CS4450

Computer Networks:
Architecture and Protocols

Lecture 2
Sharing Networks:
“Circuits” and “Packets”

Prof. Rachit Agarwal

Announcements

- The **webpage** is up!
  - https://www.cs.cornell.edu/courses/cs4450/2025sp/
  - Please read everything on the webpage carefully
    - Especially, Admin page
    - All slides, problem sets, readings, etc. will be on the webpage
    - Solutions etc. will be on Ed Discussions
- You should all be now on **Ed Discussions**
  - If you are unable to access, send us an email: cs4450-staff@cornell.edu
- **Office hours** will be announced this week
  - Office hours start next week
- I do not expect you to read notes/slides before lecture

Announcements

- **Communication with staff in 4450**
  - All enrollment-related questions: courses@cis.cornell.edu
  - Everything: First check the webpage
  - Everything that is not answered on the webpage: Ed discussions
  - **No emails**
- **Exam schedule is on the webpage. There should be on conflicts.**
- **Problem set 1 is released (on course webpage)**

Goal of Today's Lecture

- Learn about:
  - Two important performance metrics:
    - Bandwidth
    - Delay, or latency
    - Why are these important?
  - Two ways of sharing networks:
    - Circuit switching
    - Packet switching
  - **Why do current computer networks use packet switching?**

But first, Recap from last lecture

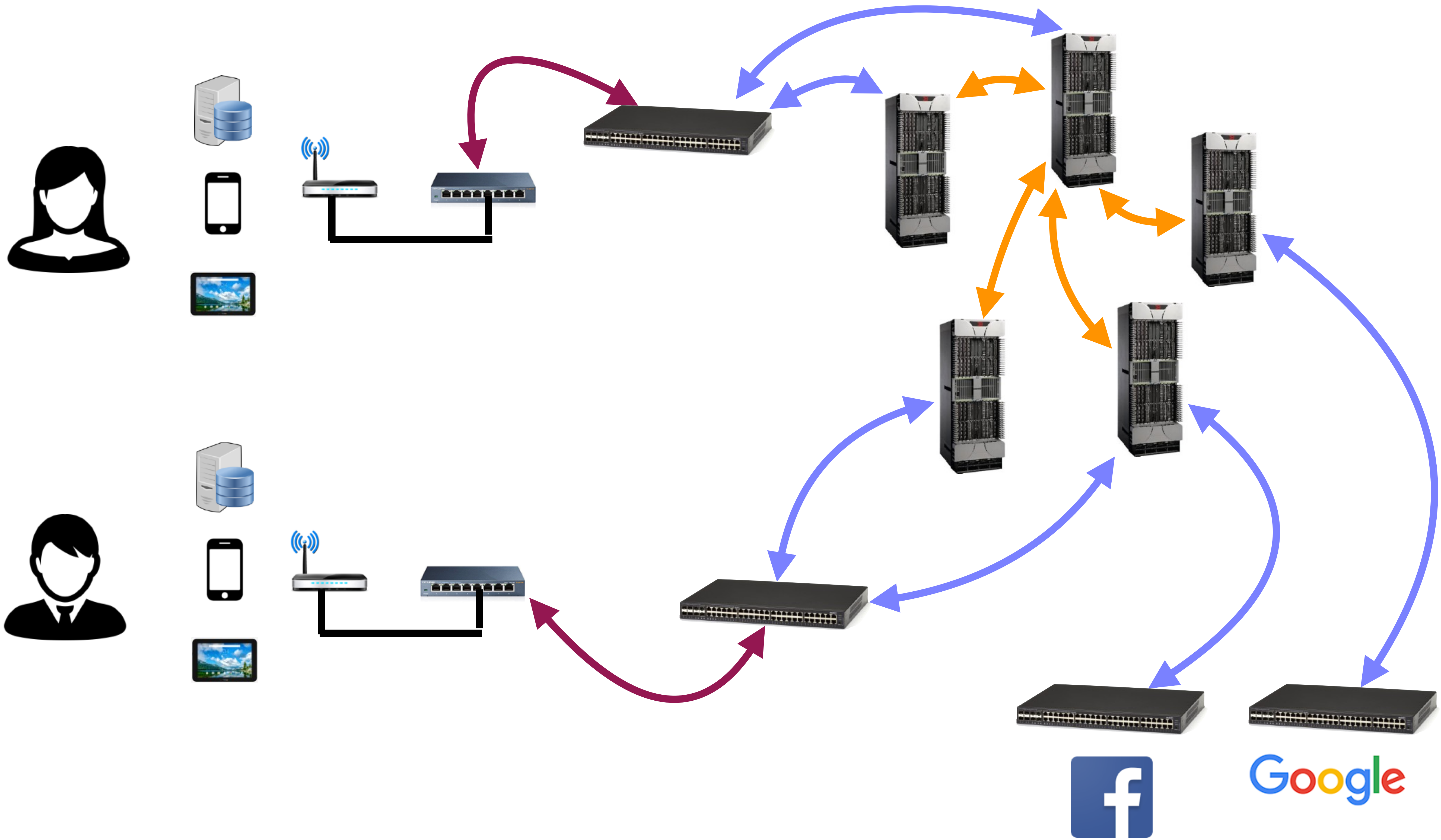
Recap: what is a computer network?

A set of network elements connected together that implement a set of protocols for the purpose of enabling (distributed) applications at end hosts

- **Three important components**:
  - **Core infrastructure**
    - A set of network elements, connected together
  - **Protocols:**
    - Needed to use the network
  - **Purpose:**
    - Enabling (distributed) applications at end hosts

Recap: what is a computer network?

