CS4450

Computer Networks: Architecture and Protocols

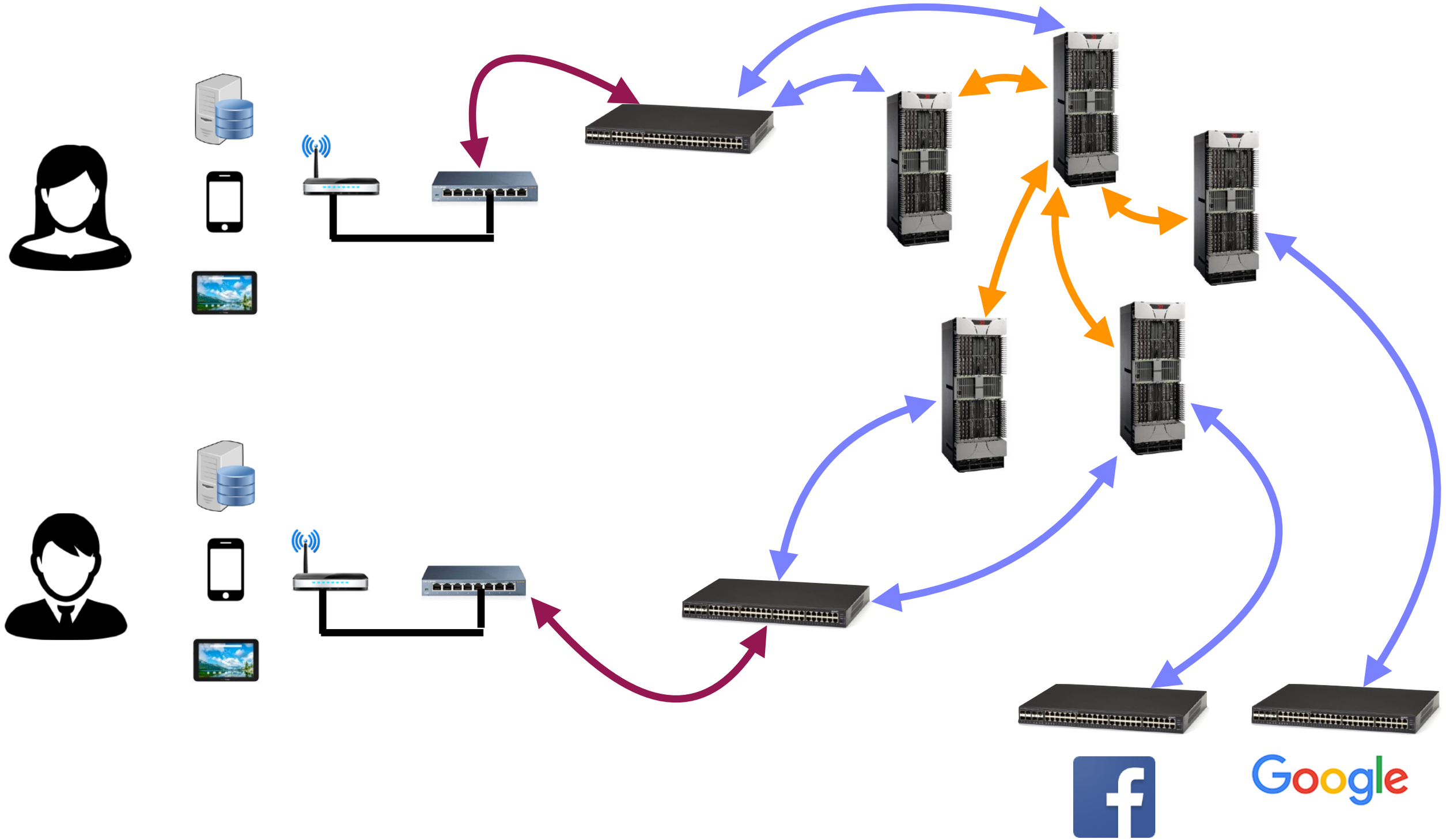
Lecture 20
Putting ALL the Pieces Together

