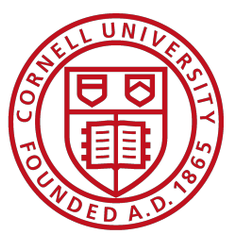
CS4410/11: Operating Systems

CPU Scheduling (Recap)

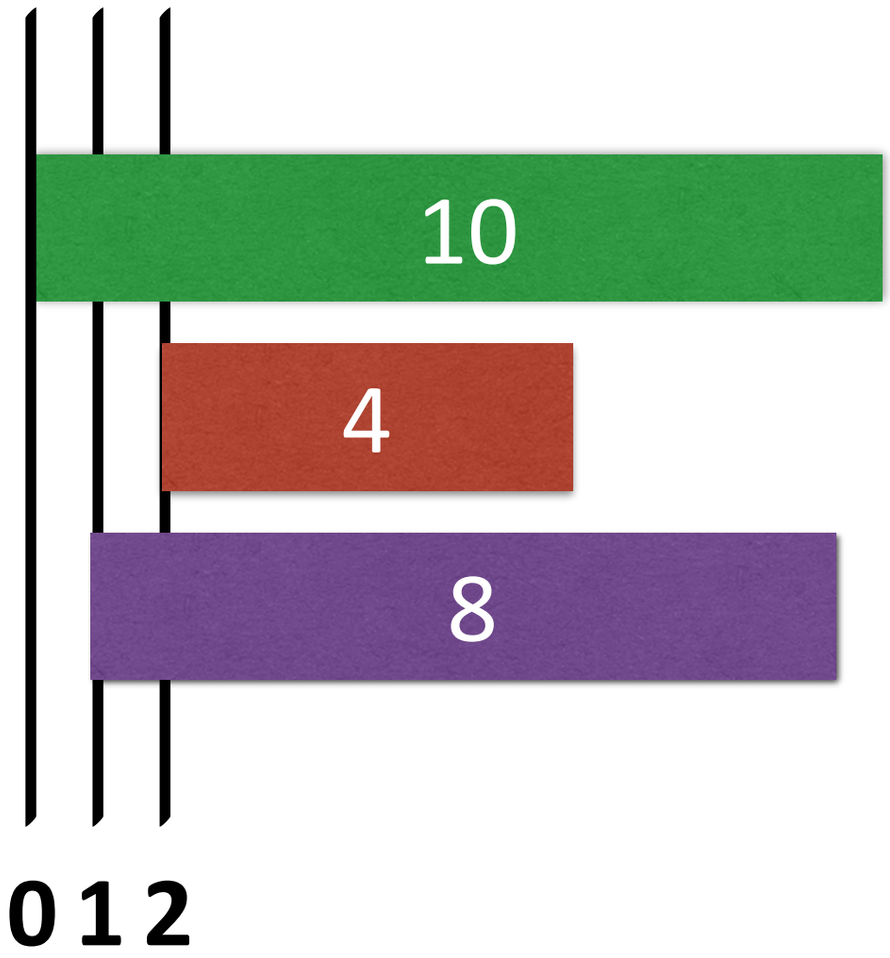
Networking

Rachit Agarwal
Anne Bracy

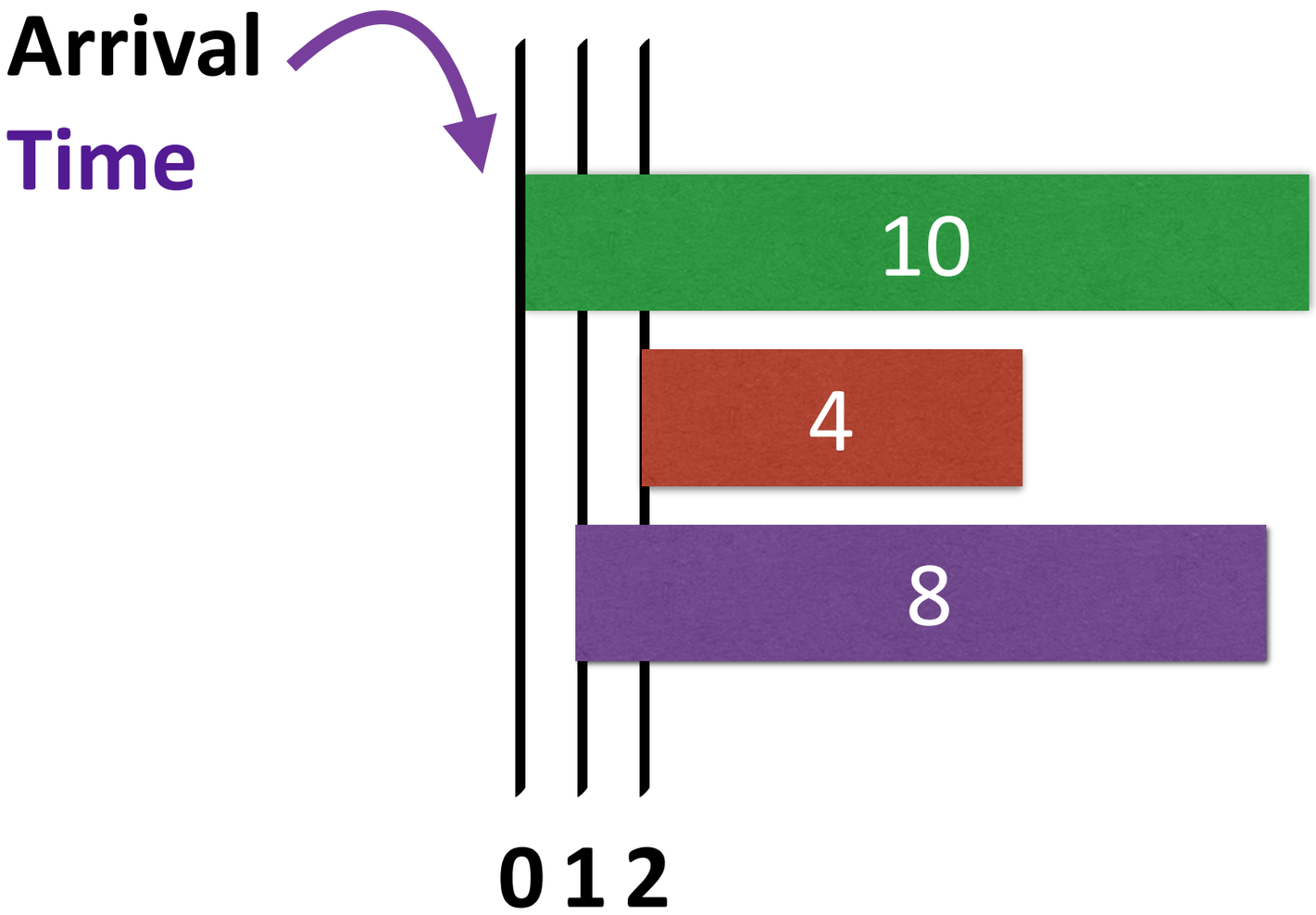
Slides based on material from Sirer, Rennesse, Rexford (Princeton)



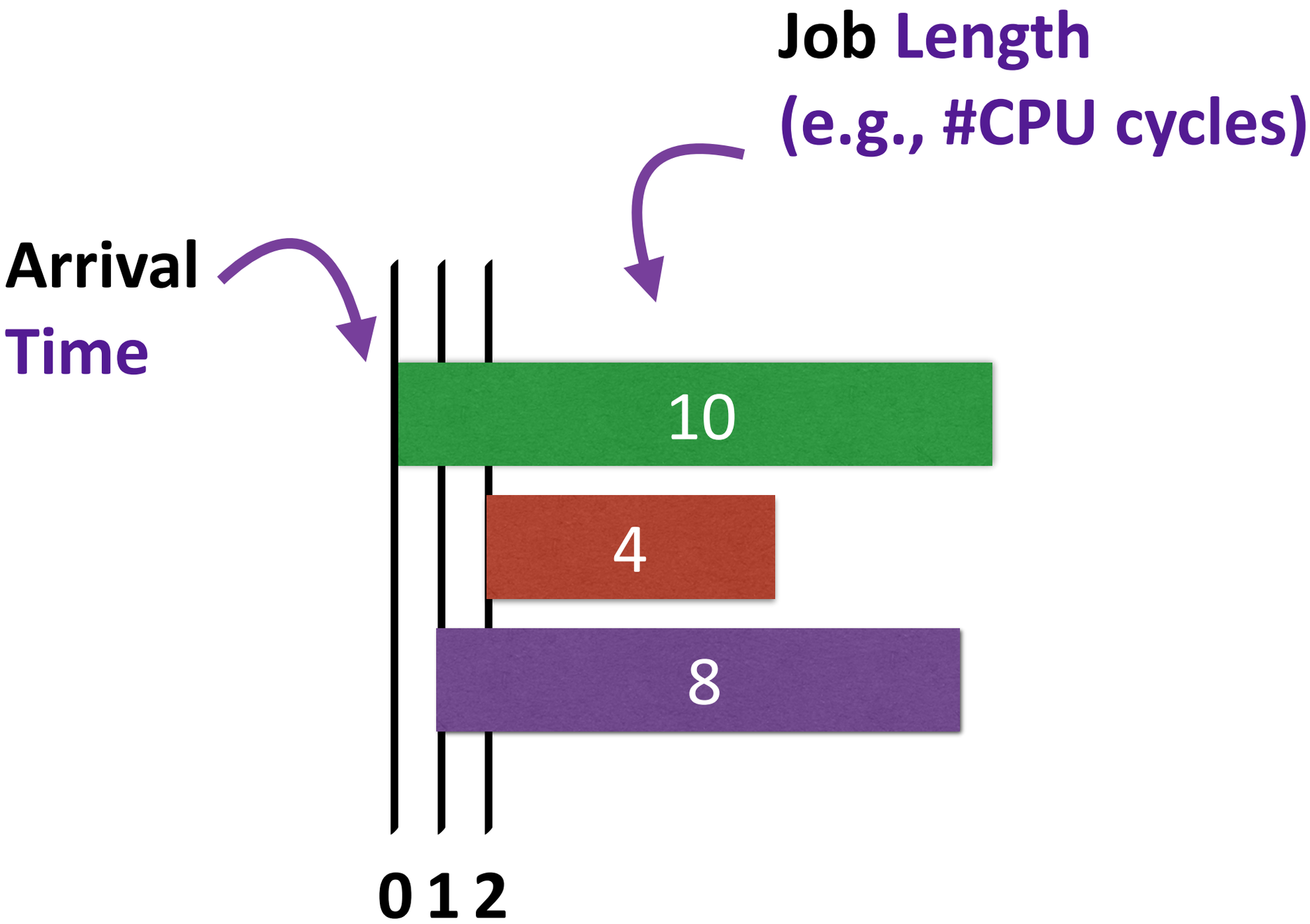
CPU Scheduling — Example



CPU Scheduling — Example



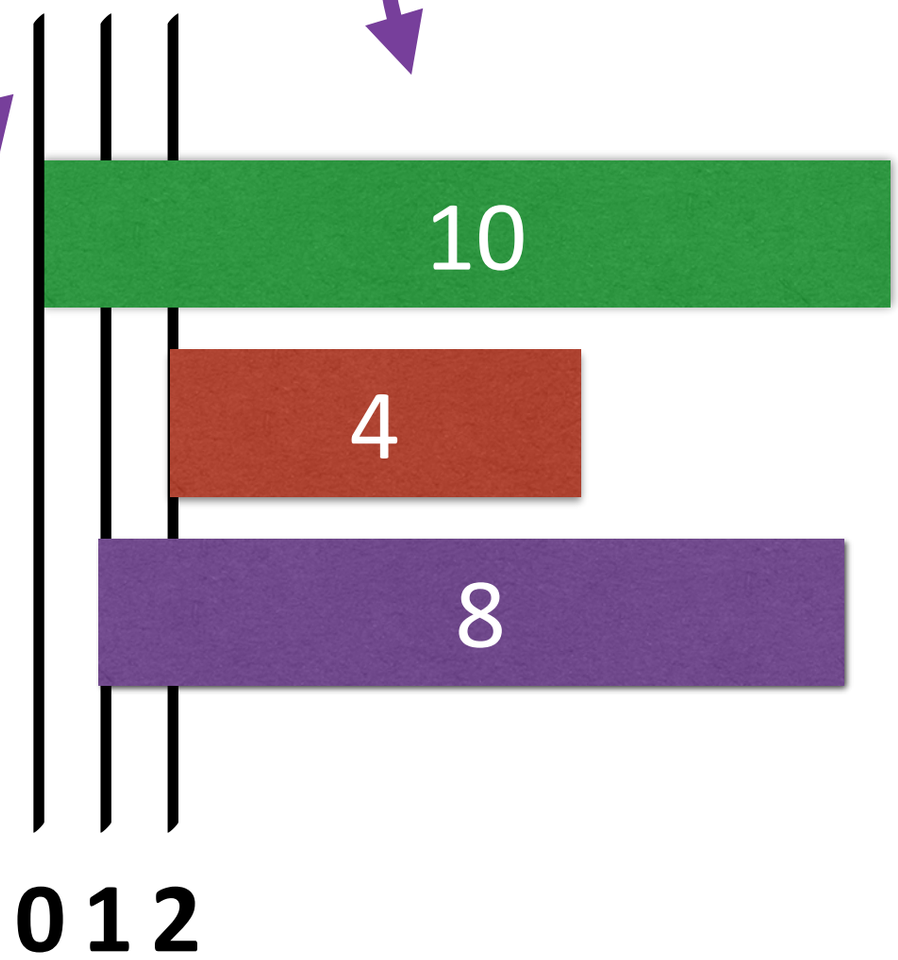
CPU Scheduling — Example



CPU Scheduling — Example

Job Length
(e.g., #CPU cycles)

Arrival Time



- FIFO
- LIFO
- SJF
- SRTF
- RR
- Priority

Networking — What is it about?