A set of network elements connected together that implement a set of protocols for the purpose of enabling (distributed) applications at end hosts



Recap: what does the core infrastructure include?

Core infrastructure includes:

- **End hosts**: they send/receive data
- **Switches/Routers:** they forward data (chopped into packets)
- **Links:** connect end hosts to switches, and switches to each other

Recap: what do computer networks do?

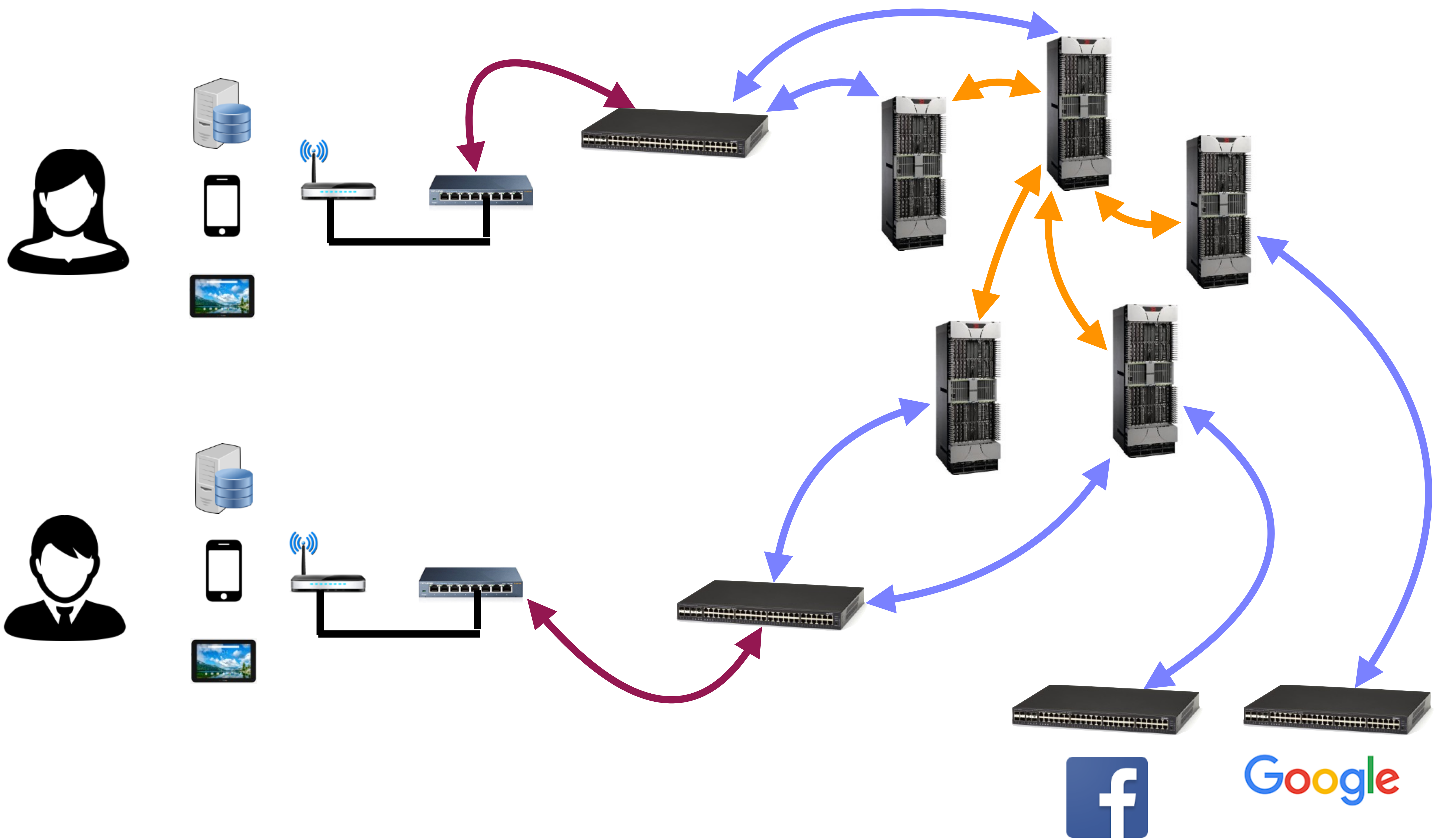
A computer network delivers data between the end points

- **One and only one task**: Delivering the data
- This delivery is done by:
    - Chopping the data into **packets**
    - Sending individual packets across the network
    - Reconstructing the data at the end points