Spring 2018

Rachit Agarwal

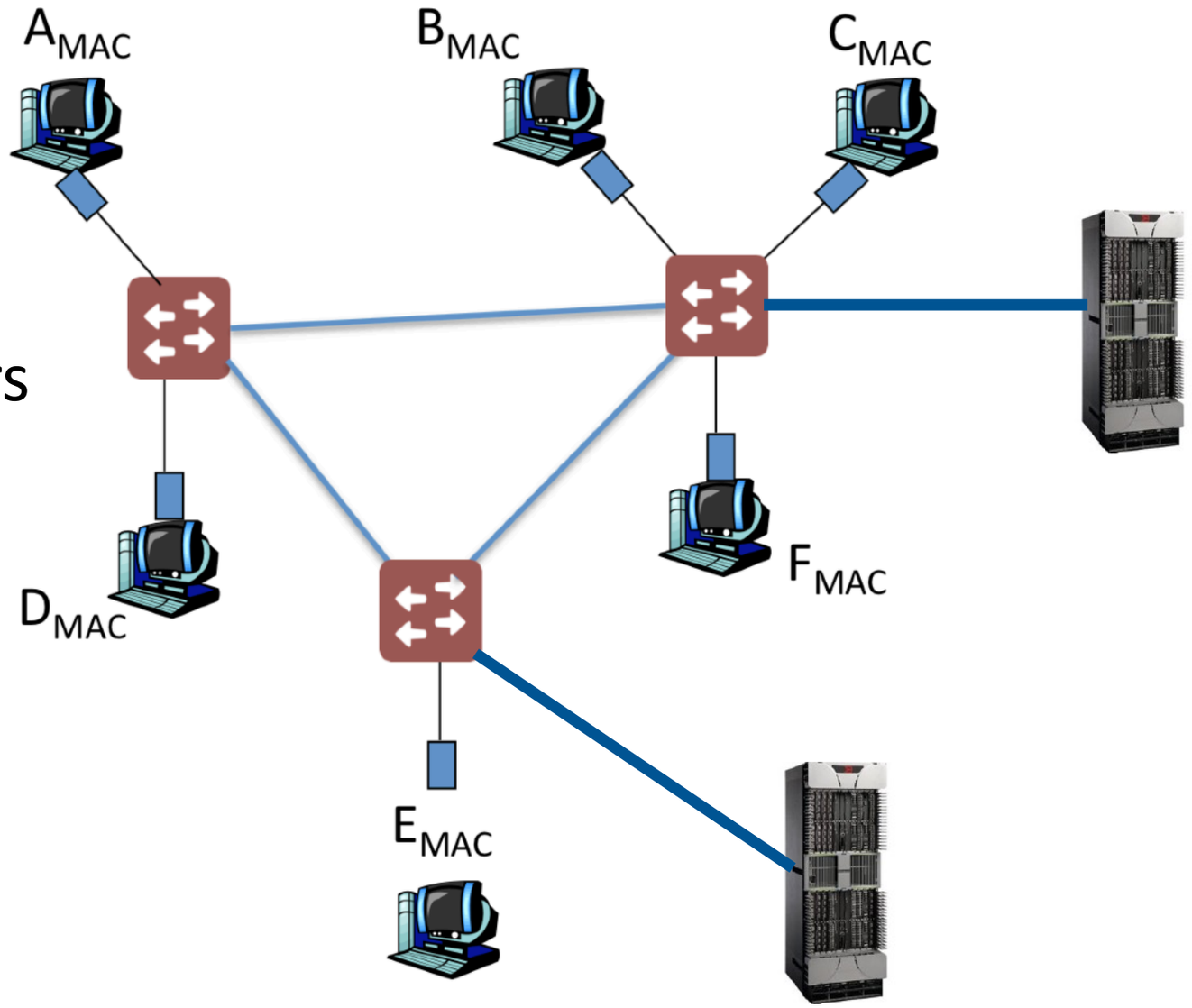
What is a computer network?

A set of network elements connected together, that implement a set of protocols for the purpose of sharing resources at the end hosts