Networking — What is it about?

So far: focused on what happens on a “machine”!

Networking — What is it about?

So far: focused on what happens on a “machine”!

- **Networking**
    - How do machines communicate?

Networking — What is it about?

So far: focused on what happens on a “machine”!

- **Networking**
    - How do machines communicate?
- **Lets start with a simple analogy**
    - How to move stuff from München to Ithaca?

Networking — Key Concepts

Networking — Key Concepts

Four “concepts”!

Networking — Key Concepts

Four “concepts”!

- **Layering**
    - Abstraction is the key to manage complexity

Networking — Key Concepts

Four “concepts”!

- **Layering**
    - Abstraction is the key to manage complexity
- **Naming**
    - A name for each computer, protocol, ..

Networking — Key Concepts

Four “concepts”!

- **Layering**
    - Abstraction is the key to manage complexity
- **Naming**
    - A name for each computer, protocol, ..
- **Protocols**
    - Computers, network devices speaking the same language

Networking — Key Concepts

Four “concepts”!

- **Layering**
    - Abstraction is the key to manage complexity
- **Naming**
    - A name for each computer, protocol, ..
- **Protocols**
    - Computers, network devices speaking the same language
- **Resource Allocation**
    - Share resources (bandwidth, wireless spectrum, paths, …)

Networking — A Stack of Protocol Layers

Networking — A Stack of Protocol Layers

Five “layers”!

Networking — A Stack of Protocol Layers

Five “layers”!

- **Modularity**

Networking — A Stack of Protocol Layers

Five “layers”!

- **Modularity**
    - Each layer relies on services from layer below

Networking — A Stack of Protocol Layers

Five “layers”!

- **Modularity**
    - Each layer relies on services from layer below
    - Each layer exports services to layer above

Networking — A Stack of Protocol Layers

Five “layers”!

- **Modularity**
    - Each layer relies on services from layer below
    - Each layer exports services to layer above
- **Interfaces**

Networking — A Stack of Protocol Layers

Five “layers”!

- **Modularity**
    - Each layer relies on services from layer below
    - Each layer exports services to layer above
- **Interfaces**
    - Hide implementation details

Networking — A Stack of Protocol Layers

Five “layers”!

- **Modularity**
    - Each layer relies on services from layer below
    - Each layer exports services to layer above
- **Interfaces**
    - Hide implementation details
    - Layers can change without disturbing other layers

Networking — A Stack of Protocol Layers

Networking — A Stack of Protocol Layers

Five “layers”!

Networking — A Stack of Protocol Layers

Five “layers”!

Physical layer

Networking — A Stack of Protocol Layers

Five “layers”!

Physical layer

Transfer “signals”

Networking — A Stack of Protocol Layers

Five “layers”!

Link layer

Physical layer

Transfer “signals”

Networking — A Stack of Protocol Layers

Five “layers”!

Link layer	Postman
Physical layer	Transfer “signals”

Networking — A Stack of Protocol Layers

Five “layers”!

Network

Link layer

Physical layer

Postman

Transfer “signals”

Networking — A Stack of Protocol Layers

Five “layers”!

Network	Airplane/rail
Link layer	Postman
Physical layer	Transfer “signals”

Networking — A Stack of Protocol Layers

Five “layers”!

Transport	
Network	Airplane/rail
Link layer	Postman
Physical layer	Transfer “signals”

Networking — A Stack of Protocol Layers

Five “layers”!

Transport	Post office
Network	Airplane/rail
Link layer	Postman
Physical layer	Transfer “signals”

Networking — A Stack of Protocol Layers

Five “layers”!

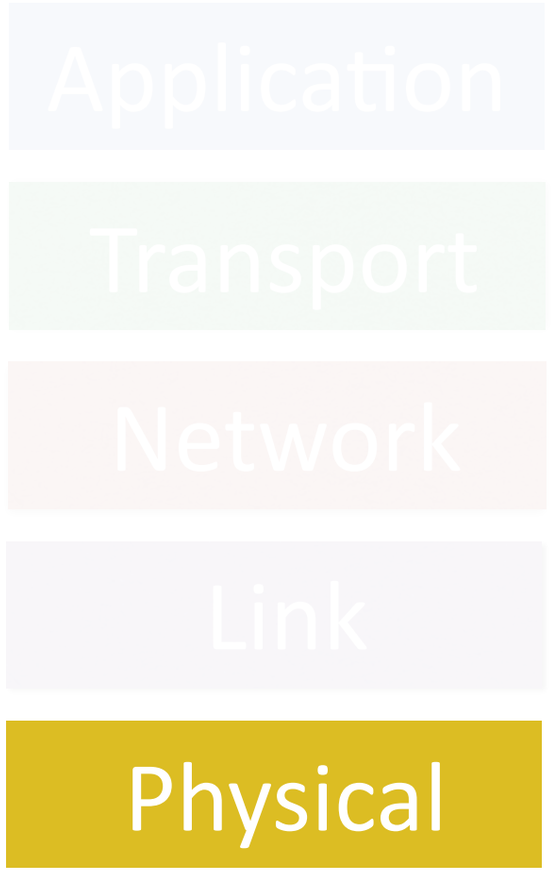
Application	
Transport	Post office
Network	Airplane/rail
Link layer	Postman
Physical layer	Transfer “signals”

Networking — A Stack of Protocol Layers

Five “layers”!

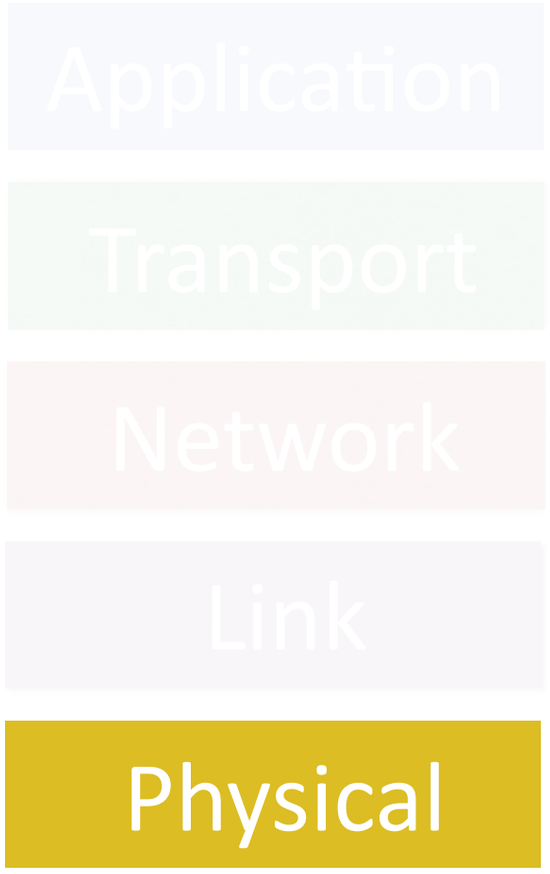
Application	You
Transport	Post office
Network	Airplane/rail
Link layer	Postman
Physical layer	Transfer “signals”