Lets make the picture simpler for today’s lecture

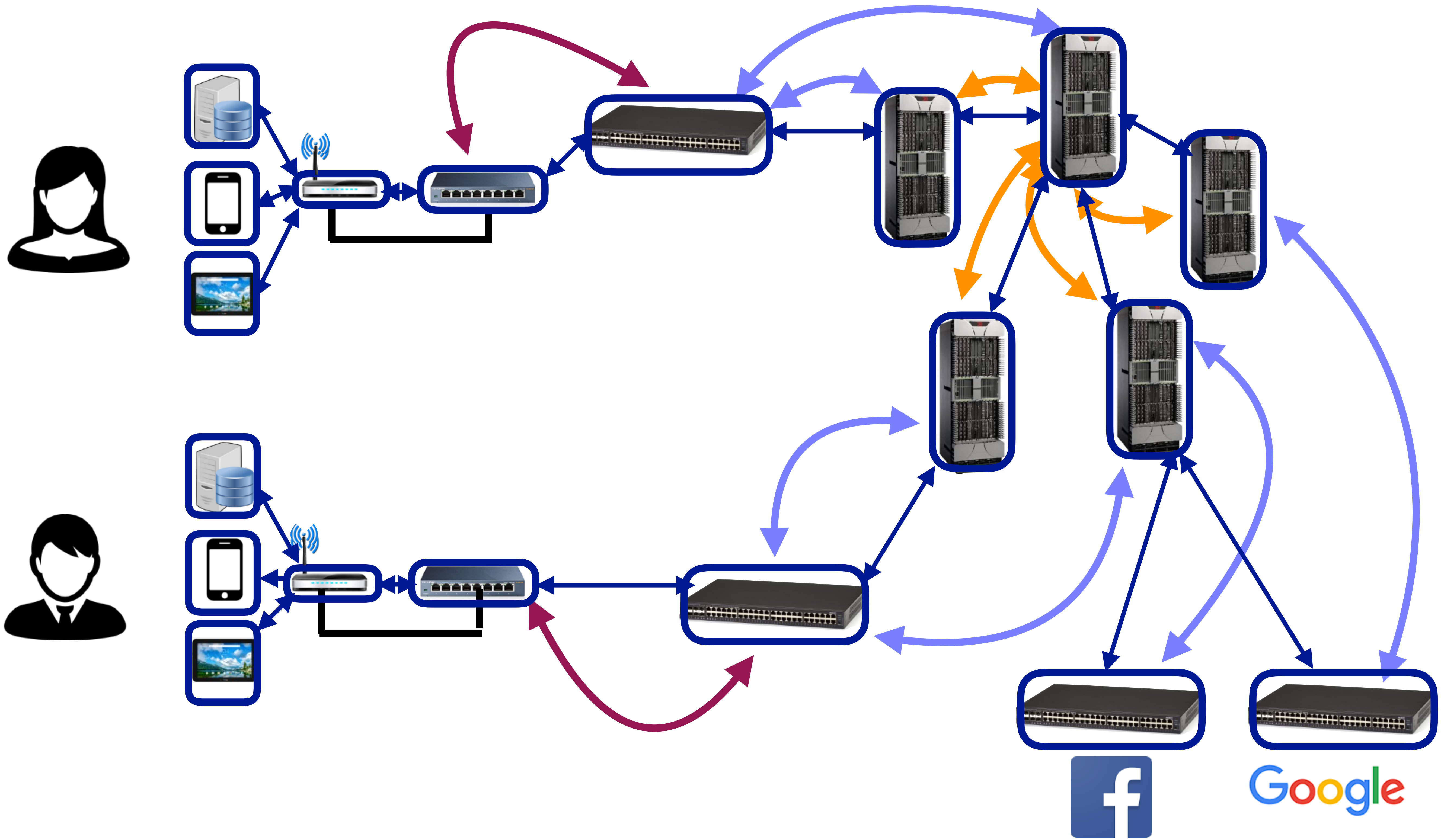
What is a computer network?

A set of network elements connected together that implement a set of protocols for the purpose of enabling (distributed) applications at end hosts

