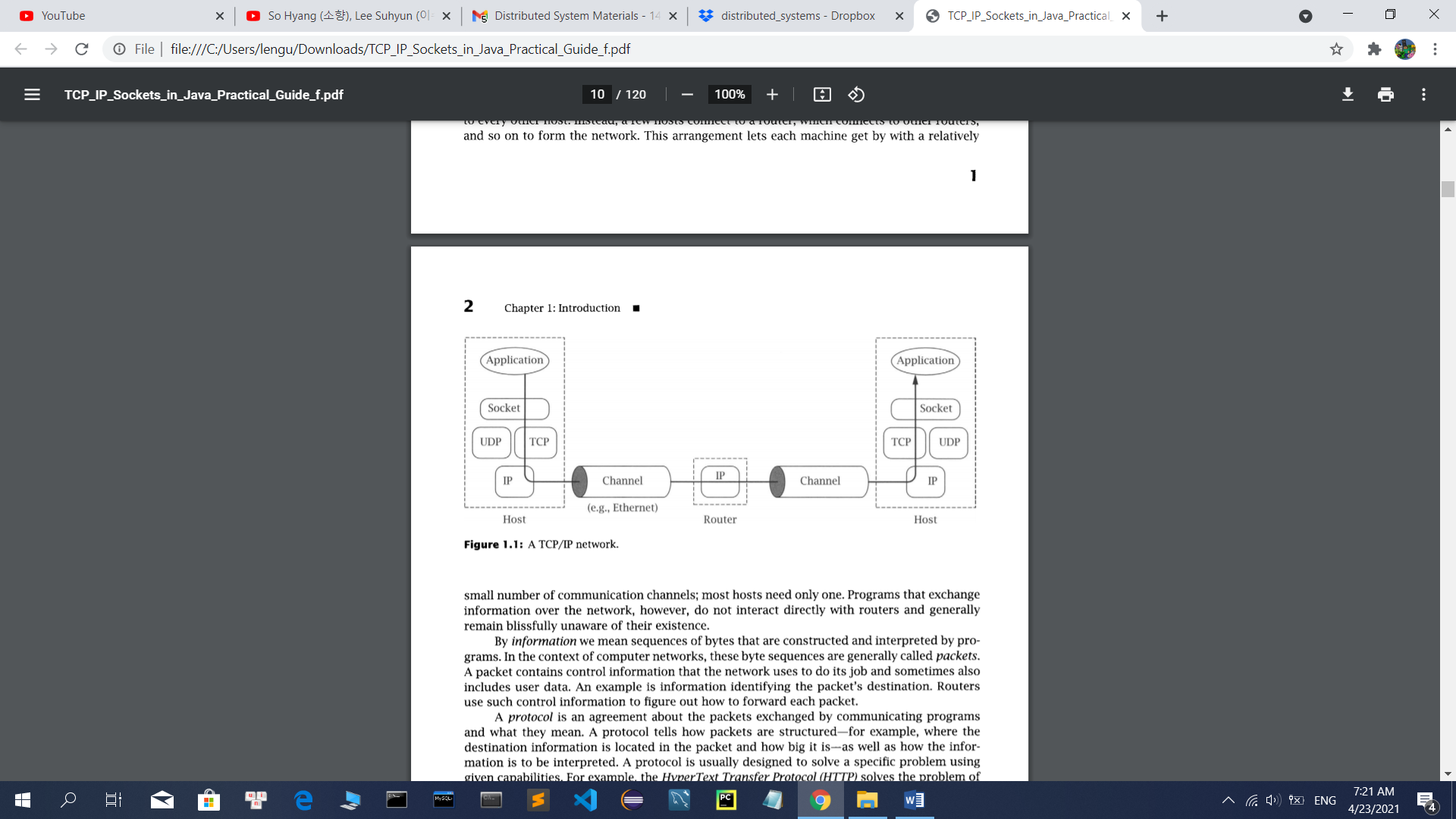
**READING ASSIGNMENTS FOR LESSON 3**

**Networks, Packets and Protocols:**

Hosts are computers that run applications such as your Web browser. Routers are machines whose job is to relay, or forward, information from one communication channel to another. A few hosts connect to a router, which connects to other routers



A protocol is an agreement about the packets exchanged by communicating programs and what they mean. The main protocols in the TCP/IP suite are the Internet Protocol (IP), the Transmission Control Protocol (TCP), and the User Datagram Protocol (UDP).

In TCP/IP, the bottom layer consists of the underlying communication channels. Those channels are used by the network layer, which deals with the problem of forwarding packets toward their destination.

The layer above IP is called the transport layer. It offers a choice between two protocols: TCP and UDP. Each builds on the service provided by IP, but they do so in different ways to provide different kinds of transport, which are used by application protocols with different needs. Both TCP and UDP use addresses, called port numbers, to identify applications within hosts.

**Addresses:**

In TCP/IP, it takes two pieces of information to identify a particular program: an Internet address, used by IP, and a port number, the additional address interpreted by the transport protocol (TCP or UDP).

Internet addresses are 32-bit binary numbers. We typically show them as a string of four decimal numbers separated by periods (e.g., 10.1.2.3). One special IP address worth knowing is the loopback address, 127.0.0.1.