Networking — Physical layer



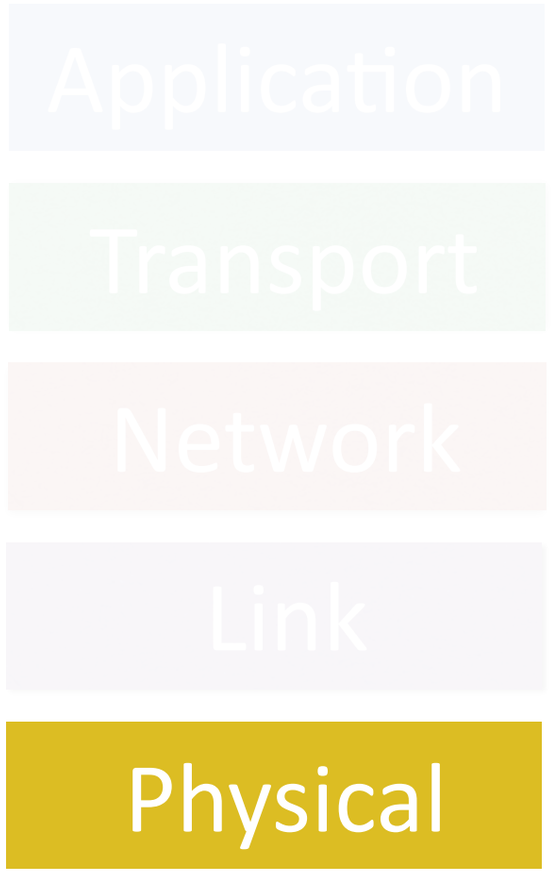
Networking — Physical layer

- **Transfer of bits**



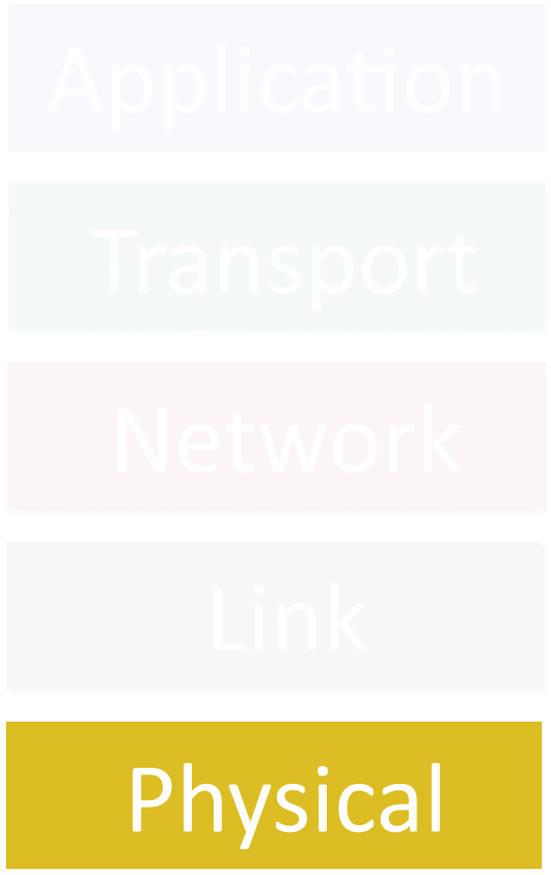
Networking — Physical layer

- **Transfer of bits**
  - 0s and 1s

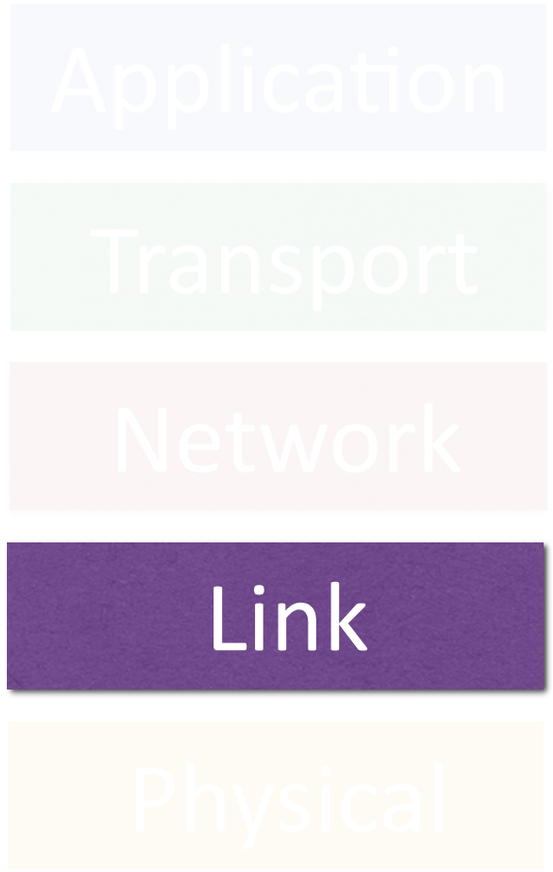


Networking — Physical layer

- **Transfer of bits**
  - 0s and 1s
  - Not concerned with protocols

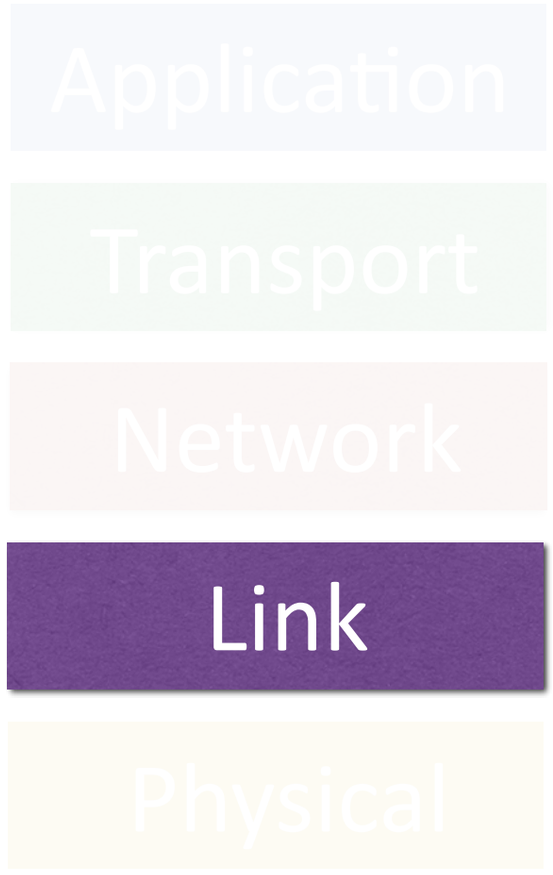


Networking — Link layer



Networking — Link layer

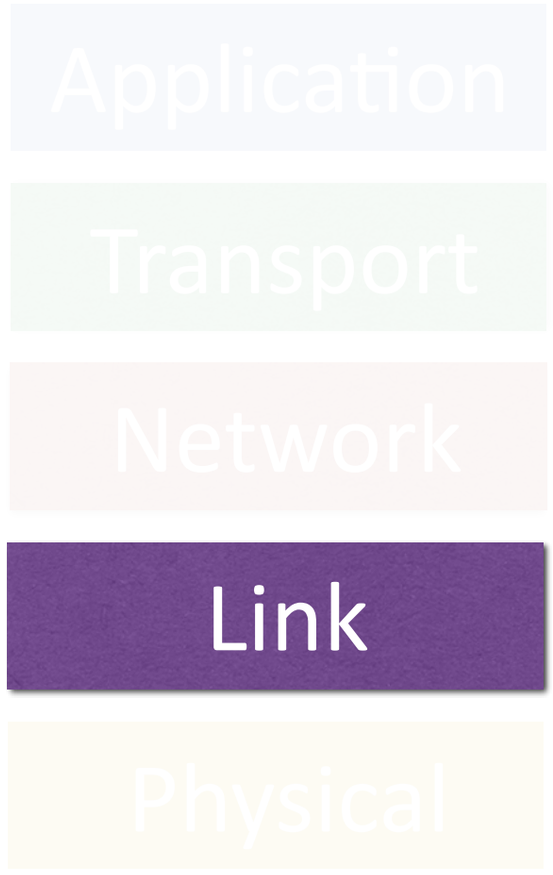
Link = Medium + Adapters



Networking — Link layer

Link = Medium + Adapters

- **Communication Medium**



Networking — Link layer

Link = Medium + Adapters

- **Communication Medium**

Networking — Link layer

Link = Medium + Adapters

- **Communication Medium**

- **Network Adapters (e.g., NIC — network interface card)**

Networking — Link layer

Link = Medium + Adapters

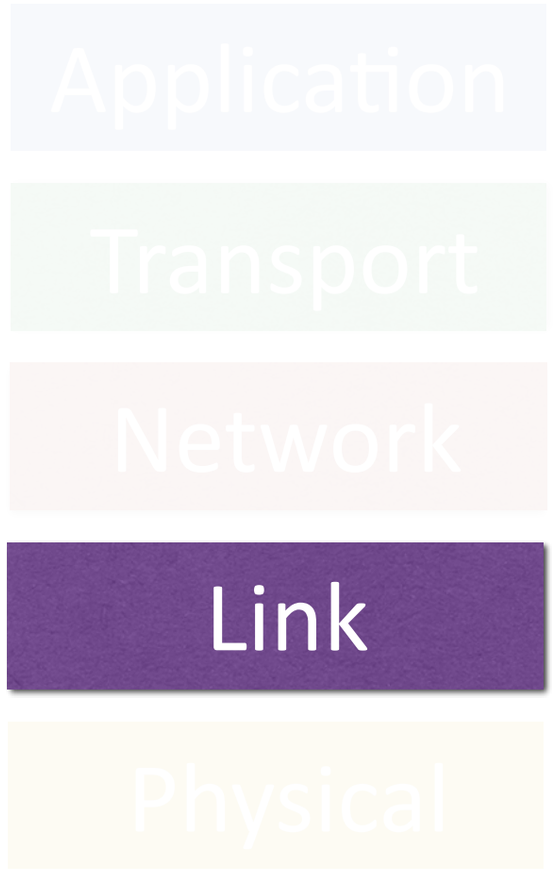
- **Communication Medium**




- **Network Adapters (e.g., NIC — network interface card)**

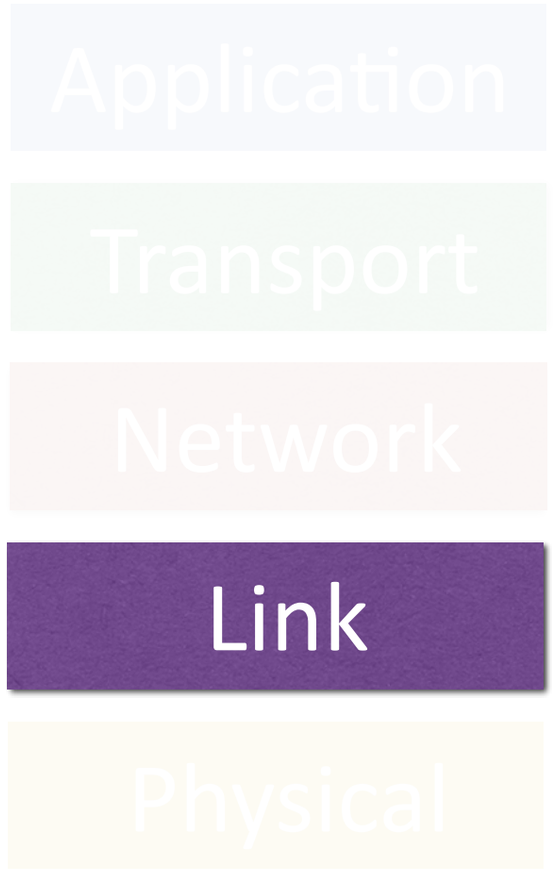


Networking — Link layer



Networking — Link layer

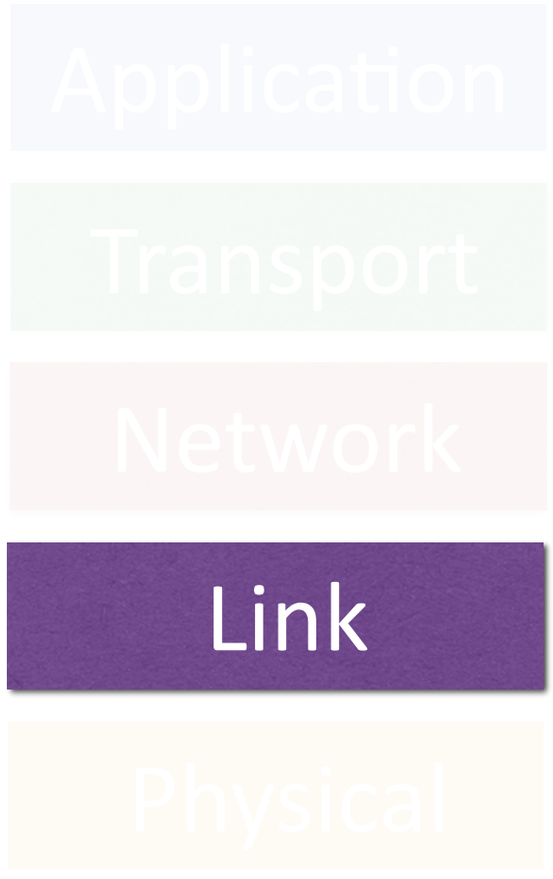
Broadcast links = Shared Medium



Networking — Link layer

Broadcast links = Shared Medium

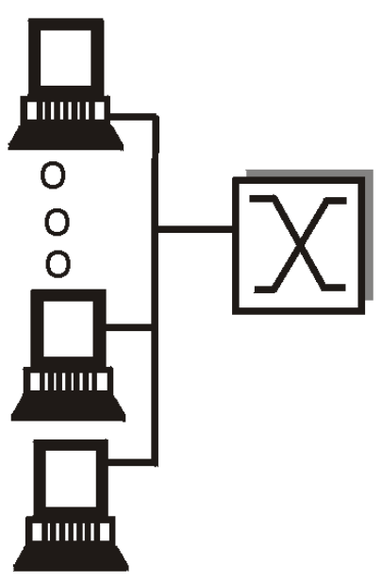
- Everyone listens to everybody



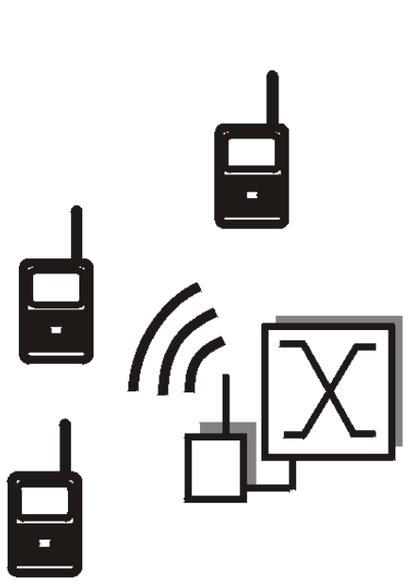
Networking — Link layer

Broadcast links = Shared Medium

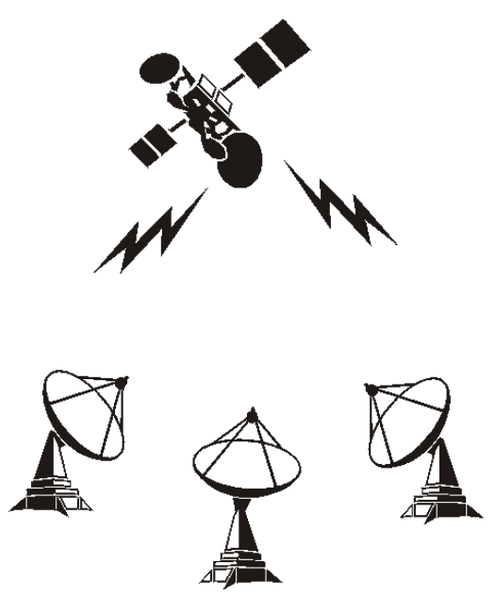
- Everyone listens to everybody



shared wire
(e.g. Ethernet)



shared wireless
(e.g. Wavelan)



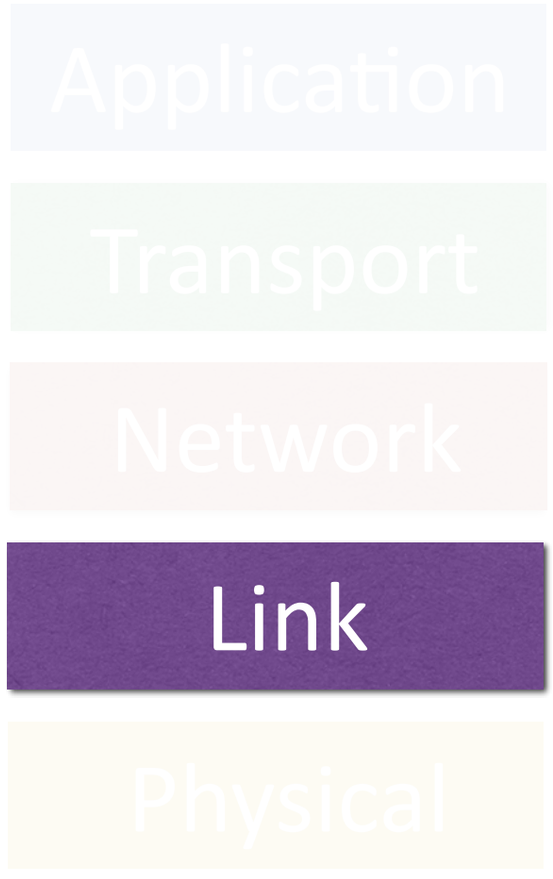
satellite

cocktail party

Networking — Link layer

Broadcast links = Shared Medium

- Everyone listens to everybody



Networking — Link layer

Broadcast links = Shared Medium

- Everyone listens to everybody

Networking — Link layer

Broadcast links = Shared Medium

- Everyone listens to everybody

source

Networking — Link layer

Broadcast links = Shared Medium

- Everyone listens to everybody

Networking — Link layer

Broadcast links = Shared Medium

- Everyone listens to everybody

Networking — Link layer

Broadcast links = Shared Medium

- Everyone listens to everybody

link-layer “protocol”

Networking — Link layer

Networking — Link layer

Five “services”!