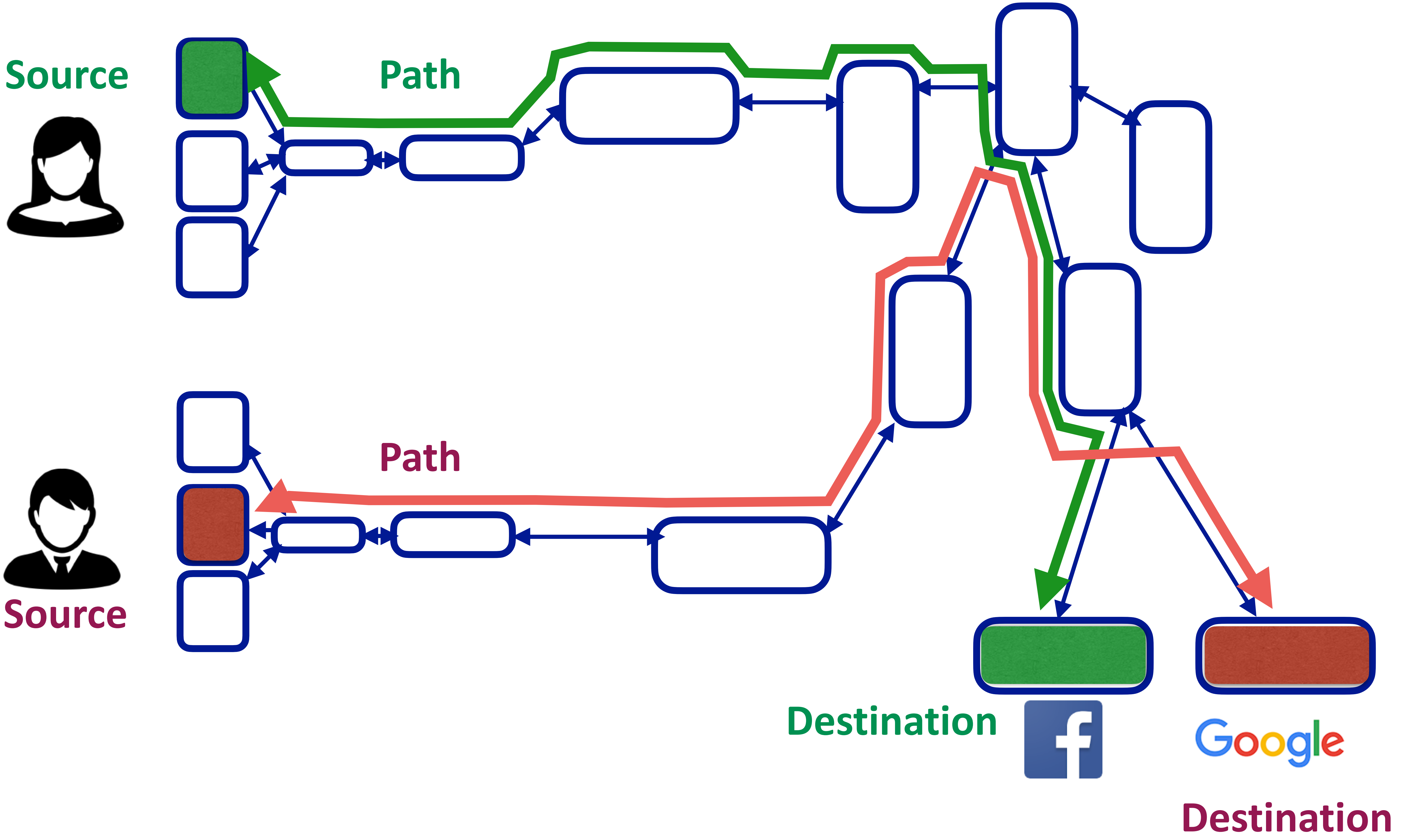


What is a computer network?

A set of network elements connected together that implement a set of protocols for the purpose of enabling (distributed) applications at end hosts



A computer network can be abstractly represented as a graph

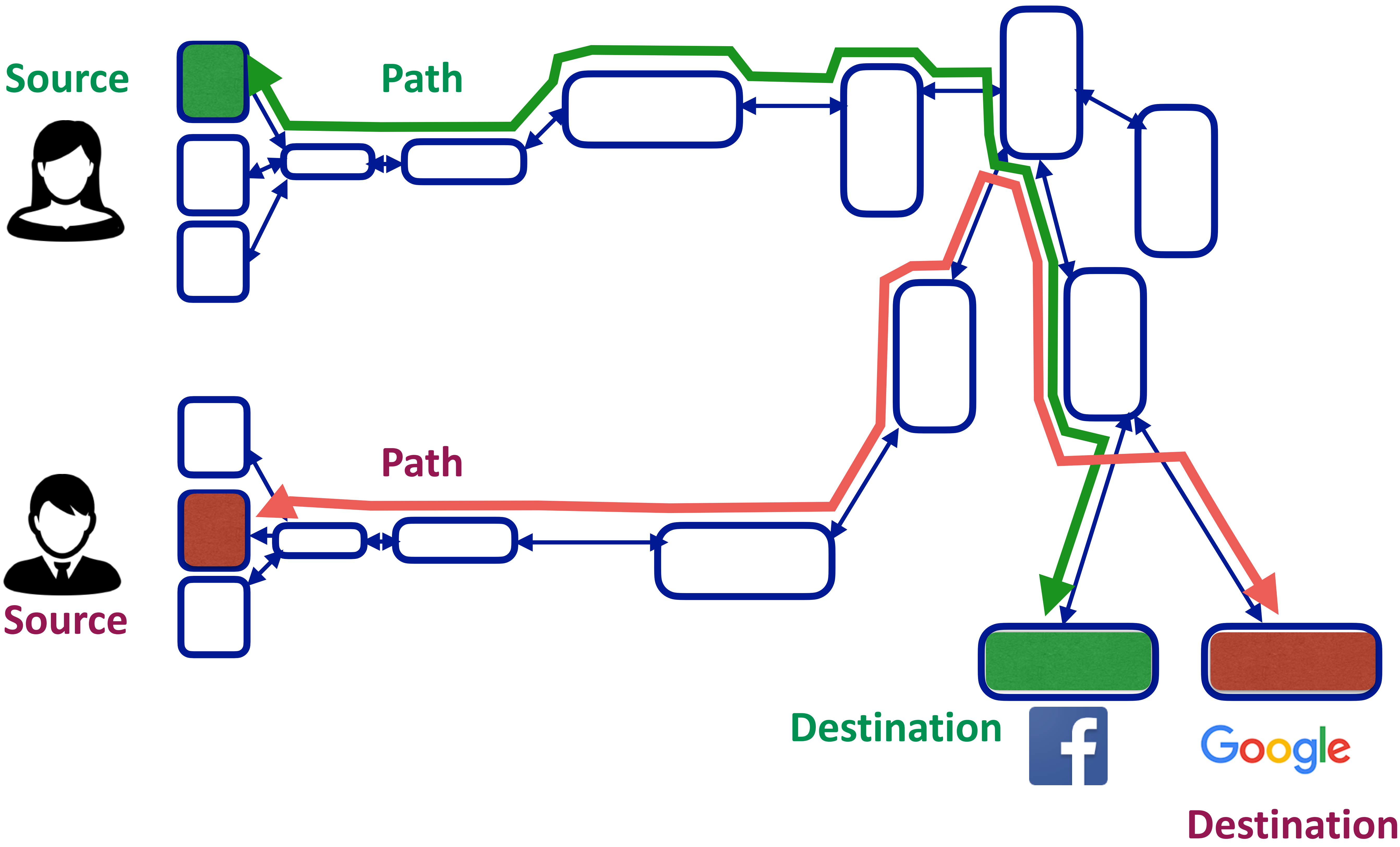


Many mechanisms underneath!