**Name:**

Resolve names into addresses:

First, names are generally easier for humans to remember than dotted-quads. Second, names provide a level of indirection, which insulates users from IP address changes.

The name-resolution service can access information from a wide variety of sources. Two of the primary sources are the Domain Name System (DNS) and local configuration databases.

**Client and Server:**

The client program initiates communication, while the server program waits passively for and then responds to clients that contact it. Together, the client and server compose the application

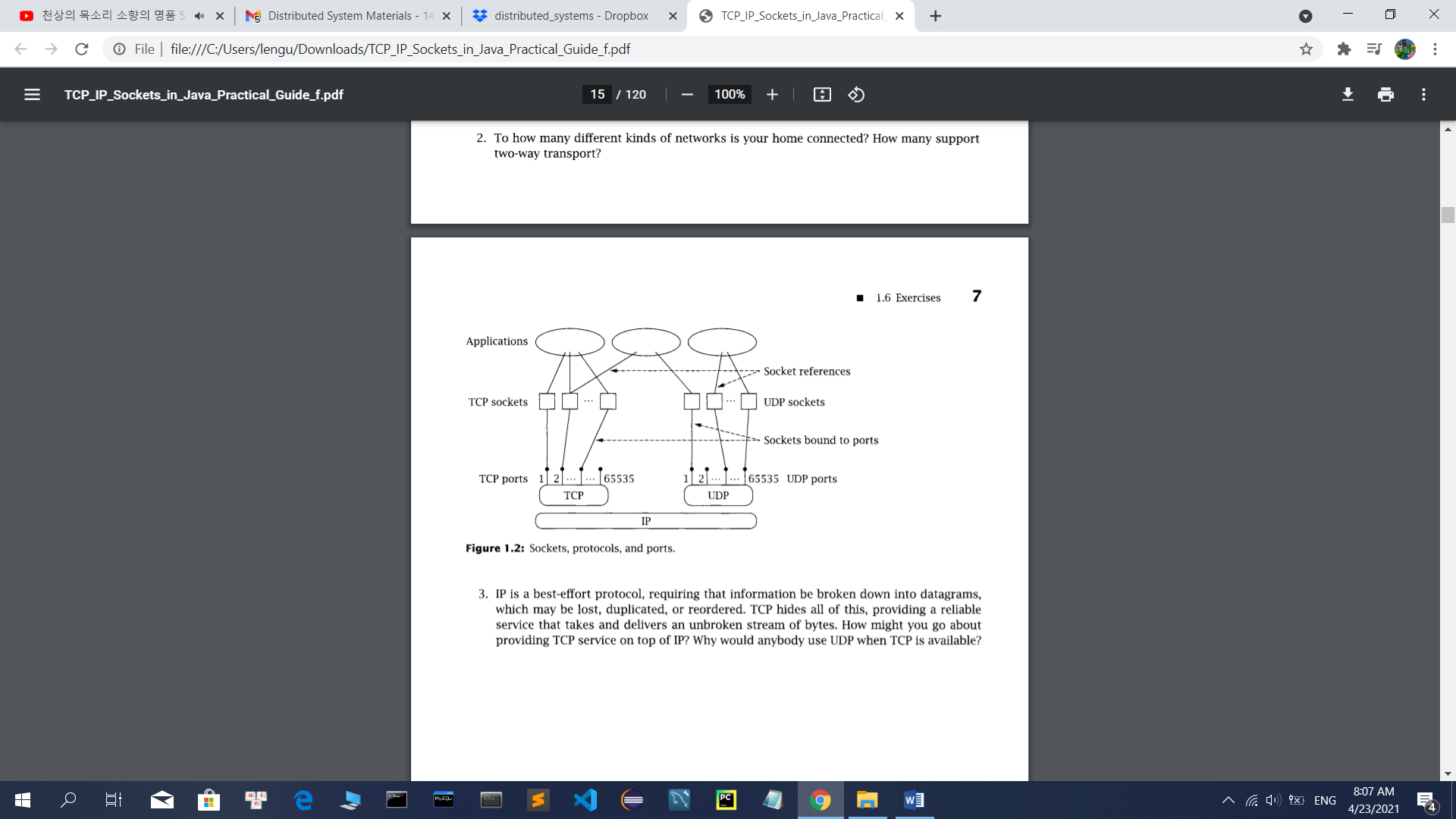
With the sockets API, the server can, if necessary, learn the client's address information when it receives the initial communication from the client

**Socket:**

A socket allows an application to plug in to the network and communicate with other applications that are plugged in to the same network. Information written to the socket by an application on one machine can be read by an application on a different machine and vice versa

The main types of sockets in TCP/IP today are stream sockets and datagram sockets. Stream sockets use TCP as the end-to-end protocol (with IP underneath) and thus provide a reliable bytestream service. Datagram sockets use UDP

A TCP/IP socket is uniquely identified by an Internet address, an end-to-end protocol (TCP or UDP), and a port number



Each program that has a reference to a particular socket can communicate through that socket. Earlier we said that a port identifies an application on a host. Actually, a port identifies a socket on a host