Networking — Link layer

Five “services”!

- **Encoding data**
    - Represented as a collection of 0s and 1s

Networking — Link layer

Five “services”!

- **Encoding data**
    - Represented as a collection of 0s and 1s
- **Framing**
    - Put data packet into a frame; add receiver address

Networking — Link layer

Five “services”!

- **Encoding data**
    - Represented as a collection of 0s and 1s
- **Framing**
    - Put data packet into a frame; add receiver address
- **Error detection and correction**
    - Detect and (optionally) correct errors

Networking — Link layer

Five “services”!

- **Encoding data**
    - Represented as a collection of 0s and 1s
- **Framing**
    - Put data packet into a frame; add receiver address
- **Error detection and correction**
    - Detect and (optionally) correct errors
- **Flow control**
    - When to send/receive frames
    - Depends on the protocol

Networking — Link layer

Networking — Link layer

Addresses

Networking — Link layer

Addresses

- **Unique identifiers for sources and destinations**

Networking — Link layer

Addresses

- **Unique identifiers for sources and destinations**
    - “Hard-coded” in the adapter

Networking — Link layer

Addresses

- **Unique identifiers for sources and destinations**
    - “Hard-coded” in the adapter
    - MAC address (e.g., 00-15-C5-49-04-A9)

Networking — Link layer

Addresses

- **Unique identifiers for sources and destinations**
    - "Hard-coded" in the adapter
    - MAC address (e.g., 00-15-C5-49-04-A9)
    - Hierarchical allocation

Networking — Link layer

Addresses

- **Unique identifiers for sources and destinations**
  - "Hard-coded" in the adapter
  - MAC address (e.g., 00-15-C5-49-04-A9)
  - Hierarchical allocation
    - Blocks: assigned to vendors (e.g., Dell) from IEEE

Networking — Link layer

Addresses

- **Unique identifiers for sources and destinations**
  - "Hard-coded" in the adapter
  - MAC address (e.g., 00-15-C5-49-04-A9)
  - Hierarchical allocation
    - Blocks: assigned to vendors (e.g., Dell) from IEEE
    - Adapters: assigned by the vendor from its block

Networking — Link layer

Addresses

- **Unique identifiers for sources and destinations**
  - "Hard-coded" in the adapter
  - MAC address (e.g., 00-15-C5-49-04-A9)
  - Hierarchical allocation
    - Blocks: assigned to vendors (e.g., Dell) from IEEE
    - Adapters: assigned by the vendor from its block
- **What if I want to send to everybody?**

Networking — Link layer

Addresses

- **Unique identifiers for sources and destinations**
  - "Hard-coded" in the adapter
  - MAC address (e.g., 00-15-C5-49-04-A9)
  - Hierarchical allocation
    - Blocks: assigned to vendors (e.g., Dell) from IEEE
    - Adapters: assigned by the vendor from its block
- **What if I want to send to everybody?**
  - Special (broadcast) address: FF-FF-FF-FF-FF-FF

Networking — Link layer

Networking — Link layer

Sharing a medium

Networking — Link layer

Sharing a medium

- **Ever been to a party?**

Networking — Link layer

Sharing a medium

- **Ever been to a party?**
    - Tried to have an interesting discussion?

Networking — Link layer

Sharing a medium

- **Ever been to a party?**
    - Tried to have an interesting discussion?
- **Collisions**

Networking — Link layer

Sharing a medium

- **Ever been to a party?**
    - Tried to have an interesting discussion?
- **Collisions**

Link layer — Sending/receiving

Link layer — Sending/receiving

Lets try to come up with a protocol to avoid collisions!

Link layer — Sending/receiving

Lets try to come up with a protocol to avoid collisions!

- **Attempt 1: Time sharing**

Link layer — Sending/receiving

Lets try to come up with a protocol to avoid collisions!

- **Attempt 1: Time sharing**
    - Everybody gets a turn to speak

Link layer — Sending/receiving

Lets try to come up with a protocol to avoid collisions!

- **Attempt 1: Time sharing**
    - Everybody gets a turn to speak
- **Goods**

Link layer — Sending/receiving

Lets try to come up with a protocol to avoid collisions!

- **Attempt 1: Time sharing**
    - Everybody gets a turn to speak
- **Goods**
    - Never have a collision

Link layer — Sending/receiving

Lets try to come up with a protocol to avoid collisions!

- **Attempt 1: Time sharing**
    - Everybody gets a turn to speak
- **Goods**
    - Never have a collision
- **Problem**

Link layer — Sending/receiving

Lets try to come up with a protocol to avoid collisions!

- **Attempt 1: Time sharing**
    - Everybody gets a turn to speak
- **Goods**
    - Never have a collision
- **Problem**
    - Underutilization of resources

Link layer — Sending/receiving

Lets try to come up with a protocol to avoid collisions!

- **Attempt 1: Time sharing**
    - Everybody gets a turn to speak
- **Goods**
    - Never have a collision
- **Problem**
    - Underutilization of resources
        - During my turn, I may have nothing to speak

Link layer — Sending/receiving

Lets try to come up with a protocol to avoid collisions!

- **Attempt 1: Time sharing**
    - Everybody gets a turn to speak
- **Goods**
    - Never have a collision
- **Problem**
    - Underutilization of resources
        - During my turn, I may have nothing to speak
        - When I have something to speak, I wait for my turn

Link layer — Sending/receiving

Link layer — Sending/receiving

Lets try another protocol to avoid collisions

Link layer — Sending/receiving

Lets try another protocol to avoid collisions

- **Attempt 2: Frequency sharing**

Link layer — Sending/receiving

Lets try another protocol to avoid collisions

- **Attempt 2: Frequency sharing**
    - For wireless and optical mediums

Link layer — Sending/receiving

Lets try another protocol to avoid collisions

- **Attempt 2: Frequency sharing**
    - For wireless and optical mediums
    - Each source assigned a particular frequency; receivers tune

Link layer — Sending/receiving

Lets try another protocol to avoid collisions

- **Attempt 2: Frequency sharing**
    - For wireless and optical mediums
    - Each source assigned a particular frequency; receivers tune
    - E.g., Divide into groups; each group talks among themselves

Link layer — Sending/receiving

Lets try another protocol to avoid collisions

- **Attempt 2: Frequency sharing**
    - For wireless and optical mediums
    - Each source assigned a particular frequency; receivers tune
    - E.g., Divide into groups; each group talks among themselves
- **Problem**

Link layer — Sending/receiving

Lets try another protocol to avoid collisions

- **Attempt 2: Frequency sharing**
    - For wireless and optical mediums
    - Each source assigned a particular frequency; receivers tune
    - E.g., Divide into groups; each group talks among themselves
- **Problem**
    - Overheads …

Link layer — Sending/receiving

Lets try another protocol to avoid collisions

- **Attempt 2: Frequency sharing**
    - For wireless and optical mediums
    - Each source assigned a particular frequency; receivers tune
    - E.g., Divide into groups; each group talks among themselves

- **Problem**
    - Overheads …
    - What if I want to talk to only a few people in the group?

Link layer — Sending/receiving

Lets try another protocol to avoid collisions

- **Attempt 2: Frequency sharing**
    - For wireless and optical mediums
    - Each source assigned a particular frequency; receivers tune
    - E.g., Divide into groups; each group talks among themselves

- **Problem**
    - Overheads …
    - What if I want to talk to only a few people in the group?
    - What if I want to talk to people in different groups?

Link layer — Sending/receiving

Lets try another protocol to avoid collisions

- **Attempt 2: Frequency sharing**
    - For wireless and optical mediums
    - Each source assigned a particular frequency; receivers tune
    - E.g., Divide into groups; each group talks among themselves

- **Problem**
    - Overheads …
    - What if I want to talk to only a few people in the group?
    - What if I want to talk to people in different groups?
    - E.g., one person wants to announce something …

Link layer — Sending/receiving

Link layer — Sending/receiving

Attempt 3: Carrier sense, Collision detection, Random access

Link layer — Sending/receiving

Attempt 3: Carrier sense, Collision detection, Random access

- **Carrier Sense**

Link layer — Sending/receiving

Attempt 3: Carrier sense, Collision detection, Random access

- **Carrier Sense**
    - Listen before speaking

Link layer — Sending/receiving

Attempt 3: Carrier sense, Collision detection, Random access

- **Carrier Sense**
    - Listen before speaking
    - …. and don’t interrupt

Link layer — Sending/receiving

Attempt 3: Carrier sense, Collision detection, Random access

- **Carrier Sense**
    - Listen before speaking
    - …. and don’t interrupt
- **Collision detection**

Link layer — Sending/receiving

Attempt 3: Carrier sense, Collision detection, Random access

- **Carrier Sense**
    - Listen before speaking
    - …. and don’t interrupt
- **Collision detection**
    - Detect simultaneous speaking

Link layer — Sending/receiving

Attempt 3: Carrier sense, Collision detection, Random access

- **Carrier Sense**
    - Listen before speaking
    - …. and don't interrupt
- **Collision detection**
    - Detect simultaneous speaking
    - …. and shut up!

Link layer — Sending/receiving

Attempt 3: Carrier sense, Collision detection, Random access

- **Carrier Sense**
  - Listen before speaking
  - …. and don’t interrupt
- **Collision detection**
  - Detect simultaneous speaking
  - …. and shut up!
- **Random access**

Link layer — Sending/receiving

Attempt 3: Carrier sense, Collision detection, Random access

- **Carrier Sense**
    - Listen before speaking
    - …. and don't interrupt
- **Collision detection**
    - Detect simultaneous speaking
    - …. and shut up!
- **Random access**
    - Wait for a random period of time

Link layer — Sending/receiving

Attempt 3: Carrier sense, Collision detection, Random access

- **Carrier Sense**
  - Listen before speaking
  - …. and don't interrupt
- **Collision detection**
  - Detect simultaneous speaking
  - …. and shut up!
- **Random access**
  - Wait for a random period of time
  - …. before trying to talk again

Link layer — Sending/receiving

Link layer — Sending/receiving

Comparing the three approaches

Link layer — Sending/receiving

Comparing the three approaches

- **Time division**

Link layer — Sending/receiving

Comparing the three approaches

- **Time division**
    - No collisions

Link layer — Sending/receiving

Comparing the three approaches

- **Time division**
    - No collisions
    - Underutilization of resources!

Link layer — Sending/receiving

Comparing the three approaches

- **Time division**
    - No collisions
    - Underutilization of resources!
    - What if token is lost?

Link layer — Sending/receiving

Comparing the three approaches

- **Time division**
    - No collisions
    - Underutilization of resources!
    - What if token is lost?
- **Frequency division**

Link layer — Sending/receiving

Comparing the three approaches

- **Time division**
    - No collisions
    - Underutilization of resources!
    - What if token is lost?
- **Frequency division**
    - Overheads

Link layer — Sending/receiving

Comparing the three approaches

- **Time division**
    - No collisions
    - Underutilization of resources!
    - What if token is lost?
- **Frequency division**
    - Overheads
- **Random access**

Link layer — Sending/receiving

Comparing the three approaches

- **Time division**
    - No collisions
    - Underutilization of resources!
    - What if token is lost?
- **Frequency division**
    - Overheads
- **Random access**
    - Efficient at low load, inefficient at high load (collisions)

Ethernet — Sending/receiving at Link layer

Ethernet — Sending/receiving at Link layer

Ethernet uses CSMA/CD

Ethernet — Sending/receiving at Link layer

Ethernet uses CSMA/CD

- **Carrier Sense: continuously listen to the channel**

Ethernet — Sending/receiving at Link layer

Ethernet uses CSMA/CD

- **Carrier Sense: continuously listen to the channel**
    - If idle: start transmitting

Ethernet — Sending/receiving at Link layer

Ethernet uses CSMA/CD

- **Carrier Sense: continuously listen to the channel**
    - If idle: start transmitting
    - If busy: wait until idle

Ethernet — Sending/receiving at Link layer

Ethernet uses CSMA/CD

- **Carrier Sense: continuously listen to the channel**
    - If idle: start transmitting
    - If busy: wait until idle
- **Collision Detection: listen while transmitting**

Ethernet — Sending/receiving at Link layer

Ethernet uses CSMA/CD

- **Carrier Sense: continuously listen to the channel**
    - If idle: start transmitting
    - If busy: wait until idle
- **Collision Detection: listen while transmitting**
    - No collision: transmission complete

Ethernet — Sending/receiving at Link layer

Ethernet uses CSMA/CD

- **Carrier Sense: continuously listen to the channel**
    - If idle: start transmitting
    - If busy: wait until idle
- **Collision Detection: listen while transmitting**
    - No collision: transmission complete
    - Collision: abort transmission; send jam signal

Ethernet — Sending/receiving at Link layer

Ethernet uses CSMA/CD

- **Carrier Sense: continuously listen to the channel**
    - If idle: start transmitting
    - If busy: wait until idle
- **Collision Detection: listen while transmitting**
    - No collision: transmission complete
    - Collision: abort transmission; send jam signal
- **Random access: exponential back off**

Ethernet — Sending/receiving at Link layer

Ethernet uses CSMA/CD

- **Carrier Sense: continuously listen to the channel**
    - If idle: start transmitting
    - If busy: wait until idle
- **Collision Detection: listen while transmitting**
    - No collision: transmission complete
    - Collision: abort transmission; send jam signal
- **Random access: exponential back off**
    - After collision, transmit after “waiting time”

Ethernet — Sending/receiving at Link layer

Ethernet uses CSMA/CD

- **Carrier Sense: continuously listen to the channel**
    - If idle: start transmitting
    - If busy: wait until idle
- **Collision Detection: listen while transmitting**
    - No collision: transmission complete
    - Collision: abort transmission; send jam signal
- **Random access: exponential back off**
    - After collision, transmit after “waiting time”
    - After k collisions, choose “waiting time” from {0, …, 2^k-1)

Ethernet — Sending/receiving at Link layer

Ethernet uses CSMA/CD

- **Carrier Sense: continuously listen to the channel**
    - If idle: start transmitting
    - If busy: wait until idle
- **Collision Detection: listen while transmitting**
    - No collision: transmission complete
    - Collision: abort transmission; send jam signal
- **Random access: exponential back off**
    - After collision, transmit after “waiting time”
    - After k collisions, choose “waiting time” from {0, …, 2^k-1)
    - (Exponentially increasing waiting times)

Networking — Link layer (Ethernet)

Networking — Link layer (Ethernet)

Interesting Properties

Networking — Link layer (Ethernet)

Interesting Properties

- **Distributed**

Networking — Link layer (Ethernet)

Interesting Properties

- **Distributed**
    - **No** Central arbitrer

Networking — Link layer (Ethernet)

Interesting Properties

- **Distributed**
    - **No** Central arbitrer
    - Why is that good?