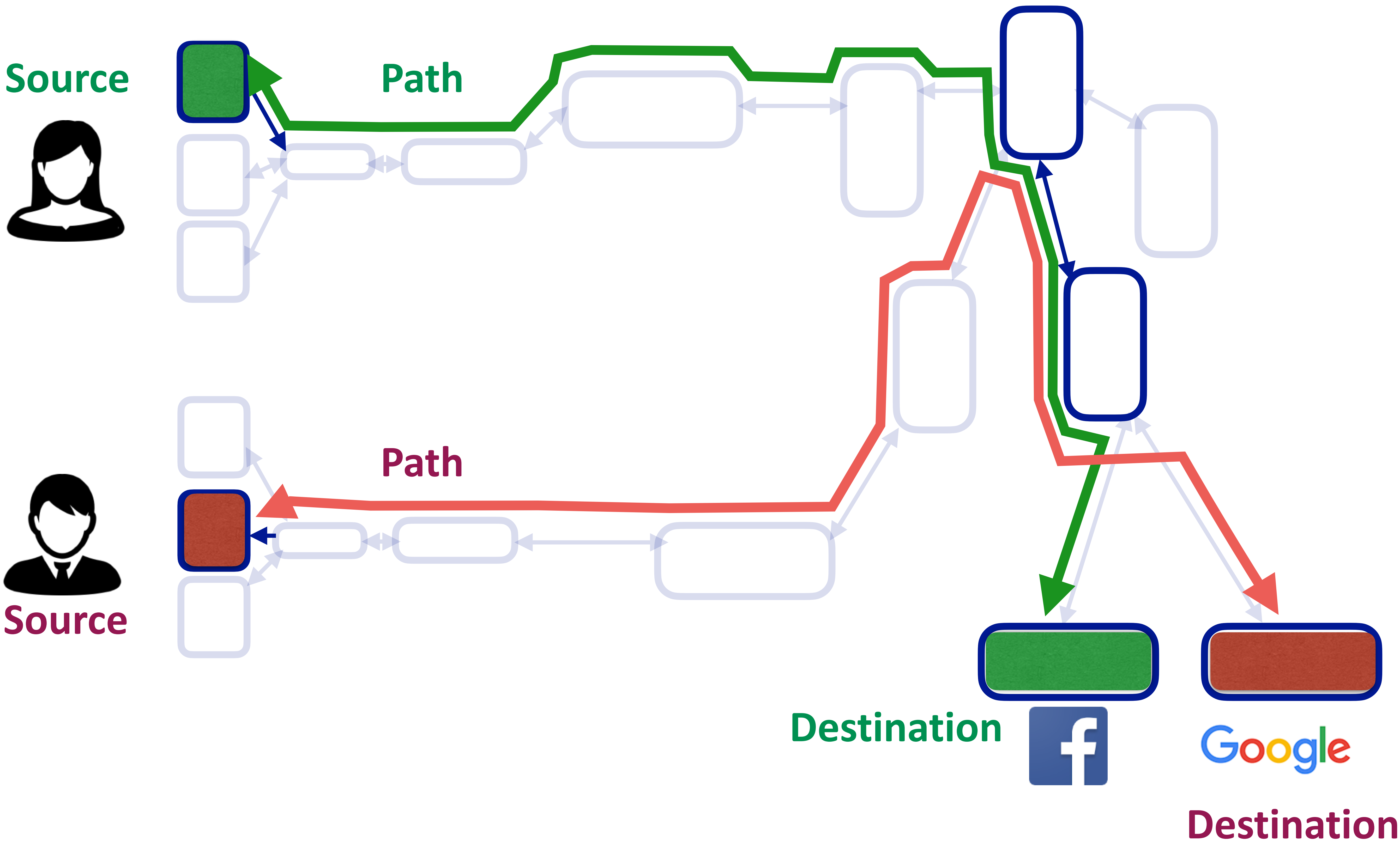
- **Locating the destination**: Naming, addressing
- **Finding a path to the destination**: Routing
- **Sending data to the destination:** Forwarding
- **Failures, reliability, etc.:** Distributed routing and congestion control

Will take the entire course to learn these

A computer network can be abstractly represented as a graph



Today's focus: sharing the network (graph)



Today's lecture: sharing computer networks

1. What does network sharing mean?
2. What are the performance metrics?
3. What are the various mechanisms for sharing networks?
4. **<u>Why</u> "packets" and "flows"?**
5. **Understanding bandwidth and latency for packets**

What does network sharing mean?

The problem of sharing networks

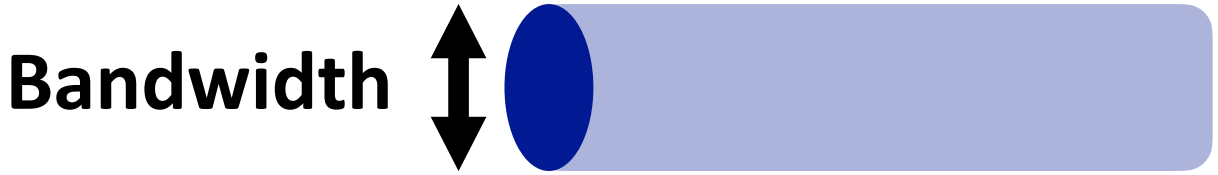
- Must support many "users" and "applications" at the same time
- Each user/application wants to use the network (send and receive data)
- Limited resources
  - We will learn, over the semester, that network has different resources.
- **Fundamental question:**
  - **How does the network decide which resource to allocate to which user/application at any given point of time?**

Resources relate to performance.

