

Notes for ECE 46300 - Introduction To Computer Communication Networks

Zeke Ulrich

September 3, 2025

Contents

<i>Course Description</i>	1
<i>Computer Networks</i>	2
<i>Packets</i>	4
<i>Circuit Switching</i>	4
<i>Packet Switching</i>	4
<i>Packet Headers</i>	5

Course Description

An introduction to the design and implementation of computer communication networks. The focus is on the concepts and the fundamental design principles that have contributed to the global Internet success. Topics include: digital transmission and multiplexing, protocols, MAC layer design (Ethernet/802.11), LAN interconnects and switching, congestion/flow/error control, routing, addressing, performance evaluation, internetworking (Internet) including TCP/IP, HTTP, DNS etc. This course will include one or more programming projects.

Computer Networks

The high-level question this course will answer is "how do computers reliably communicate?"

The answer is through computer networks, a group of interconnected nodes or computing devices that exchange data and resources with each other.

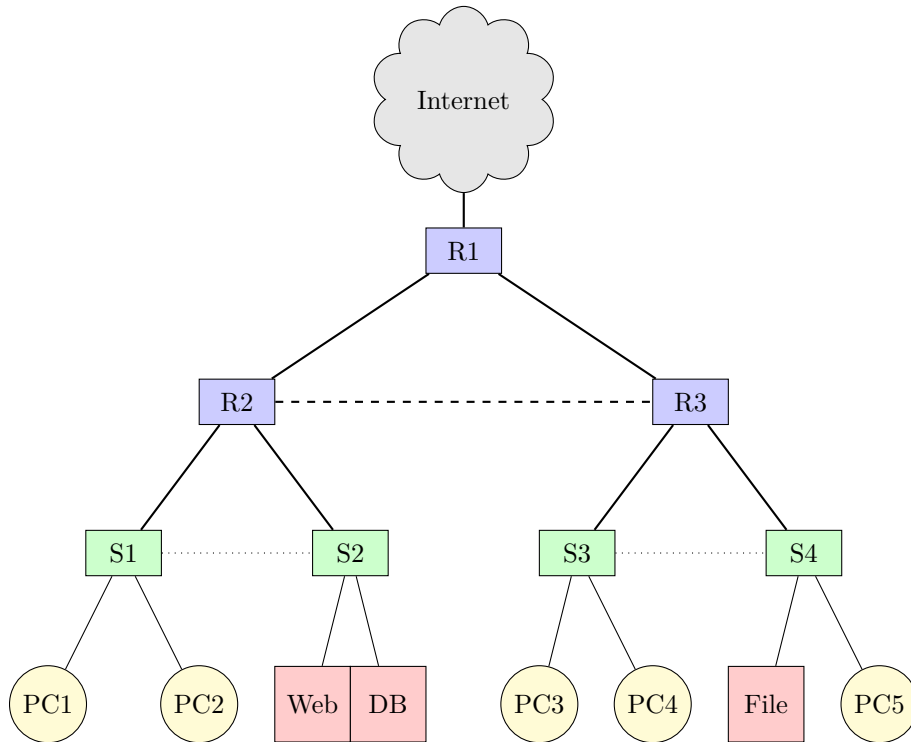


Figure 1: Computer Network

A computer network enables communication between users and their devices.

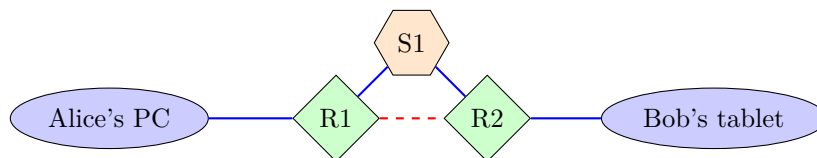


Figure 2: Simple Network

The first step that most home devices take in connecting to other devices is the router. From there, the router can route data and find a path for Alice's PC to get to Bob's tablet. The core of any network is routers, which figure out the best way to get data from one device to another device.

This process can get complex, with cell tower connections, different ISPs, different edge devices, and more. Let's abstract the important elements of the network.

- Links: carry data from one endpoint to another
- End hosts: sitting at the edge of a network. Generate and receive data.
- Routers: forward data through the network.

Any network can be abstracted as a connection of links, end hosts, and routers. We can thus represent any computer network as a graph, in the mathematical sense, and apply all our graph algorithms to it.

Packets

In our abstraction of routers, we leave the problem of finding a path from end hosts unanswered. Since we can represent a computer network as a graph, a shortest-path algorithm like Dijkstra's is a relatively simple way to find a good path between Alice and Bob.

We wish to answer the question "how do users access shared network resources?" Assume no coordination between users and assume users initiate access.

There are two ways to answer this:

- Reserving resources (circuit switching).
- On-demand (packet switching)

Circuit Switching

The idea at the core of this method is that if Alice wants to communicate with Bob, then they reserve a path through the network to do that. The reserved path is called a *circuit*. They then send data along the reserved path.

A pro of circuit switching is that users are guaranteed to have a path through the network. No queuing, no waiting. The routers along the path also don't need to make any decisions in real time. They know the path in advance.

A con is that a resource could be reserved but not used. When users don't coordinate, circuit switching leads to very inefficient use of resources. Another con is that circuits need to be set up and torn down. Imagine the overhead of old telephone networks, where workers had to manually connect telephone wires from one caller to another.

In classical circuit switching, all the resources along the path are reserved for a single circuit and there is no path sharing amongst multiple circuits. In virtual circuit switching, each circuit reserves a subset of resources along its path. It's still reservation based, but two or more circuits can share a same resource (e.g. if two users want to send data that takes up half the bandwidth of a connection, they can both send it at once).

Packet Switching

To overcome the shortcomings of circuit switching, packet switching was invented. The way this works is data is broken into small units called *packets*. You send packets over the network whenever you have them and trust the routers to figure out the path your packets take on the fly.

Each packet needs to have metadata that describes things like the destination of the packets, the order of the packets, and so on.

A pro of packet switching is that we have much better resource utilization. Also, there's no overhead of circuit setup and teardown.

A con of packet switching is that there's no guarantee that a given user will have the resources to send their data. Now we also have overhead from the packet headers, and routers now need to process packet headers and find paths on the fly.

Although both have advantages and disadvantages, modern networks almost exclusively use packet switching and unless otherwise specified all networks in this class are assumed to be packet switched.

Packet Headers

Packet headers require, at least, the following:

- Destination address, used by network to send packet to destination
- Destination port, used by network stack at the destination for application multiplexing
- Source address, used by network to send packet back to source
- Source port, used by network stack at the source for application multiplexing