Networking — Link layer (Ethernet)

Interesting Properties

- **Distributed**
    - **No** Central arbitrer
    - Why is that good?
- **Inexpensive**

Networking — Link layer (Ethernet)

Interesting Properties

- **Distributed**
    - **No** Central arbitrer
    - Why is that good?
- **Inexpensive**
    - No state in the network

Networking — Link layer (Ethernet)

Interesting Properties

- **Distributed**
    - **No** Central arbitrer
    - Why is that good?
- **Inexpensive**
    - No state in the network
    - Cheap physical links

Networking — Link layer (Ethernet)

Networking — Link layer (Ethernet)

Connection-less, unreliable service

Networking — Link layer (Ethernet)

Connection-less, unreliable service

- **Connection less**

Networking — Link layer (Ethernet)

Connection-less, unreliable service

- **Connection less**
    - E.g., I am going to talk to you without getting permission first

Networking — Link layer (Ethernet)

Connection-less, unreliable service

- **Connection less**
    - E.g., I am going to talk to you without getting permission first
    - Networking terminology: No “handshaking”

Networking — Link layer (Ethernet)

Connection-less, unreliable service

- **Connection less**
    - E.g., I am going to talk to you without getting permission first
    - Networking terminology: No “handshaking”
- **Unreliable**

Networking — Link layer (Ethernet)

Connection-less, unreliable service

- **Connection less**
    - E.g., I am going to talk to you without getting permission first
    - Networking terminology: No “handshaking”
- **Unreliable**
    - Destination adapter does not acknowledge

Networking — Link layer (Ethernet)

Connection-less, unreliable service

- **Connection less**
    - E.g., I am going to talk to you without getting permission first
    - Networking terminology: No “handshaking”
- **Unreliable**
    - Destination adapter does not acknowledge
        - Did you listen to what I said?

Networking — Link layer (Ethernet)

Connection-less, unreliable service

- **Connection less**
    - E.g., I am going to talk to you without getting permission first
    - Networking terminology: No “handshaking”
- **Unreliable**
    - Destination adapter does not acknowledge
        - Did you listen to what I said?
    - Adversarial behavior could bring the connections down

Networking — Link layer (Ethernet)

Connection-less, unreliable service

- **Connection less**
    - E.g., I am going to talk to you without getting permission first
    - Networking terminology: No “handshaking”
- **Unreliable**
    - Destination adapter does not acknowledge
        - Did you listen to what I said?
    - Adversarial behavior could bring the connections down
        - I am going to ignore the protocol

Networking — Link layer (Ethernet)

Connection-less, unreliable service

- **Connection less**
    - E.g., I am going to talk to you without getting permission first
    - Networking terminology: No “handshaking”
- **Unreliable**
    - Destination adapter does not acknowledge
        - Did you listen to what I said?
    - Adversarial behavior could bring the connections down
        - I am going to ignore the protocol
    - Untrusted data access

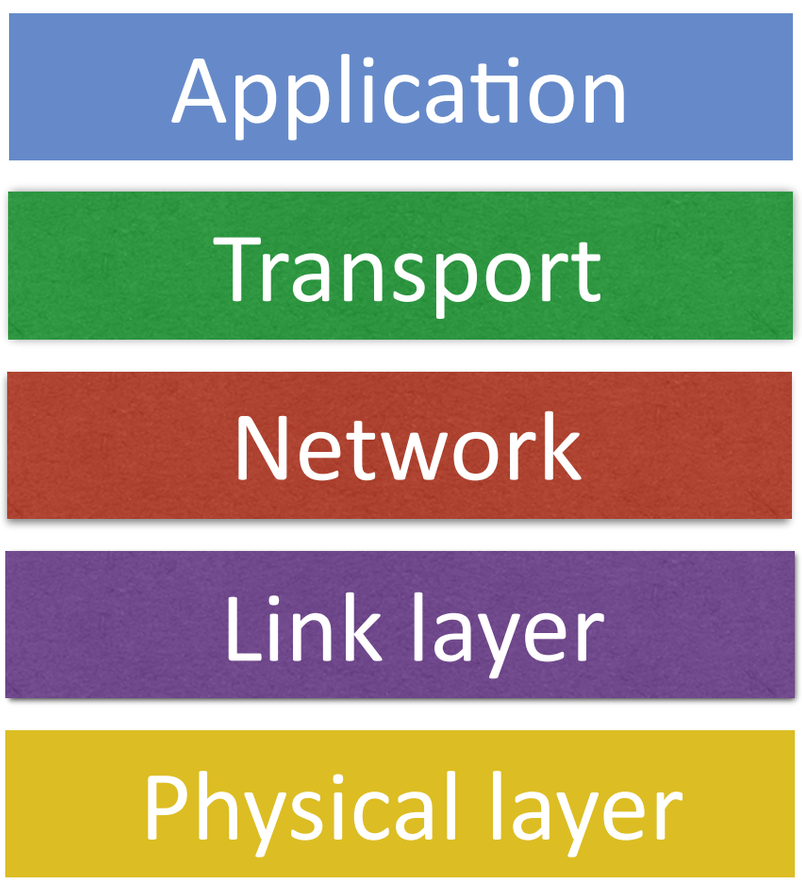
Networking — Link layer (Ethernet)

Connection-less, unreliable service

- **Connection less**
    - E.g., I am going to talk to you without getting permission first
    - Networking terminology: No “handshaking”
- **Unreliable**
    - Destination adapter does not acknowledge
        - Did you listen to what I said?
    - Adversarial behavior could bring the connections down
        - I am going to ignore the protocol
    - Untrusted data access
        - I want to listen to what others are talking

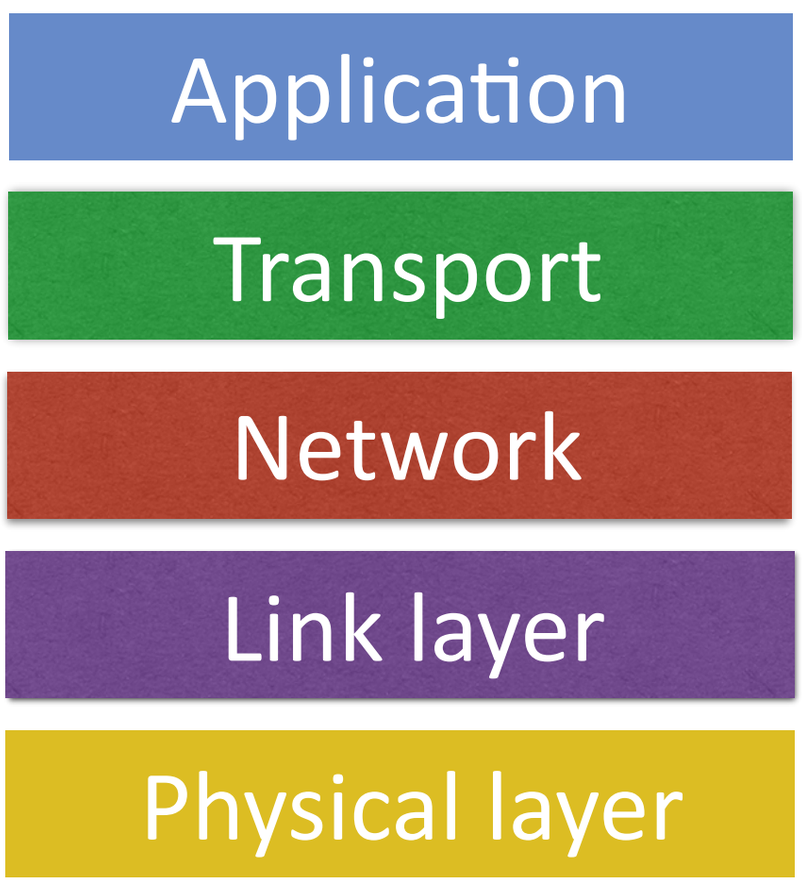
Networking — A Stack of Protocol Layers

Five “layers”!



Networking — A Stack of Protocol Layers

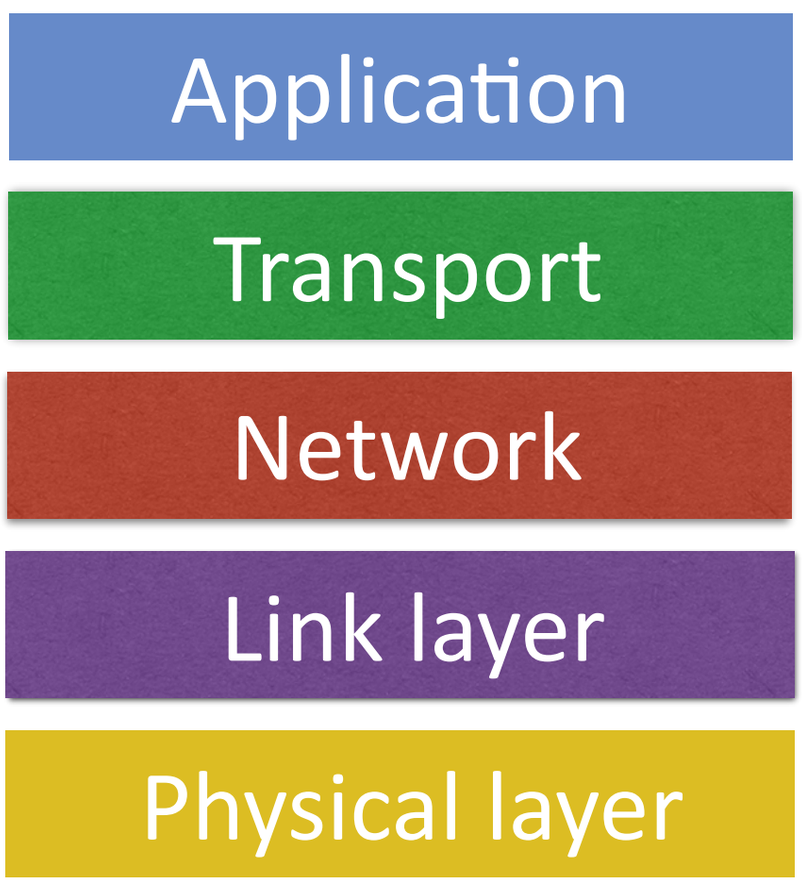
Five “layers”!



Deliver **signals**

Networking — A Stack of Protocol Layers

Five “layers”!