What are the performance metrics?

Performance metrics in computer networks!

- **Bandwidth: <u>Number of bits</u> sent per second (bits per second, or bps)**
  - Depends on
    - Hardware
    - Network traffic conditions
    - ….



- **Delay: Time for <u>all bits</u> to go from source to destination (seconds)**
  - Depends on
    - Hardware
    - Distance
    - Traffic from other sources
    - ….
- **Many other performance metrics (reliability, fairness, etc.)**
  - We will come back to other metrics later …

What are the various mechanisms for sharing networks?

Group Exercise 1:

How would you design a sharing mechanism?

Hint:

Think about sharing any resource (say, a computer)

Two approaches to sharing networks

- Reservations
- On demand

Two approaches to sharing networks

- **First: Reservations**
  - Reserve bandwidth needed in advance
  - Set up circuits and send data over that circuit
  - Must reserve for peak bandwidth

- How much bandwidth to reserve?
  - Applications may generate data at rate varying over time
  - 100MB in first second
  - 10MB in second second …
  - Reservations must be made for “peak”

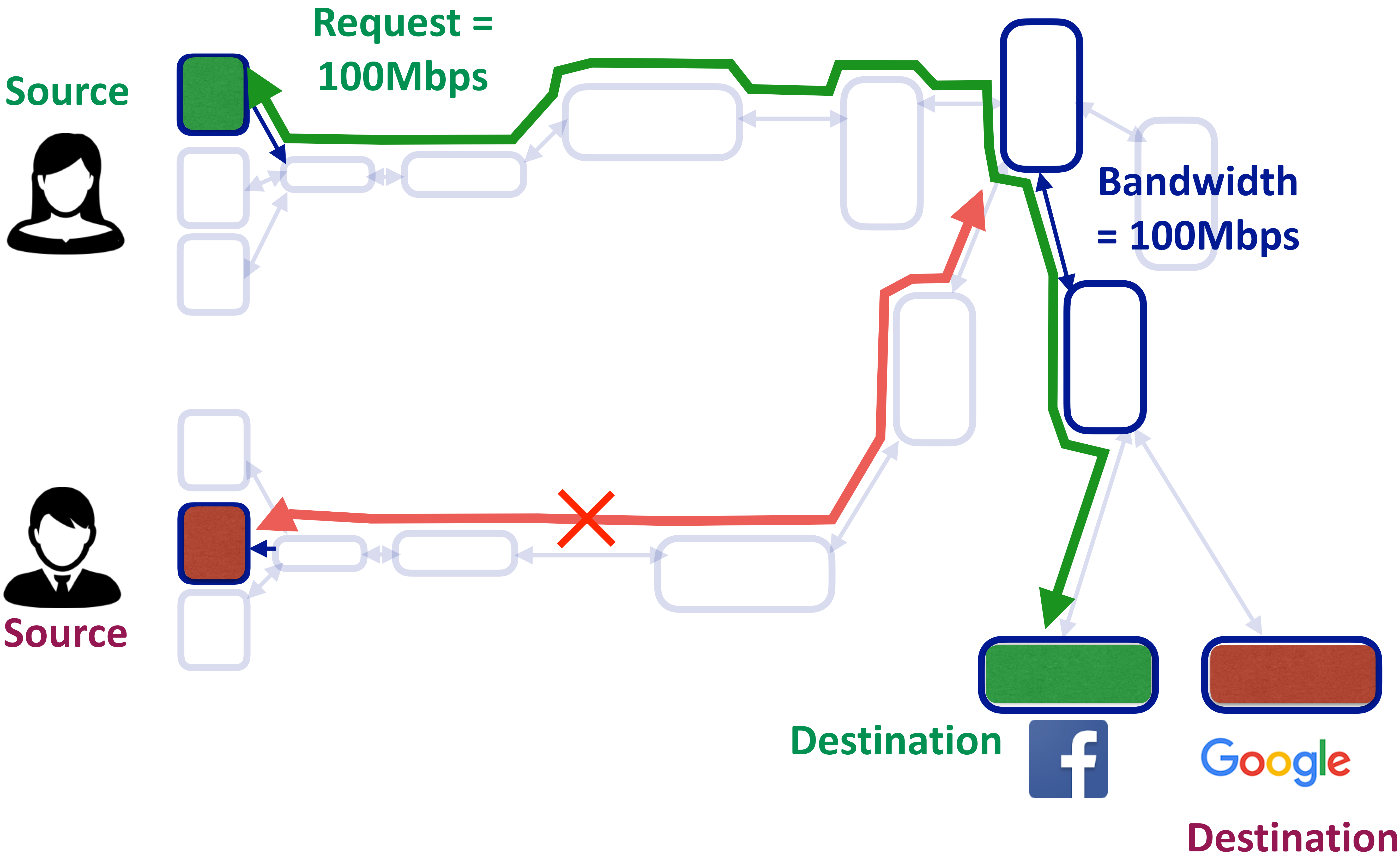
Circuit switching: Implementing reservations since …