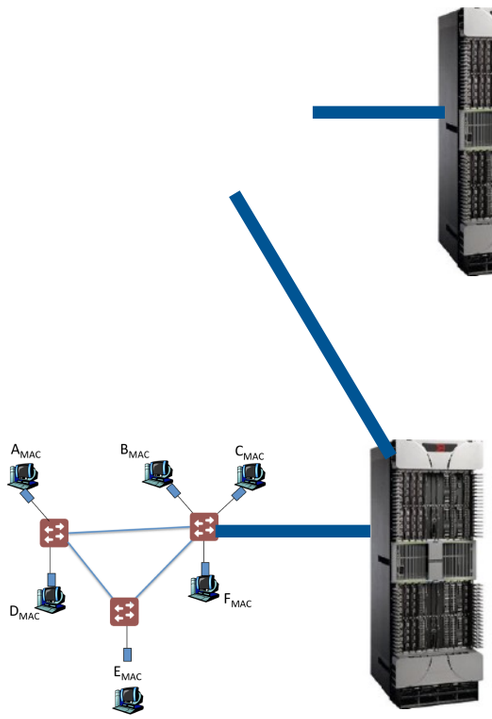


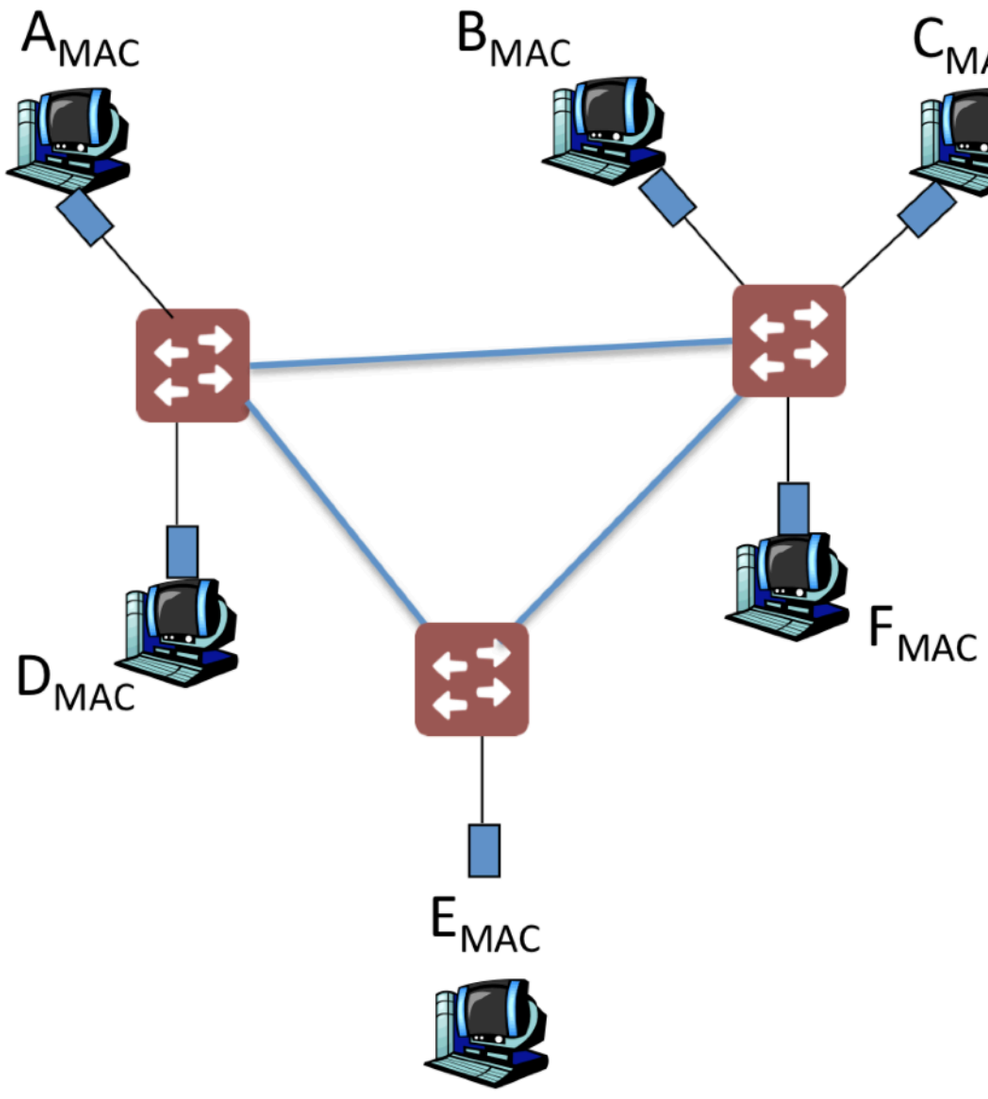
What does Internet actually look like?

- **The smallest component:**
  - A Network Interface Card (NIC), or a machine, or a server
  - Has a Link Layer MAC name/address
- **Multiple NICs connected in a Local Area Network (LAN) via**
  - Broadcast Ethernet,
  - Or, Switched Ethernet
- **Switches in LAN**
  - Connected to larger routers