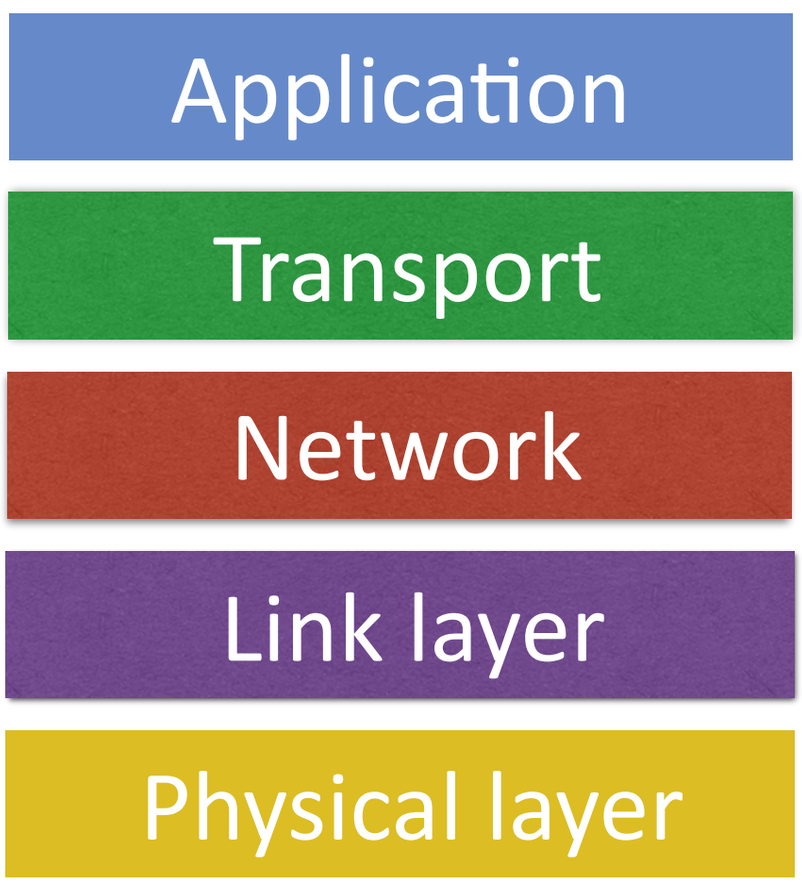


Deliver **locally**

Deliver **signals**

Networking — A Stack of Protocol Layers

Five “layers”!



Deliver **globally**

Deliver **locally**

Deliver **signals**

Networking — A Stack of Protocol Layers

Five “layers”!

Application	
Transport	Deliver **(un)reliably**
Network	Deliver **globally**
Link layer	Deliver **locally**
Physical layer	Deliver **signals**

Networking — A Stack of Protocol Layers

Five “layers”!

Application	**Deliver**
Transport	Deliver **(un)reliably**
Network	Deliver **globally**
Link layer	Deliver **locally**
Physical layer	Deliver **signals**

Networking — A Stack of Protocol Layers

Five “layers”!

Application	Deliver
Transport	Deliver **(un)reliably**
Network	Deliver **globally**
Link layer	Deliver **locally**
Physical layer	Deliver **signals**

Networking — Network layer

Networking — Network layer

Three concepts

Networking — Network layer

Three concepts

- **Naming**

Networking — Network layer

Three concepts

- **Naming**
    - A way to identify the source/destination

Networking — Network layer

Three concepts

- **Naming**
    - A way to identify the source/destination
    - E.g., house address

Networking — Network layer

Three concepts

- **Naming**
    - A way to identify the source/destination
    - E.g., house address
- **Routing**

Networking — Network layer

Three concepts

- **Naming**
    - A way to identify the source/destination
    - E.g., house address
- **Routing**
    - Finding “how to” move towards the destination

Networking — Network layer

Three concepts

- **Naming**
    - A way to identify the source/destination
    - E.g., house address
- **Routing**
    - Finding “how to” move towards the destination
    - E.g., which airplane should the stuff go on

Networking — Network layer

Three concepts

- **Naming**
    - A way to identify the source/destination
    - E.g., house address
- **Routing**
    - Finding “how to” move towards the destination
    - E.g., which airplane should the stuff go on
- **Forwarding**

Networking — Network layer

Three concepts

- **Naming**
    - A way to identify the source/destination
    - E.g., house address
- **Routing**
    - Finding “how to” move towards the destination
    - E.g., which airplane should the stuff go on
- **Forwarding**
    - Actually “moving” towards the destination

Networking — Network layer

Three concepts

- **Naming**
    - A way to identify the source/destination
    - E.g., house address
- **Routing**
    - Finding “how to” move towards the destination
    - E.g., which airplane should the stuff go on
- **Forwarding**
    - Actually “moving” towards the destination
    - E.g., Using airplane/truck/rail

Networking — Network layer

Networking — Network layer

Naming

Networking — Network layer

Naming

- **Give every computer a unique name**

Networking — Network layer

Naming

- **Give every computer a unique name**
    - Challenges?

Networking — Network layer

Naming

- **Give every computer a unique name**
    - Challenges?
    - Scalability — why?

Networking — Network layer

Naming

- **Give every computer a unique name**
    - Challenges?
    - Scalability — why?
    - Assignment — why?

Networking — Network layer

Networking — Network layer

Naming

Networking — Network layer

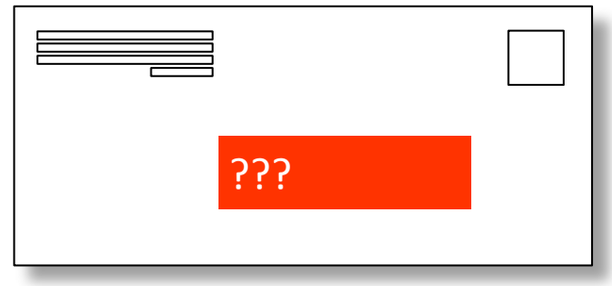
Naming

- **Hierarchical addressing**

Networking — Network layer

Naming

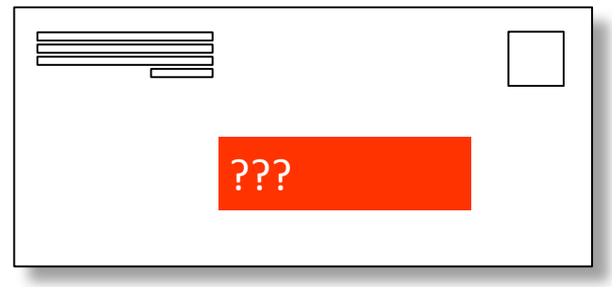
- **Hierarchical addressing**



Networking — Network layer

Naming

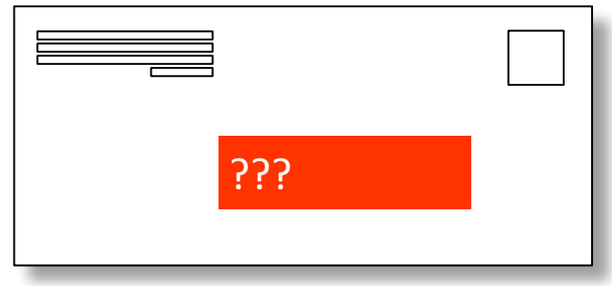
- **Hierarchical addressing**
    - E.g., addresses for houses



Networking — Network layer

Naming

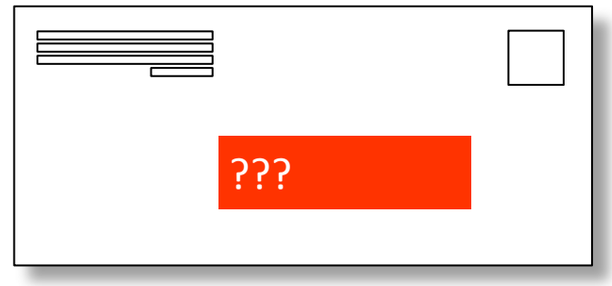
- **Hierarchical addressing**
    - E.g., addresses for houses
    - **Country:** USA



Networking — Network layer

Naming

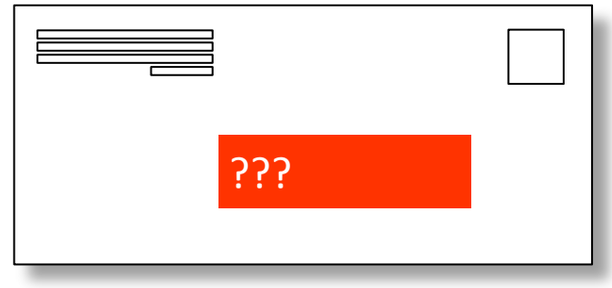
- **Hierarchical addressing**
    - E.g., addresses for houses
    - **Country:** USA
    - **City, State:** Ithaca, NY



Networking — Network layer

Naming

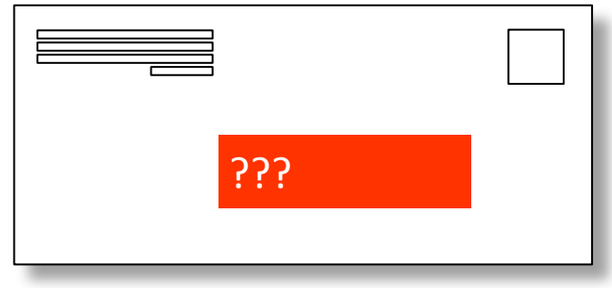
- **Hierarchical addressing**
    - E.g., addresses for houses
    - **Country:** USA
    - **City, State:** Ithaca, NY
    - **Number, Street**: 306 State St.



Networking — Network layer

Naming

- **Hierarchical addressing**
    - E.g., addresses for houses
    - **Country:** USA
    - **City, State:** Ithaca, NY
    - **Number, Street**: 306 State St.
    - **Name:** Rachit Agarwal



Networking — Network layer

Networking — Network layer

Hierarchical addressing

Networking — Network layer

Hierarchical addressing

Country	City, State	Street, Number	Occupant

Networking — Network layer

Hierarchical addressing

Country	City, State	Street, Number	Occupant
(8 bits)	(8 bits)	(8 bits)	(8 bits)

Networking — Network layer

Hierarchical addressing

Country	City, State	Street, Number	Occupant
(8 bits)	(8 bits)	(8 bits)	(8 bits)
10000000	0-1010100	10001011	00000-101

Networking — Network layer

Hierarchical addressing

Country	City, State	Street, Number	Occupant
(8 bits)	(8 bits)	(8 bits)	(8 bits)
10000000	0-1010100	10001011	00000-101
128	84	139	5

Networking — Network layer

Hierarchical addressing

Country	City, State	Street, Number	Occupant
(8 bits)	(8 bits)	(8 bits)	(8 bits)
10000000	0-1010100	10001011	00000-101
128	84	139	5

Network

Networking — Network layer

Hierarchical addressing

Country	City, State	Street, Number	Occupant
(8 bits)	(8 bits)	(8 bits)	(8 bits)
10000000	0-1010100	10001011	00000-101
128	84	139	5

Networking — Network layer

Hierarchical addressing

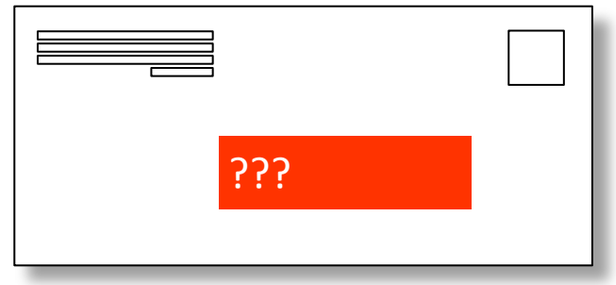
Country	City, State	Street, Number	Occupant
(8 bits)	(8 bits)	(8 bits)	(8 bits)
10000000	0-1010100	10001011	00000-101
128	84	139	5

IP address: 128.84.139.5

Networking — Network layer

Networking — Network layer

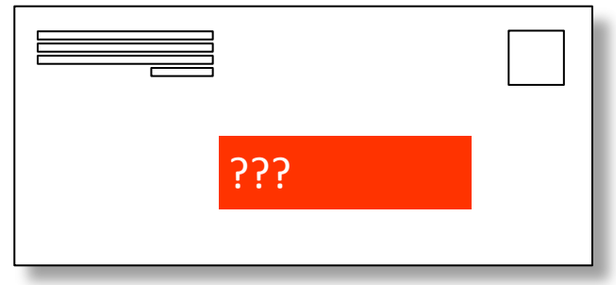
Hierarchical addressing



Networking — Network layer

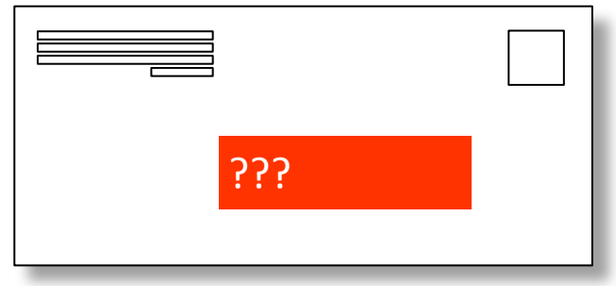
Hierarchical addressing

- **Why is it more scalable?**



Networking — Network layer

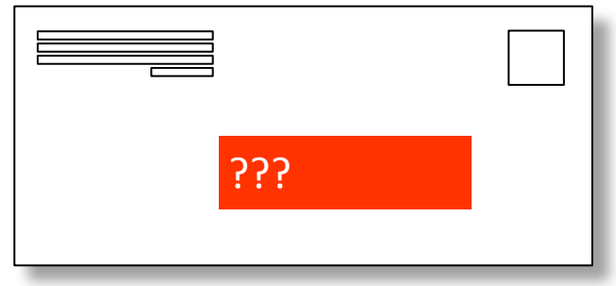
Hierarchical addressing



- **Why is it more scalable?**

Networking — Network layer

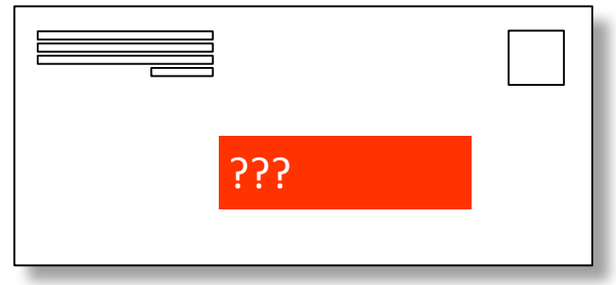
Hierarchical addressing



- **Why is it more scalable?**
    - Need to keep track of next step only!