Telephone networks

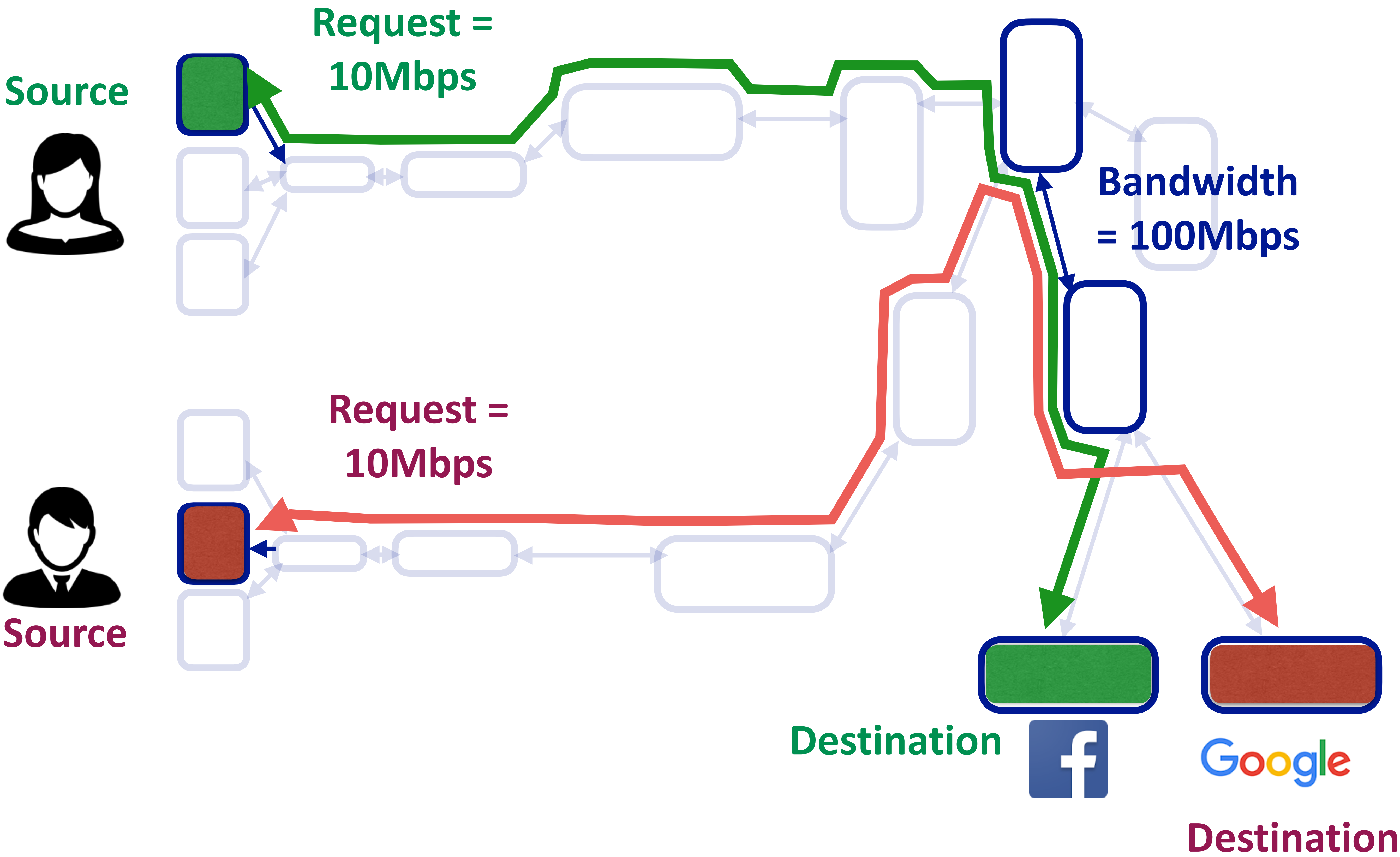
- One of the many approaches to implementing reservations

- **Mechanism:**
    - Source sends a reservation request for peak demand to destination
    - Switches/routers establish a "circuit"
    - Source sends data
    - Source sends a "teardown circuit" message

Circuit switching: an example (red request fails)



Circuit switching: another example (red request succeeds)



Circuit switching and failures

- Circuit is established
- **Link fails along path (!!!!!!!)**
    - First time we have seen failures making our life complicated.
    - Remember this moment.
    - Its gonna happen, over and over again.
- Must establish new circuit

Circuit switching doesn’t route around failures!!