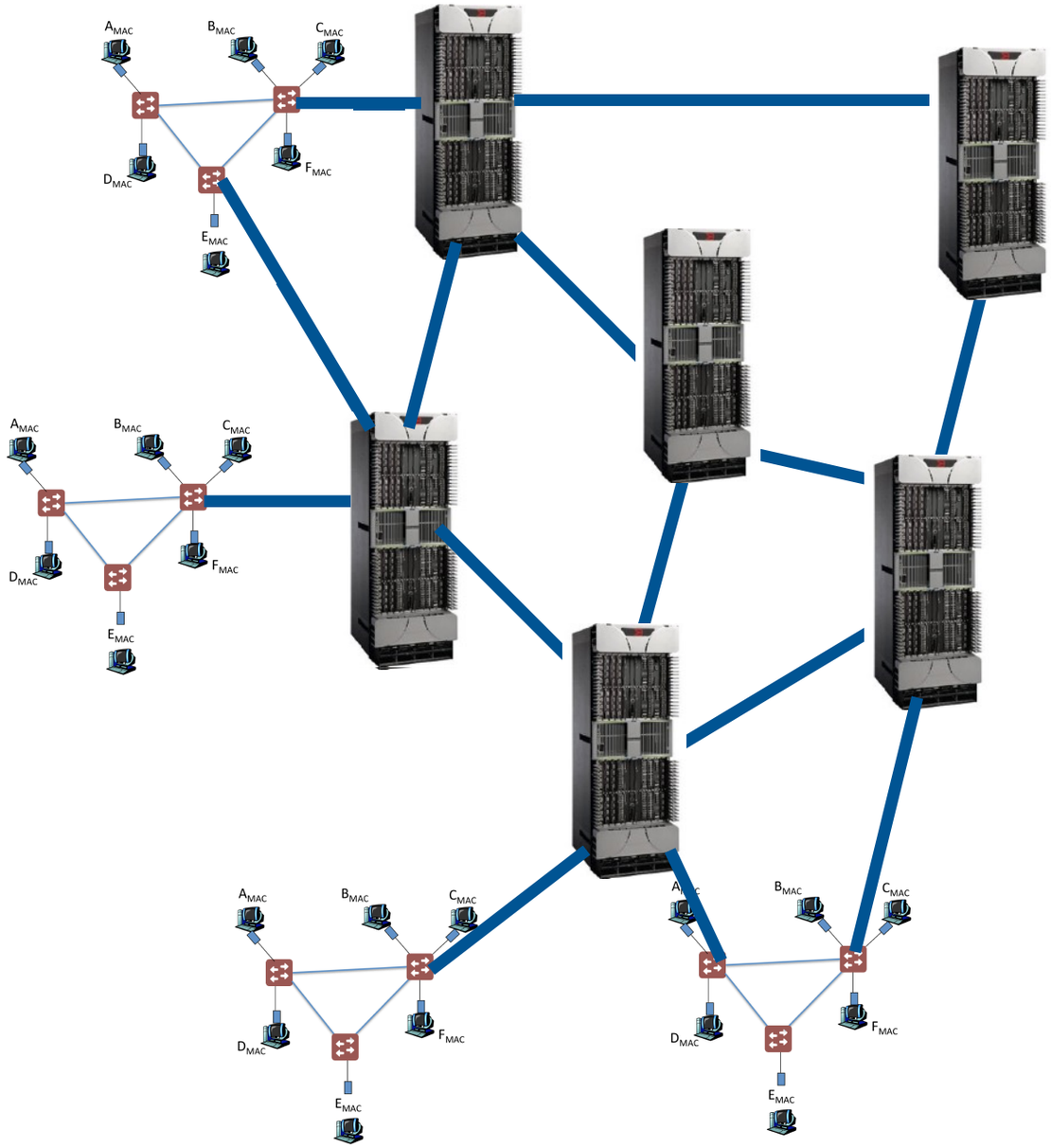


What does Internet actually look like?



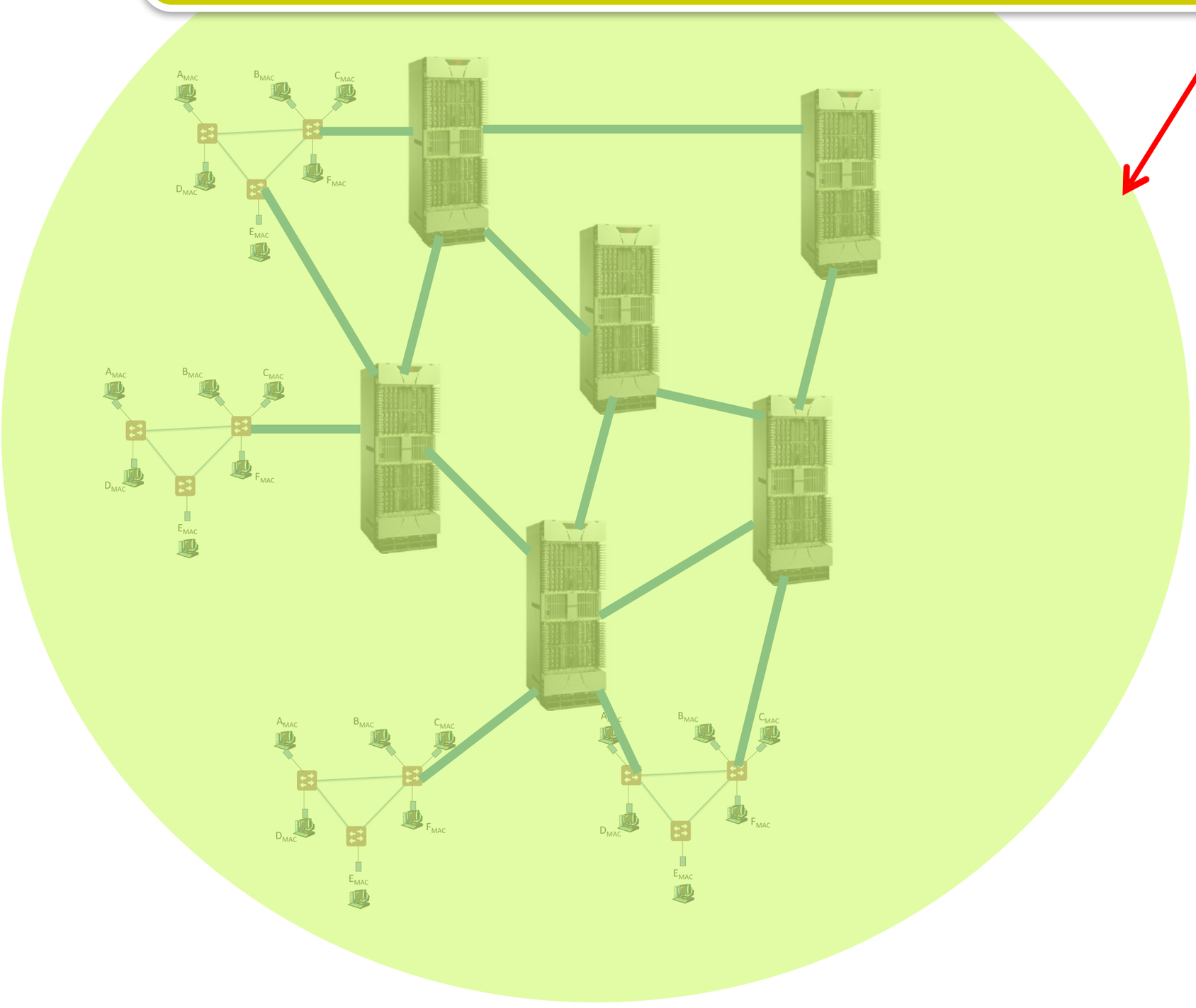


What does Internet actually look like?



