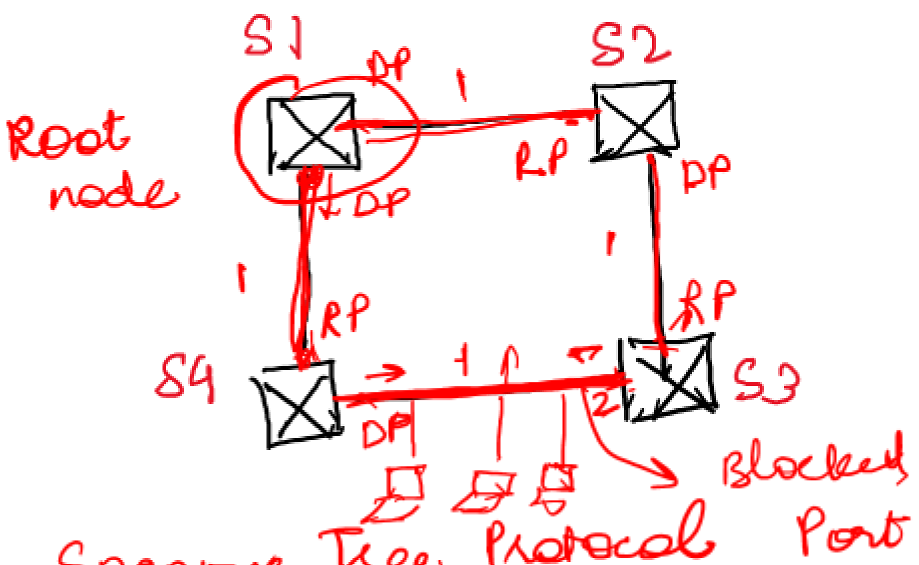
Break the Loop!

- **Idea**: Create a logical spanning tree in the network
  - Spanning tree: subgraph that covers all vertices but contains no cycles
  - Switches will only forward on ports lying on the spanning tree

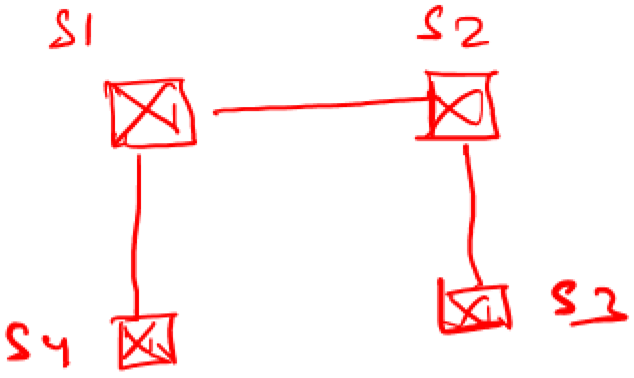


- **Challenge:** Need to create the same spanning tree in a distributed manner

How to generate the same spanning tree?