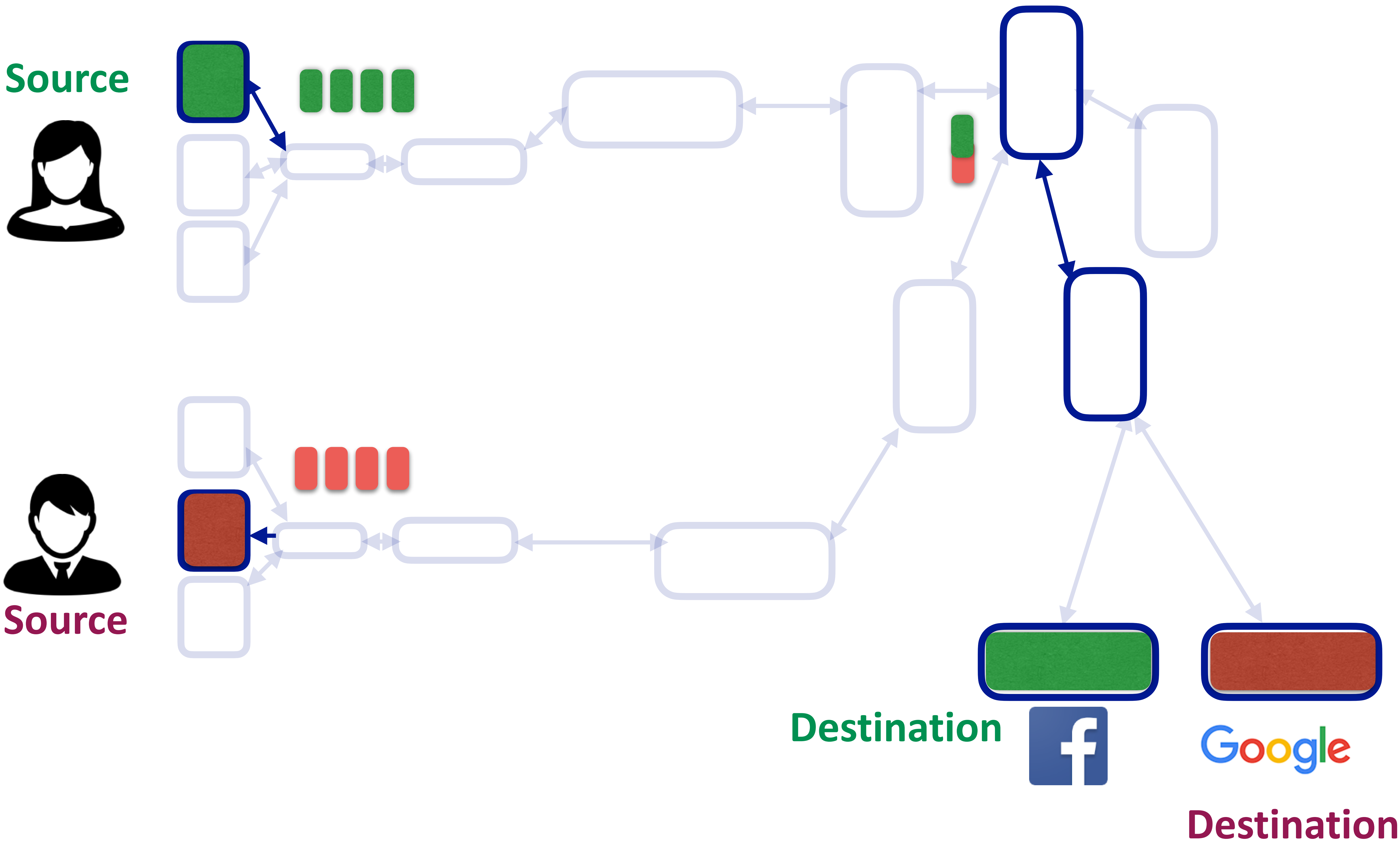
Circuit switching (reservation-based sharing) summary

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

Two approaches to sharing networks

- **Second: On demand (also known as “best effort”)**
  - **Designed specifically for the Internet**
  - Break data into packets
  - Send packets when you have them
  - Hope for the best …

Packet switching: an example



Packets

- **Packets carry data (are bag of bits)**:
  - Header: meaningful to network (and the OS on the end host)
    - can be multiple headers
  - Body: meaningful only to application
  - More discussion in next lecture

- **Body can be bits in a file, image, whatever**
  - can have its own application “header”

- **What information goes in the header?**

What must headers contain to enable network functionality?

- **Packets must describe where it should be sent**
  - Requires an address for the destination host
    - can be multiple headers
- **Packets must describe where its coming from**
  - why?
  - Acknowledgments, etc.
- **Thats the only way a router/switch can know what to do with the packet**

Recap: Packet switching summary

- **Goods:**
  - With proper mechanisms in place
    - Easier to handle failures
  - 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:**
  - Unpredictable performance
  - High latency
  - Packet header overhead

Circuits vs packets

- Pros for circuits:
  - Better application performance (reserved bandwidth)
  - More predictable and understandable (w/o failures)
- Pros for packets:
  - Better resource utilization
  - Easier recovery from failures
  - Faster startup to first packet delivered

Summary of network sharing

Statistical multiplexing

- **Statistical multiplexing:** combining demands to share resources efficiently

- Long history in computer science
    - Processes on an OS (vs every process has own core)
    - Cloud computing (vs every one has own datacenter)

- Based on the premise that:
    - **Peak of aggregate load is << aggregate of peak load**

- Therefore, it is better to share resources than to strictly partition them …

Two approaches to sharing networks

Both embody statistical multiplexing

- Reservation: sharing at <u>connection</u> level
  - Resources shared between connections currently in system
  - **Reserve the peak demand**
- On-demand: sharing at <u>packet</u> level
  - Resources shared between packets currently in system
  - Resources given out on packet-by-packet basis
  - **No reservation** of resources

Understanding delay/latency

Packet Delay/Latency

- Consists of four components
  - **Transmission delay** (hardware properties)
  - **Propagation delay** (hardware properties, distance)
  - **Queueing delay** (traffic, switch internals)
  - Processing delay (switch internals, end hosts)
- First, consider transmission and propagation delays
- Queueing delay and processing delay later in the course

Transmission delay

- How long does it take to push **all the bits of a packet** into a link?
- **Packet size / Link Bandwidth**
- Example:
    - Packet size = 1500Byte
    - Bandwidth = 100Mbps
    - 1500*8/100*1024*1024 seconds
- **Independent of the link length (distance that the packet traverses)**

Propagation delay

- How long does it take to move **one bit** from one end of a link to the other?
- **Link length / Propagation speed of link**
    - Propagation speed ~ some fraction of speed of light
- Example:
    - Length = 30,000 meters
    - Delay = 30*1000/3*100,000,000 second = 100us
- **Independent of packet size and bandwidth**

Group Exercise 2:

How long does it take for a *packet* on a link?

Constraints:

- Packet size = 1000bits
- Rate = 100bps
- Length = 3*speed of light