

# 【CN】 Day11

🕒 Created	@May 29, 2022 9:33 AM
▼ Class	
▼ Type	
☰ Materials	Network Application Architecture and Process Communicating
☑ Reviewed	<input type="checkbox"/>

## 【Ch2】 Application Layer

### 2.1 Principles of Network Applications

At the core of network application development is writing programs that **run on different end systems** and **communicate with each other** over the network.

#### 2.1.1 Network Application Architectures

The **application architecture** is designed by the application developer and dictates **how the application is structured** over the various end systems.

In choosing the application architecture, an application developer will likely draw on one of the two predominant architectural paradigms: **the client-server architecture** or **the peer-to-peer(P2P) architecture**.

In a **client-server architecture**, there is an always-on host, called **the server**, which **services requests** from many other hosts, called **clients**.

A classic example is the Web application. When a Web server receives a request for an object from a client host, it responds by sending the requested object to the client host. Note that with the client-server architecture, **clients do not directly communicate with each other**.

Another characteristic of the client-server architecture is that the server has a fixed, well-known **IP address**, and because the server is always on, a client can always

contact the server by sending a packet to the server's IP address.

Often in a client-server application, **a single-server host is incapable of keeping up with all the requests from clients.**

For this reason, **a data center**, housing a large number of hosts, is often used to create a powerful virtual server.

In a **P2P architecture**, there is **minimal(or no) reliance on dedicated servers in data centers.** Instead the application exploits direct communication between pairs of intermittently connected hosts, called peers.

### 2.1.2 Processes Communicating

Before building our network application, we also need a basic understanding of how the programs communicate with each other.

It is not actually programs but **processes that communicate.**

Processes on two different end systems communicate with each other by exchanging **messages** across the computer network. A sending process **creates and sends messages** into the network; a receiving process **receives the messages and possibly responds** by sending messages back.

#### *Client and Server Processes*

For each pair of communicating processes, we typically label one of the two processes as **the client** and the other process as **the server.**

With the Web, a browser is a client process and a Web server is a server process.

With P2P file sharing, the peer that is downloading the file is labelled as the client, and the peer that is uploading the file is labelled as the server.

In the context of a communication session between a pair of processes, **the process that initiates the communication is labelled as the client.** The process that **waits to be**

contacted to begin the session is the server.

### *The Interface Between the Process and the Computer Network*

A process sends messages into, and receives messages from, the network through a software interface called a [socket](#).

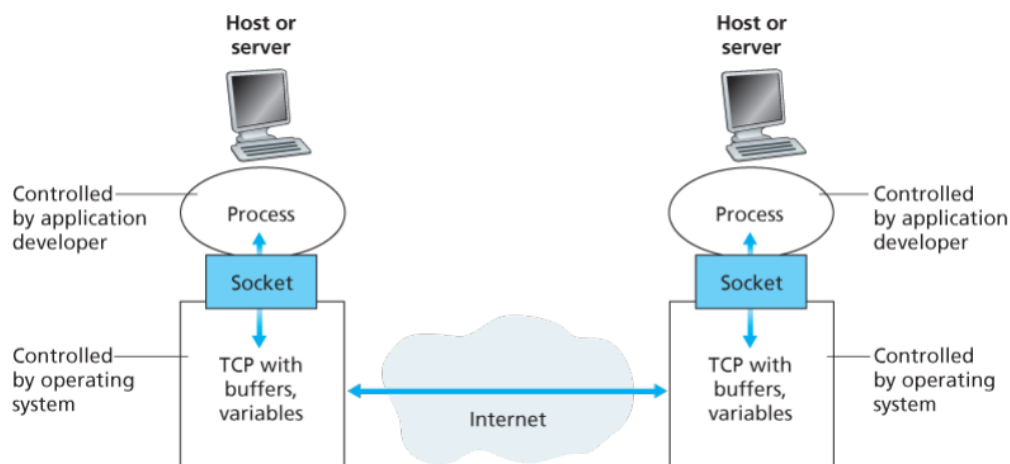


Figure 2.3 Application processes, sockets, and underlying transport protocol

### *Addressing Processes*

In order for a process running on one host to send packets to a process running on another host, the receiving process needs to have an address.

The host is identified by its [IP address](#). A [destination port number](#) identifies the receiving process.