Spanning Tree Protocol



Assign a root node, e.g. switch with the lowest ID

→ Port → Root Port
if it is the upstream port that is closest to root node

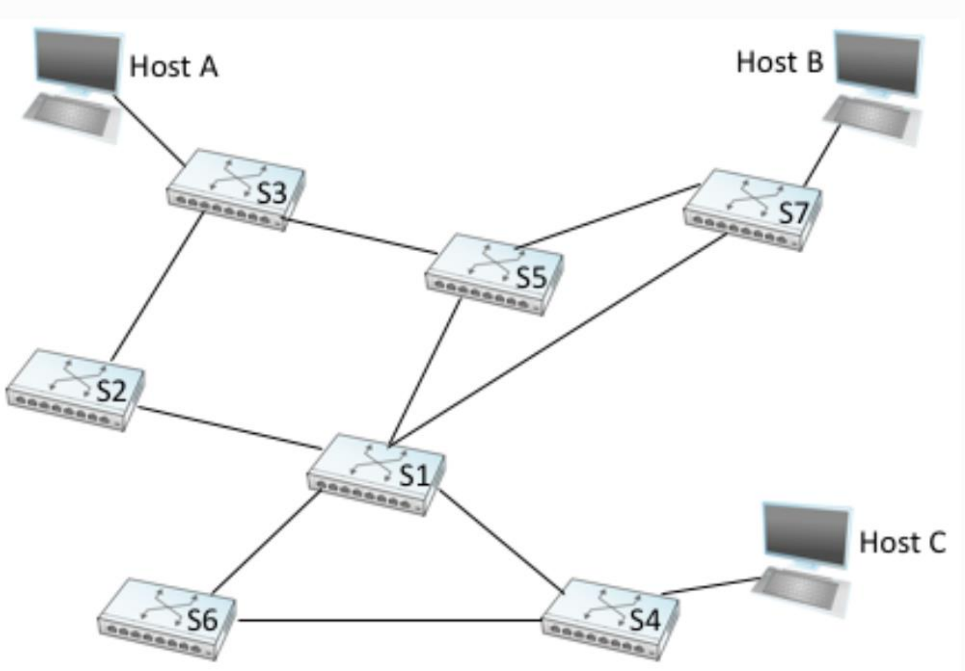
→ Designated Port : Port that is closest to root node.
assigned for all links

Any port that is not RP or DP, Block that port

Break ties: ① Lower root switch ID
② smaller distance
③ same distance, same root, smaller switch ID

Spanning Tree Protocol

- **Algorithm**
  - Elect the switch with smallest ID as the root of the spanning tree
  - Identify port that is closest to the root, root port
  - In case of a tie, select the port with smaller switch ID
  - For each link (LAN segment), assign the port closest to root node as the designated port
  - All other ports that are not either root port or designated port are disabled



- How is it exactly done?

SPT (Details)

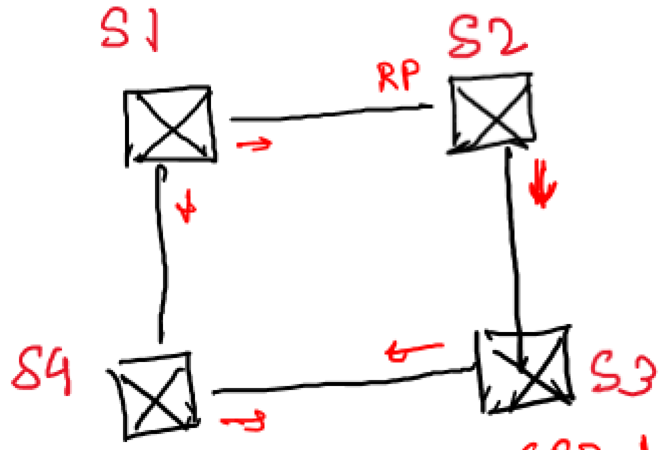
- Each bridge has an ID
  - 8 bytes: 2 bytes configurable, 6 bytes of MAC address
- Switch X announces configuration messages:
  - (Y, d, X)

Root node

ID of the switch announcing the msg

distance to root node

- Initially, each switch thinks it is the root
- Stop generating own configuration messages
  - When receives message from a smaller switch ID



It 2: (S2, 1, S3)

It 1:

S1: (S1, 0, S1)

S2: (S2, 0, S2)

S3: (S3, 0, S3)

S4 (S4, 0, S4)