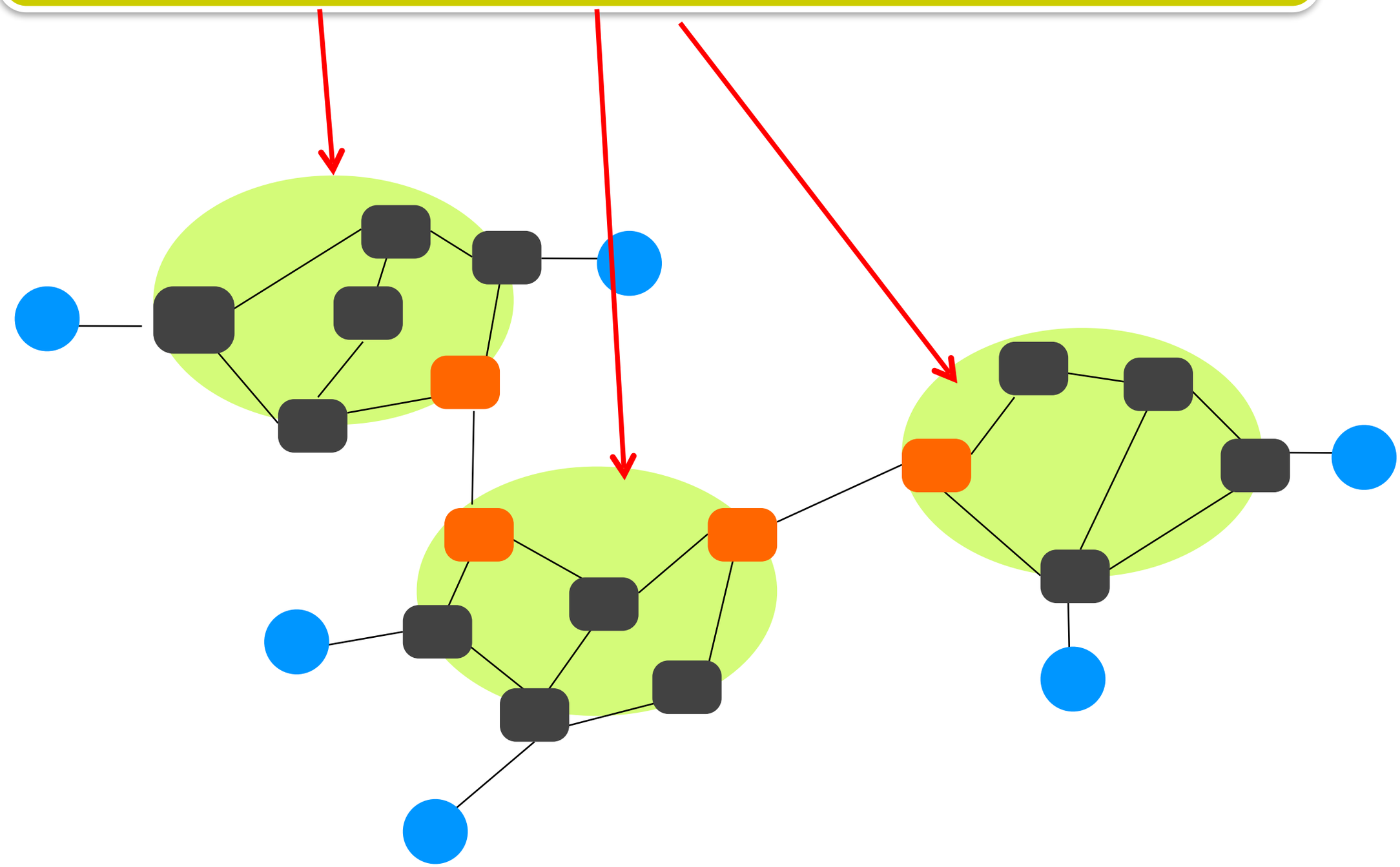
What does Internet actually look like?