Networking — Network layer

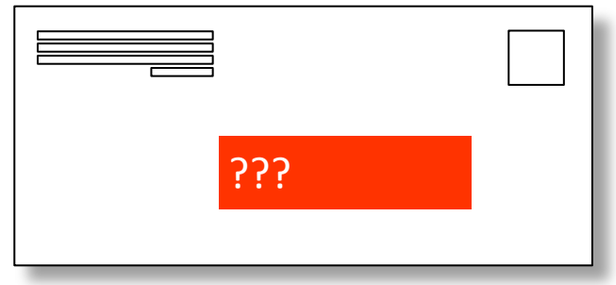
Hierarchical addressing



- **Why is it more scalable?**
    - Need to keep track of next step only!
    - **Flight to:** USA

Networking — Network layer

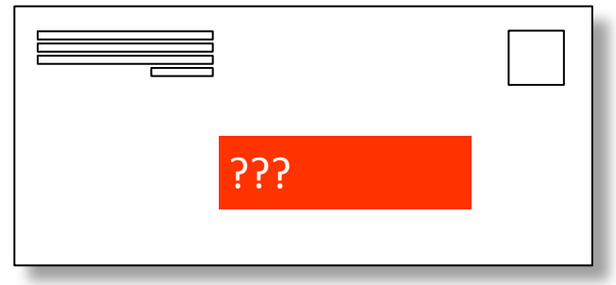
Hierarchical addressing



- **Why is it more scalable?**
    - Need to keep track of next step only!
    - **Flight to:** USA
    - **Truck to:** Ithaca, NY

Networking — Network layer

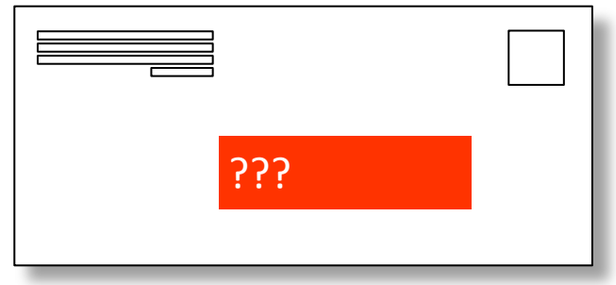
Hierarchical addressing



- **Why is it more scalable?**
    - Need to keep track of next step only!
    - **Flight to:** USA
    - **Truck to:** Ithaca, NY
    - **Direction to**: 306 State St.

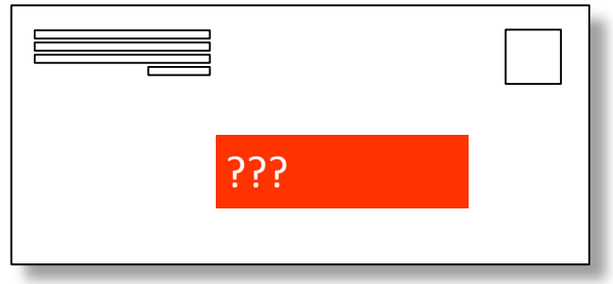
Networking — Network layer

Hierarchical addressing



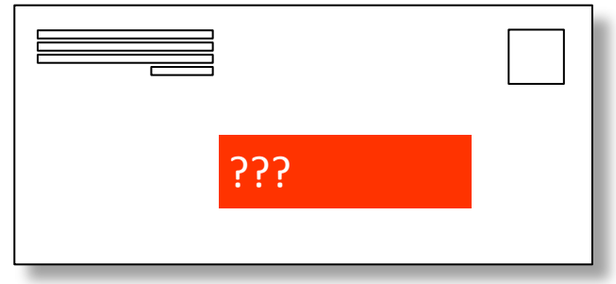
- **Why is it more scalable?**
    - Need to keep track of next step only!
    - **Flight to:** USA
    - **Truck to:** Ithaca, NY
    - **Direction to**: 306 State St.
    - **Mailbox:** Rachit Agarwal

Networking — Network layer



Networking — Network layer

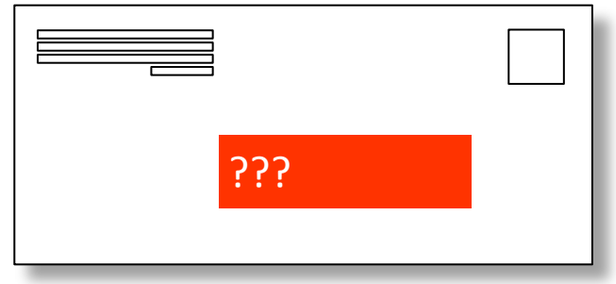
Hierarchical addressing



Networking — Network layer

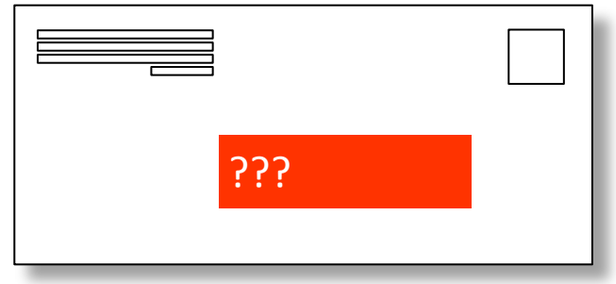
Hierarchical addressing

- **Why is it easier to assign?**



Networking — Network layer

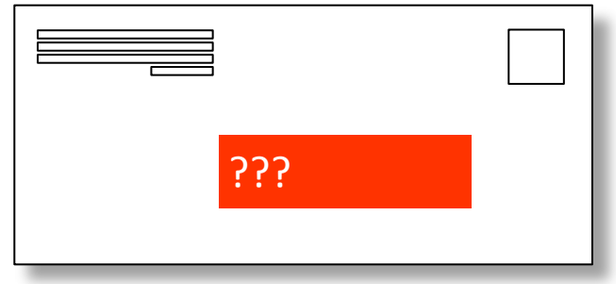
Hierarchical addressing



- **Why is it easier to assign?**
    - Just assign a new machine a “local” address!

Networking — Network layer

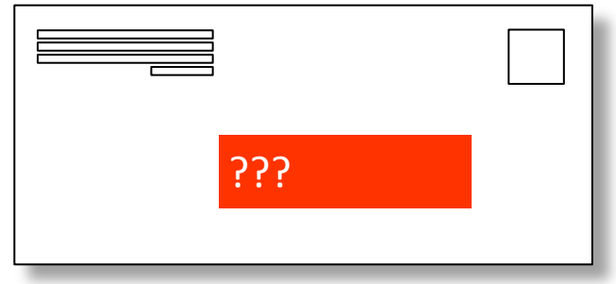
Hierarchical addressing



- **Why is it easier to assign?**
  - Just assign a new machine a “local” address!
  - **E.g.,** adding a new machine to Cornell network

Networking — Network layer

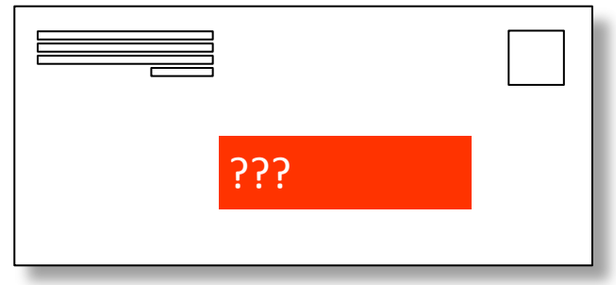
Hierarchical addressing



- **Why is it easier to assign?**
    - Just assign a new machine a “local” address!
    - **E.g.,** adding a new machine to Cornell network
    - **If last local address:** 128.84.139.5

Networking — Network layer

Hierarchical addressing



- **Why is it easier to assign?**
    - Just assign a new machine a “local” address!
    - **E.g.,** adding a new machine to Cornell network
    - **If last local address:** 128.84.139.5
    - **New machine gets**: 128.84.139.6

Networking — Network layer

Three concepts

- **Naming**
  - A way to identify the source/destination
  - E.g., house address
- **Routing**
  - Finding “how to” move towards the destination
  - E.g., which airplane should the stuff go on
- **Forwarding**
  - Actually “moving” towards the destination
  - E.g., Using airplane/truck/rail

Network layer — Forwarding

Network layer — Forwarding

Lets come up with an approach? Generalize Ethernet ideas?

Network layer — Forwarding

Network layer — Forwarding

Attempt 1: Broadcast

Network layer — Forwarding

Attempt 1: Broadcast

- **Send to everybody**

Network layer — Forwarding

Attempt 1: Broadcast

- **Send to everybody**
- **Goods**

Network layer — Forwarding

Attempt 1: Broadcast

- **Send to everybody**
- **Goods**
    - Oh, well, simplicity

Network layer — Forwarding

Attempt 1: Broadcast

- **Send to everybody**
- **Goods**
    - Oh, well, simplicity
- **Not-so-goods**

Network layer — Forwarding

Attempt 1: Broadcast

- **Send to everybody**
- **Goods**
    - Oh, well, simplicity
- **Not-so-goods**
    - Oh, well, everything else

Network layer — Forwarding

Attempt 1: Broadcast

- **Send to everybody**
- **Goods**
    - Oh, well, simplicity
- **Not-so-goods**
    - Oh, well, everything else
    - Bandwidth overheads

Network layer — Forwarding

Network layer — Forwarding

Attempt 2: Time division Multiplexing

Network layer — Forwarding

Attempt 2: Time division Multiplexing

- **Each source-destination pair assigned a time slot**

Network layer — Forwarding

Attempt 2: Time division Multiplexing

- **Each source-destination pair assigned a time slot**
    - Can send data only during that slot

Network layer — Forwarding

Attempt 2: Time division Multiplexing

- **Each source-destination pair assigned a time slot**
    - Can send data only during that slot
- **Goods**

Network layer — Forwarding

Attempt 2: Time division Multiplexing

- **Each source-destination pair assigned a time slot**
    - Can send data only during that slot
- **Goods**
    - No collisions

Network layer — Forwarding

Attempt 2: Time division Multiplexing

- **Each source-destination pair assigned a time slot**
    - Can send data only during that slot
- **Goods**
    - No collisions
- **Not-so-goods**

Network layer — Forwarding

Attempt 2: Time division Multiplexing

- **Each source-destination pair assigned a time slot**
    - Can send data only during that slot
- **Goods**
    - No collisions
- **Not-so-goods**
    - Underutilization of resources

Network layer — Forwarding

Network layer — Forwarding

Attempt 3: Frequency division Multiplexing

Network layer — Forwarding

Attempt 3: Frequency division Multiplexing

- **Each source-destination pair assigned a subset of resources**

Network layer — Forwarding

Attempt 3: Frequency division Multiplexing

- **Each source-destination pair assigned a subset of resources**
    - Can use only “assigned” resources (e.g., bandwidth)

Network layer — Forwarding

Attempt 3: Frequency division Multiplexing

- **Each source-destination pair assigned a subset of resources**
    - Can use only “assigned” resources (e.g., bandwidth)
- **Goods**

Network layer — Forwarding

Attempt 3: Frequency division Multiplexing

- **Each source-destination pair assigned a subset of resources**
    - Can use only “assigned” resources (e.g., bandwidth)
- **Goods**
    - Predictable performance

Network layer — Forwarding

Attempt 3: Frequency division Multiplexing

- **Each source-destination pair assigned a subset of resources**
    - Can use only “assigned” resources (e.g., bandwidth)
- **Goods**
    - Predictable performance
- **Not-so-goods**

Network layer — Forwarding

Attempt 3: Frequency division Multiplexing

- **Each source-destination pair assigned a subset of resources**
    - Can use only “assigned” resources (e.g., bandwidth)
- **Goods**
    - Predictable performance
- **Not-so-goods**
    - Underutilization of resources

Network layer — Forwarding

Network layer — Forwarding

Attempt 2 and 3: Circuit Switching

Network layer — Forwarding

Attempt 2 and 3: Circuit Switching

- **Source establishes connection**

Network layer — Forwarding

Attempt 2 and 3: Circuit Switching

- **Source establishes connection**
    - Resources along the path are reserved

Network layer — Forwarding

Attempt 2 and 3: Circuit Switching

- **Source establishes connection**
    - Resources along the path are reserved
- **Source sends data**

Network layer — Forwarding

Attempt 2 and 3: Circuit Switching

- **Source establishes connection**
    - Resources along the path are reserved
- **Source sends data**
    - Transmit data using the reserved resources