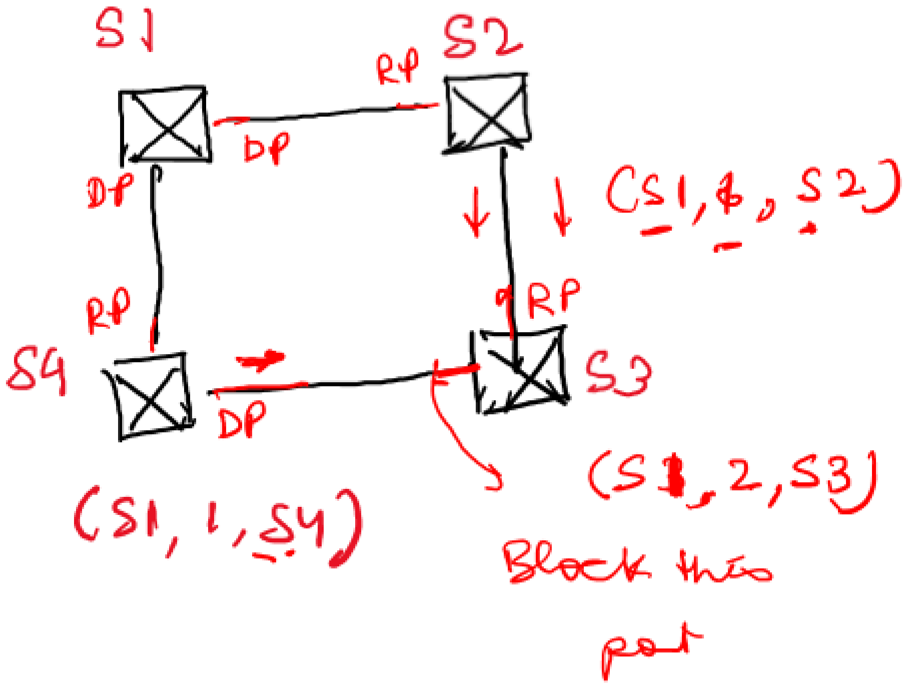
It 2:

S2: (S1, 1, S2) to S3

S4: (S1, 1, S4)

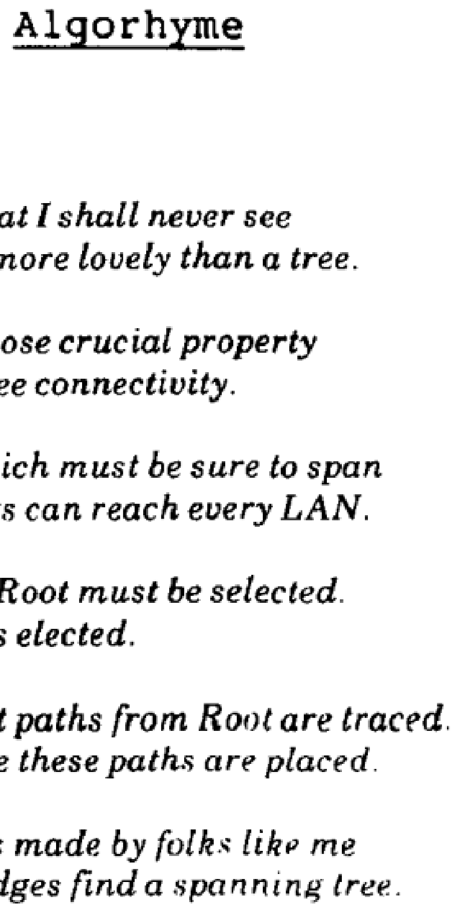
SPT (Continued)

- A received message is better:
  - It identifies a smaller root ID
  - Same root ID, but shorter distance
  - Same root ID, same distance, smaller switch ID
- Stop sending on a port from where the switch received a better message



Spanning Tree Protocol

- Fun fact: Invented by Radia Perlman from Digital Equipment Corporation (DEC)



Algorhyme

I think that I shall never see
A graph more lovely than a tree.

A tree whose crucial property
Is loop-free connectivity.

A tree which must be sure to span
So packets can reach every LAN.

First the Root must be selected.
By ID it is elected.

Least cost paths from Root are traced.
In the tree these paths are placed.

A mesh is made by folks like me
Then bridges find a spanning tree.