"Autonomous System (AS)" or "Domain"
Region of a network under a single administrative entity

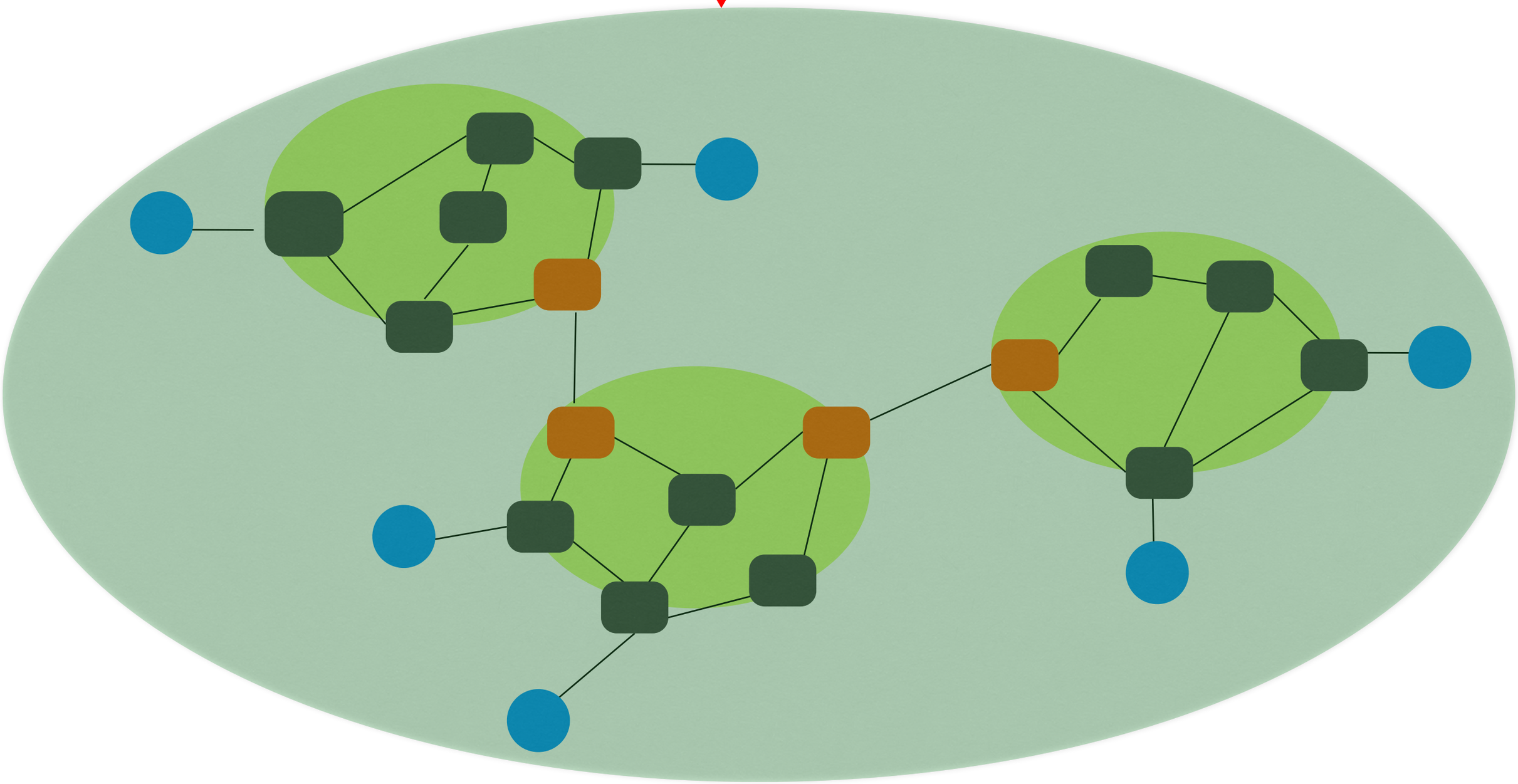


What does Internet actually look like?

Multiple “Autonomous Systems (AS)” or “Domains” connect together using Border Routers



This entire infrastructure is a part of the INTERNET :-)