Network layer — Forwarding

Attempt 2 and 3: Circuit Switching

- **Source establishes connection**
    - Resources along the path are reserved
- **Source sends data**
    - Transmit data using the reserved resources
- **Source tears down connection**

Network layer — Forwarding

Attempt 2 and 3: Circuit Switching

- **Source establishes connection**
    - Resources along the path are reserved
- **Source sends data**
    - Transmit data using the reserved resources
- **Source tears down connection**
    - Free resources for others to use

Network layer — Forwarding

Network layer — Forwarding

Circuit Switching

Network layer — Forwarding

Circuit Switching

- **Goods:**

Network layer — Forwarding

Circuit Switching

- **Goods:**
    - Predictable performance

Network layer — Forwarding

Circuit Switching

- **Goods:**
    - Predictable performance
    - Reliable delivery

Network layer — Forwarding

Circuit Switching

- **Goods:**
    - Predictable performance
    - Reliable delivery
    - Simple forwarding mechanism

Network layer — Forwarding

Circuit Switching

- **Goods:**
    - Predictable performance
    - Reliable delivery
    - Simple forwarding mechanism
- **Not-so-goods**

Network layer — Forwarding

Circuit Switching

- **Goods:**
    - Predictable performance
    - Reliable delivery
    - Simple forwarding mechanism
- **Not-so-goods**
    - Resource underutilization

Network layer — Forwarding

Circuit Switching

- **Goods:**
    - Predictable performance
    - Reliable delivery
    - Simple forwarding mechanism
- **Not-so-goods**
    - Resource underutilization
    - Blocked connections

Network layer — Forwarding

Circuit Switching

- **Goods:**
    - Predictable performance
    - Reliable delivery
    - Simple forwarding mechanism
- **Not-so-goods**
    - Resource underutilization
    - Blocked connections
    - Connection set up overheads

Network layer — Forwarding

Circuit Switching

- **Goods:**
    - Predictable performance
    - Reliable delivery
    - Simple forwarding mechanism
- **Not-so-goods**
    - Resource underutilization
    - Blocked connections
    - Connection set up overheads
    - Per-connection state in switches (scalability problem)

Network layer — Forwarding

Network layer — Forwarding

Attempt 4: Packet Switching

Network layer — Forwarding

Attempt 4: Packet Switching

- **Divide the message into packets**

Network layer — Forwarding

Attempt 4: Packet Switching

- **Divide the message into packets**
- **Put destination address in the header of each packet**

Network layer — Forwarding

Attempt 4: Packet Switching

- **Divide the message into packets**
- **Put destination address in the header of each packet**
    - Just like shipping stuff

Network layer — Forwarding

Attempt 4: Packet Switching

- **Divide the message into packets**
- **Put destination address in the header of each packet**
    - Just like shipping stuff
- **Each device stores a "look-up table"**

Network layer — Forwarding

Attempt 4: Packet Switching

- **Divide the message into packets**
- **Put destination address in the header of each packet**
    - Just like shipping stuff
- **Each device stores a "look-up table"**
    - Whats the next hop towards the destination?

Network layer — Forwarding

Attempt 4: Packet Switching

- **Divide the message into packets**
- **Put destination address in the header of each packet**
    - Just like shipping stuff
- **Each device stores a "look-up table"**
    - Whats the next hop towards the destination?
- **Destination receives the packet(s)**

Network layer — Forwarding

Attempt 4: Packet Switching

- **Divide the message into packets**
- **Put destination address in the header of each packet**
    - Just like shipping stuff
- **Each device stores a "look-up table"**
    - Whats the next hop towards the destination?
- **Destination receives the packet(s)**
    - And reconstructs the message

Network layer — Forwarding

Network layer — Forwarding

Packet Switched forwarding

Network layer — Forwarding

Packet Switched forwarding

- **Hop-by-hop forwarding**

Network layer — Forwarding

Packet Switched forwarding

- **Hop-by-hop forwarding**
- **Each router has a “look-up table” (forwarding information base)**

Network layer — Forwarding

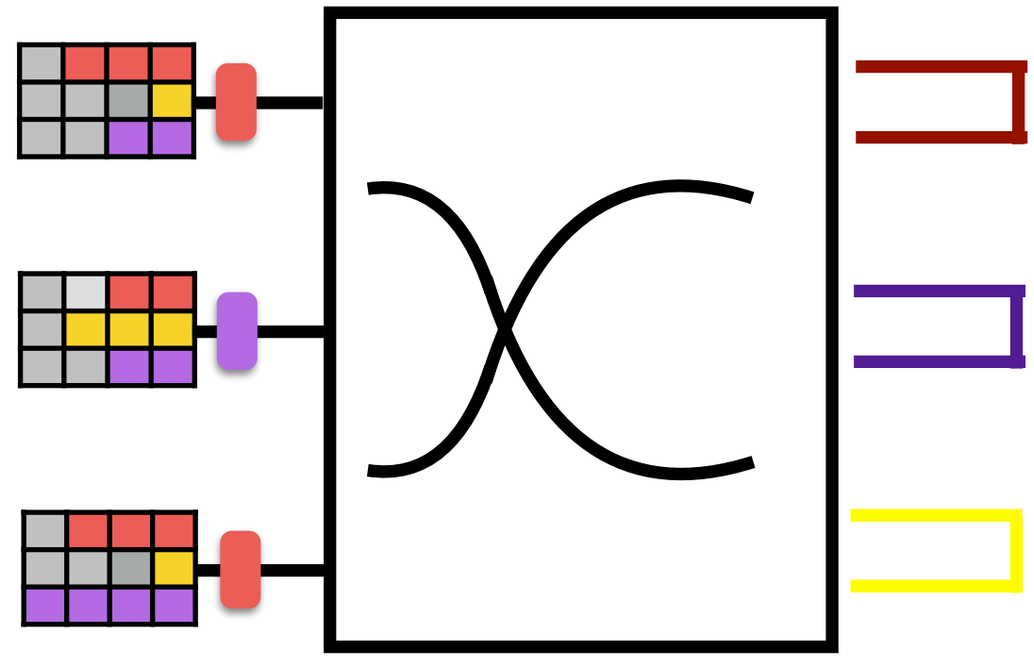
Packet Switched forwarding

- **Hop-by-hop forwarding**
- **Each router has a “look-up table” (forwarding information base)**
    - What should be stored in this table?

Network layer — Forwarding

Packet Switched forwarding

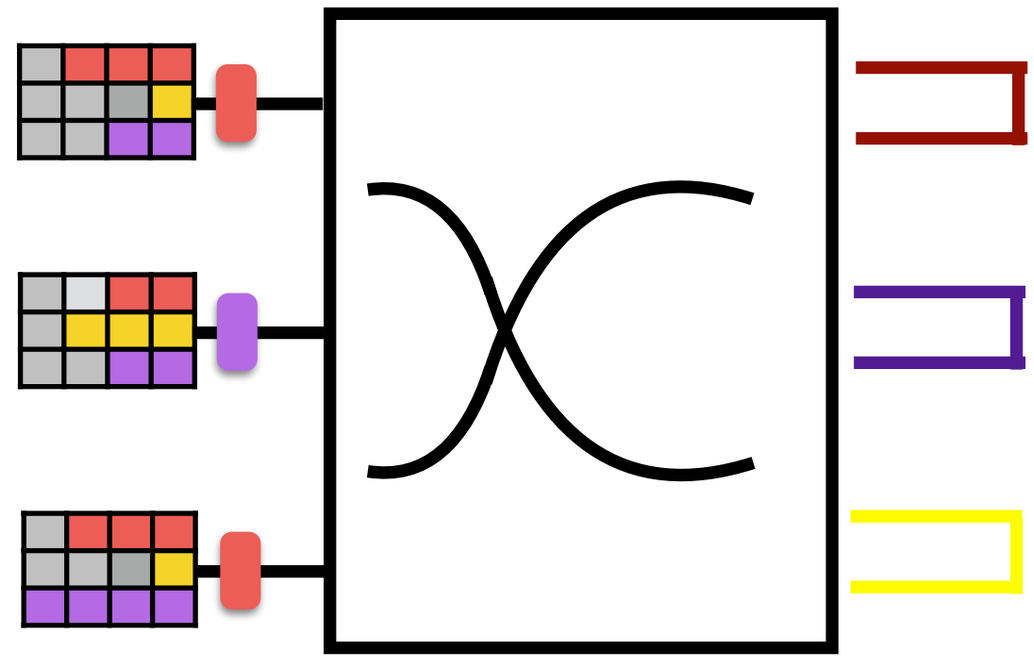
- **Hop-by-hop forwarding**
- **Each router has a “look-up table” (forwarding information base)**
    - What should be stored in this table?



Network layer — Forwarding

Packet Switched forwarding

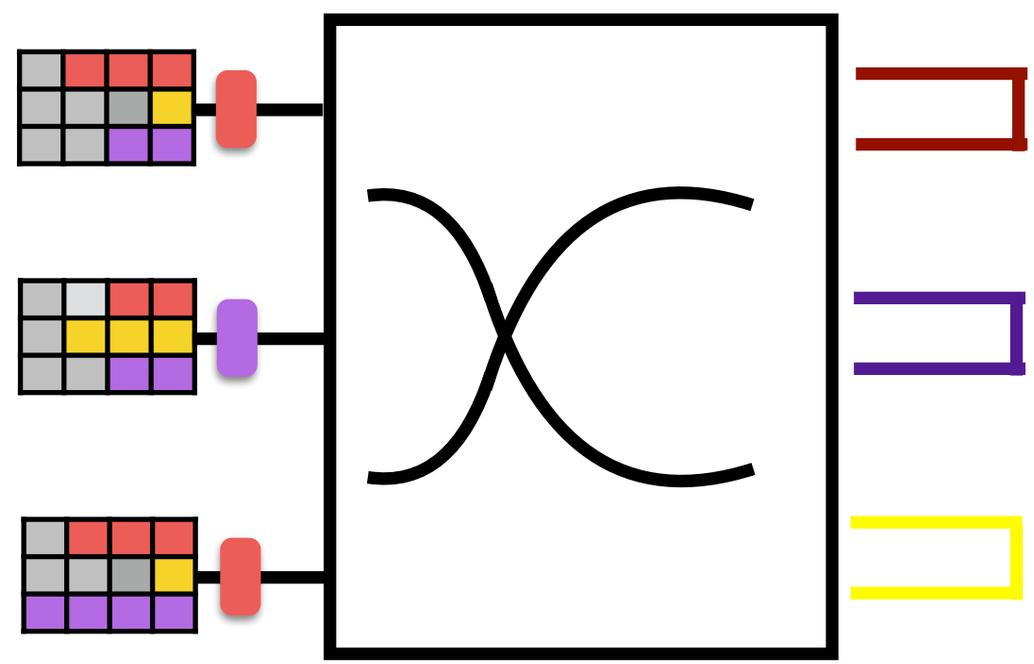
- **Hop-by-hop forwarding**
- **Each router has a "look-up table" (forwarding information base)**
    - What should be stored in this table?
    - Prefix-based forwarding **(longest-prefix matching)**



Network layer — Forwarding

Packet Switched forwarding

- **Hop-by-hop forwarding**
- **Each router has a "look-up table" (forwarding information base)**
    - What should be stored in this table?
    - Prefix-based forwarding **(longest-prefix matching)**
    - Maps **prefixes** to the next-hop



Network layer — Forwarding

Network layer — Forwarding

Packet Switching

Network layer — Forwarding

Packet Switching

- **Goods:**

Network layer — Forwarding

Packet Switching

- **Goods:**
    - No resource underutilization

Network layer — Forwarding

Packet Switching

- **Goods:**
    - No resource underutilization
        - A source can send more if others don’t use resources

Network layer — Forwarding

Packet Switching

- **Goods:**
    - No resource underutilization
        - A source can send more if others don’t use resources
    - No blocked connection problem

Network layer — Forwarding

Packet Switching

- **Goods:**
    - No resource underutilization
        - A source can send more if others don’t use resources
    - No blocked connection problem
    - No per-connection state

Network layer — Forwarding

Packet Switching

- **Goods:**
    - No resource underutilization
        - A source can send more if others don’t use resources
    - No blocked connection problem
    - No per-connection state
    - No set-up cost

Network layer — Forwarding

Packet Switching

- **Goods:**
    - No resource underutilization
        - A source can send more if others don’t use resources
    - No blocked connection problem
    - No per-connection state
    - No set-up cost
- **Not-so-goods:**

Network layer — Forwarding

Packet Switching

- **Goods:**
    - No resource underutilization
        - A source can send more if others don’t use resources
    - No blocked connection problem
    - No per-connection state
    - No set-up cost
- **Not-so-goods:**
    - Packet header overhead

Network layer — Forwarding

Packet Switching

- **Goods:**
    - No resource underutilization
        - A source can send more if others don’t use resources
    - No blocked connection problem
    - No per-connection state
    - No set-up cost
- **Not-so-goods:**
    - Packet header overhead
    - Network failures become a problem