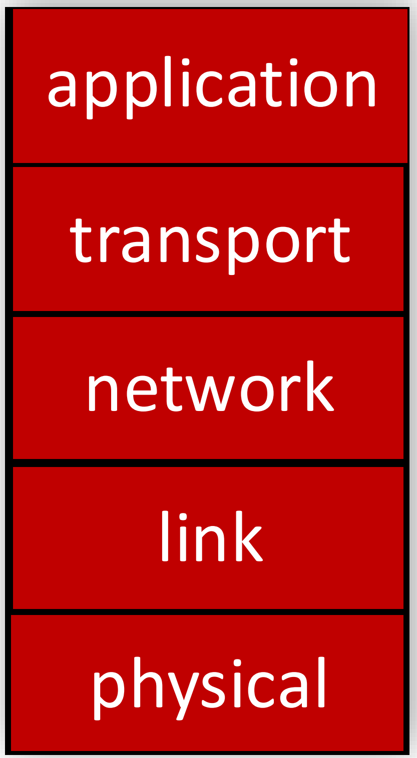
Summary: Link layer

- Link layer services
  - Encoding
  - Framing
  - Error detection
  - Medium Access Control (MAC)
- Ethernet protocol
- Forwarding and routing in an Ethernet switched network
- A lot of interesting things happening at L2
  - Virtual LAN (VLANs) / MPLS → circuit switching
  - Special networks: Data center networks, cellular networks ..

L2 routing

Layered Internet protocol stack

- *application:* supporting network applications
  - HTTP, IMAP, SMTP, DNS
- *transport:* process-process data transfer
  - TCP, UDP
- *network:* routing of datagrams from source to destination
  - IP, routing protocols
- *link:* data transfer between neighboring network elements
  - Ethernet, 802.11 (WiFi), PPP
- *physical:* bits “on the wire”