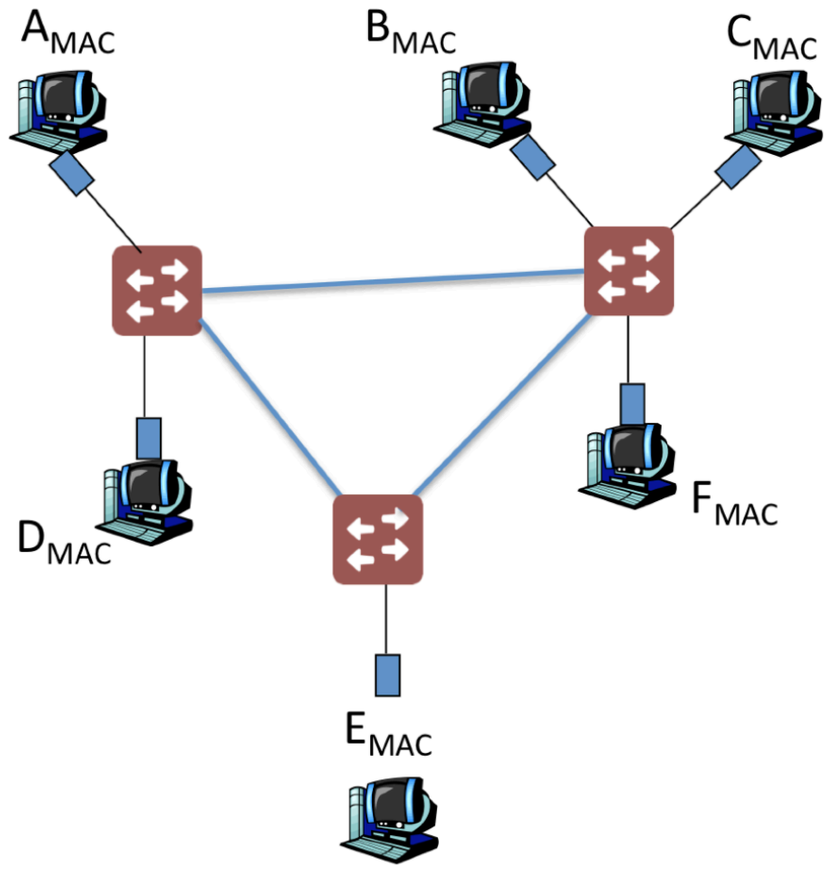


What is the other part of the Internet?

Protocols!

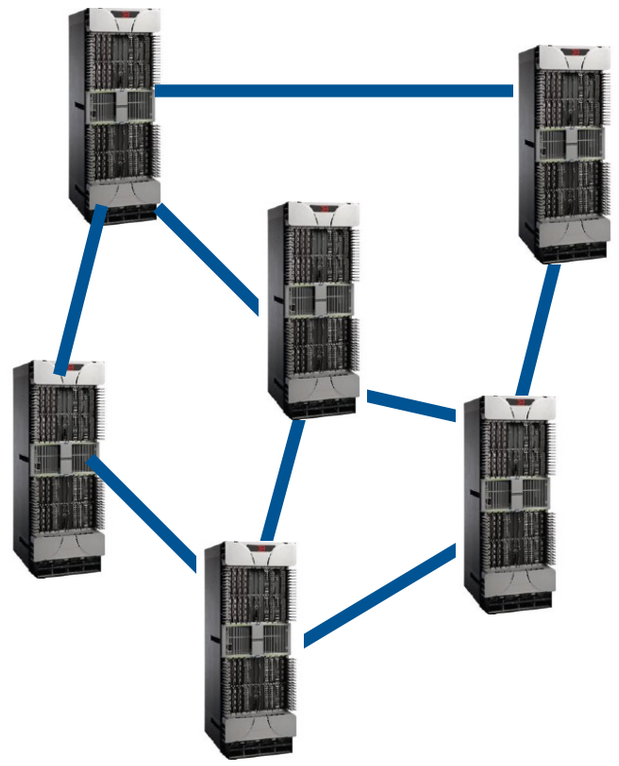
What protocols have we learnt on LAN?

- **Addresses**
    - Link Layer MAC names/addresses: come with the hardware
- **CSMA/CD Protocol:**
    - For transmitting frames on broadcast Ethernet
- **Spanning Tree Protocol:**
    - For transmitting frames on switched Ethernet



What have we learnt beyond LAN?

- **Link-state and Distance-vector Protocols:**
    - For finding routes (and a next-hop) to an IP address within an ISP
- **Border Gateway Protocol:**
    - For finding routes to an IP address range
- **Forwarding at routers**
    - Store **routing tables** (map **destination prefixes** to outgoing port)
    - Longest prefix match for destination address lookup



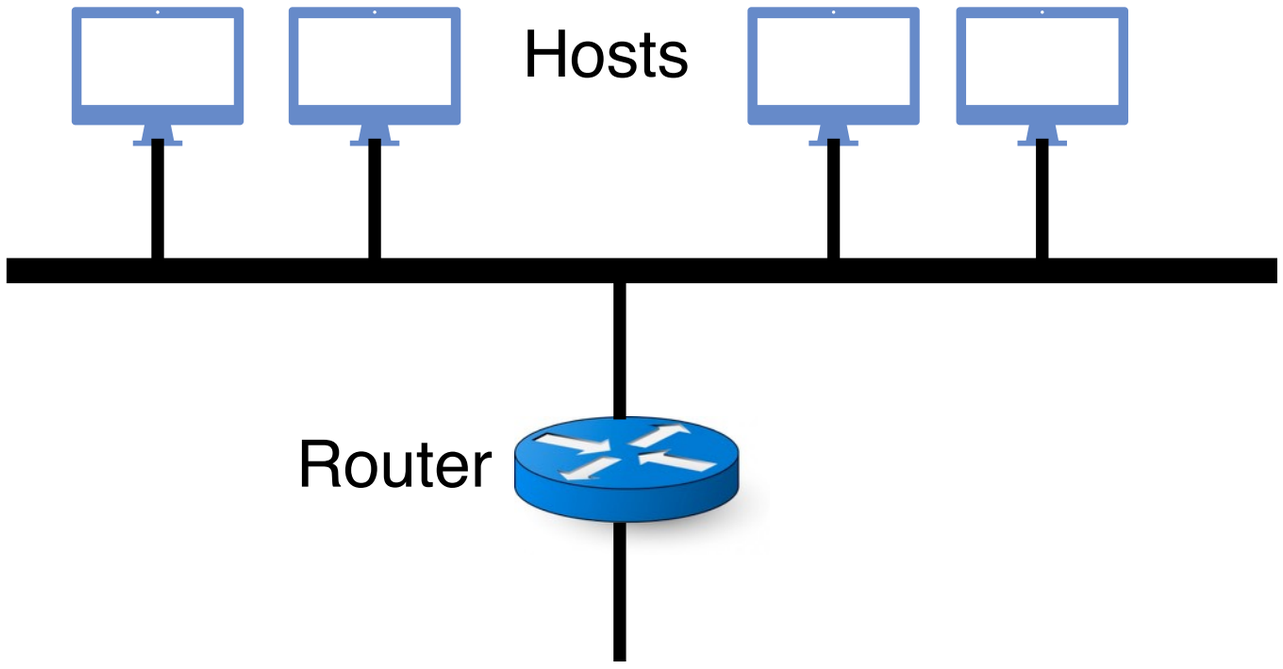
Suppose Host A wants to communication with Host B