application

transport

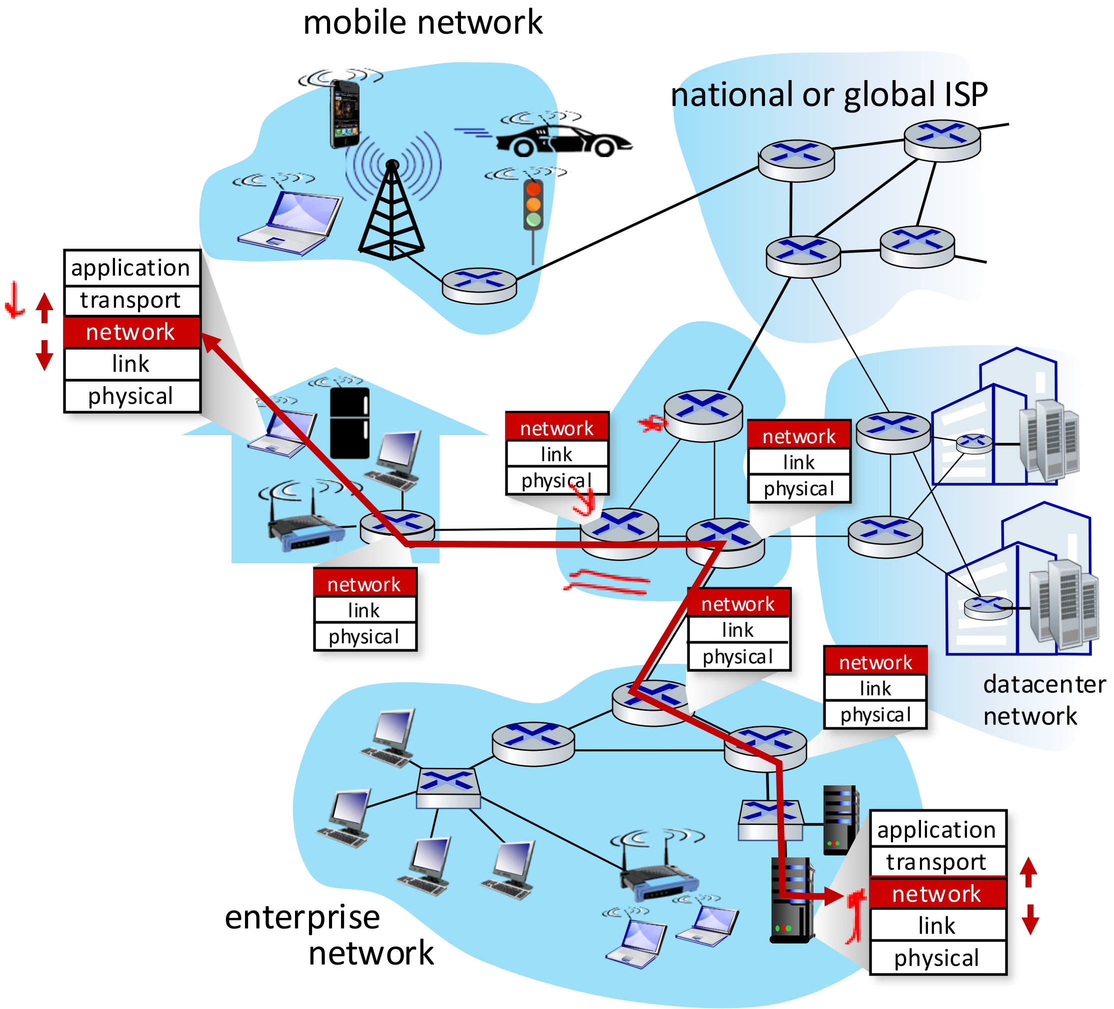
network

link

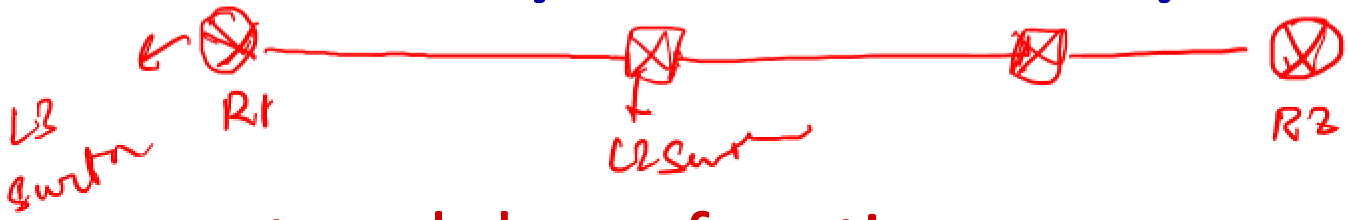
physical

Network-layer services and protocols

- deliver a segment from sending to receiving host
  - sender: encapsulates segments into datagrams, passes to link layer
  - receiver: delivers segments to transport layer protocol
- network layer protocols in *every Internet device*: hosts, routers
- Routers or L3 switches:
  - examines header fields in all IP datagrams passing through it
  - moves datagrams from input ports to output ports to transfer datagrams along end-end path



Two key network-layer functions



network-layer functions:

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

analogy: taking a trip

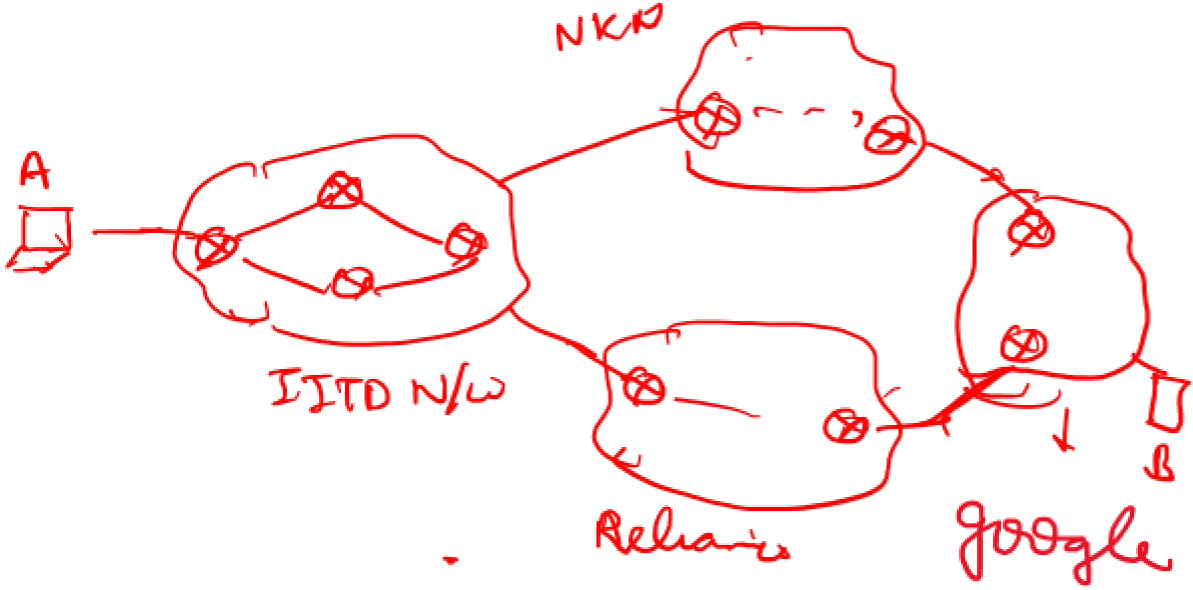
- *forwarding:* process of getting through single interchange
- *routing:* process of planning trip from source to destination

forwarding

routing

Routing Algorithm

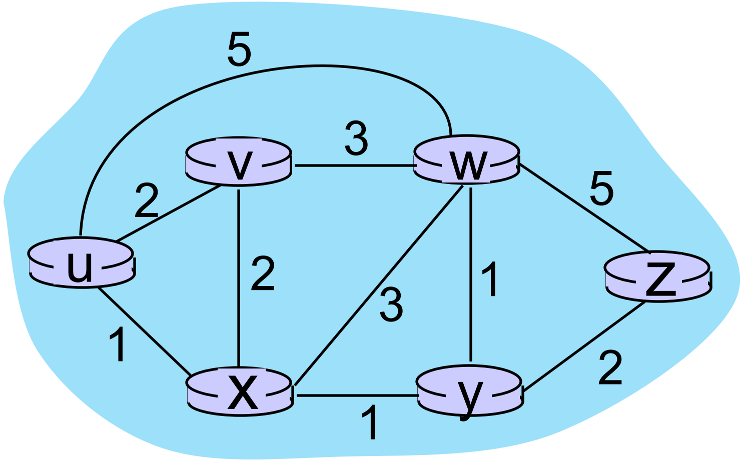
- **Goal:** determine "good" paths from sending host to receiving host through networks of routers
- **Good:** high throughput, low latency, or low cost (economic)
- Routing algorithm taxonomy:
  - **At what level:** intra-domain or inter-domain
  - **How:** centralized or distributed



Different intra-domain routing protocols

One inter-domain routing protocols

Intra-domain Routing: Graph Abstraction



$c_{a,b}$: cost of *direct* link connecting *a* and *b*

e.g., $c_{w,z} = 5$, $c_{u,z} = \infty$

cost defined by network operator: could always be 1, or inversely related to bandwidth, or inversely related to congestion

graph: *G = (N,E)*

N: set of routers = { *u, v, w, x, y, z* }

E: set of links ={ *(u,v), (u,x), (v,x), (v,w), (x,w), (x,y), (w,y), (w,z), (y,z)* }

How to determine shortest path from one node to all other nodes in a graph?