Discovery

- Suppose I am host A
- I want to communicate with B (say, www.google.com)
- I was “born” knowing **only** my name — my MAC address :-)
- Must discover some information before I can communicate with B
    - What is my IP address?
    - What is B’s IP address?
        - How do we do this? You already learnt that!
    - Is B within my LAN?
    - If yes, what is B’s MAC address?
    - If not, what is the address of my first-hop router to B?
    - …

DHCP and ARP

- Link layer discovery protocols
    - DHCP — Dynamic Host Configuration Protocol
    - ARP — Address Resolution Protocol
    - Configured to a single LAN
    - Rely on broadcast capability



DHCP and ARP

- Link layer discovery protocols
- Serve two functions
    1. Discovery of local end-hosts
        - For communication between hosts on the same LAN
    2. Bootstrap communication with remote hosts
        - What's my IP address?
        - Who/where is my local DNS server?
        - Who/where is my first hop router?

DHCP

- Dynamic Host Configuration Protocol
    - Defined in RFC 2131
- A host uses DHCP to discover
    - Its own IP address
    - Subnet masks — allows to test whether an IP address is local or not
    - IP address(es) for its local DNS name server(s)
    - IP address(es) for its first-hop “default” router(s)

ARP: Address Resolution Protocol

- Every host maintains an ARP table
  - List of (IP address — MAC address) pairs
  - For IP addresses within the same LAN
- Consult the table when sending a packet
  - Map destination IP address to destination MAC address
- But: what if IP address not in the table?
  - Either its not local (detected using DHCP)
  - If its local:
    - Sender broadcasts: “Who has IP address 1.2.3.156?”
    - Caches the answer in ARP table

Taking Stock: Discovery

Layer	Examples	Structure	Configuration	Resolution Service
App Layer	www.cs.cornell.edu	Organizational hierarchy	~ manual	DNS
Network Layer	123.45.6.78	Topological hierarchy	DHCP	ARP
Link Layer	45-CC-4E-12-F0-97	Vendor(flat)	Hard-coded	

How does the Internet work?

Are you ready?

(Count the number of protocols used for each packet)

How does Internet work — end-to-end?

- Network stack receives the packet from the application (roughly speaking)
- What is my IP address? (using DHCP)
- What is the destination IP address? (using DNS)
- Is destination IP address within my LAN? (using DHCP)
- **If destination IP address local:**
    - What is destination MAC address (using ARP)?
    - Convert packet into frames with correct source/destination address
    - Convert frames into bits
    - Forward the bits to the wire …
- **Each switch:**
    - Forwards to destination (using STP/CSMA/CD)

How does Internet work — end-to-end?

- Network stack receives the packet from the application (roughly speaking)
- What is my IP address? (using DHCP)
- What is the destination IP address? (using DNS)
- Is destination IP address within my LAN? (using DHCP)
- **If destination IP address remote:**
    - **What is my first-hop router IP address? (using DHCP)**
    - **What is my first-hop router MAC address? (using ARP)**
    - Convert packet into frames with correct source/destination address
    - Convert frames into bits
    - Forward the bits to the wire …
- **Each router ….**

How does Internet work — end-to-end?

A router upon receiving a packet (implicit questions)

- **Is the destination in a LAN connected to me?**
    - Forward the packet to the destination
    - Using STP/CSMA/CD
- **Is the destination not in my LAN but in my ISP?**
    - Forward the packet to the next-hop router towards the destination
    - Using distance-vector routing algorithm
- **Is the destination in a different ISP?**
    - Forward the packet to the next-hop router towards the destination
    - Using BGP routing algorithm

Are We There Yet?

- Yes!
- How can we be sure?
- Lets go back to where we started ….

Recall the end-to-end story from our fifth lecture :-)

- Application opens a **socket** that allows it to connect to the **network stack**
- Maps **name** of the web site to its **address** using **DNS**
- The network stack at the source embeds the address and **port** for both the source and the destination in **packet header**
- Each **router** constructs a **routing table** using a distributed algorithm
- Each router uses destination address in the packet header to look up the **outgoing link** in the routing table
    - And when the link is free, forwards the packet
- When a packet arrives the destination:
    - The network stack at the destination uses the port to forward the packet to the right application