Chapter 6
Computers and Networking

Adam E. Flanders

Contents

A. E. Flanders (✉)
Thomas Jefferson University, Philadelphia, PA, USA
e-mail: Adam.Flanders@jefferson.edu

B. F. Branstetter IV (ed.), *Practical Imaging Informatics*,
https://doi.org/10.1007/978-1-0716-1756-4_6

6.1 Introduction

The core infrastructure of any modern radiology department is made up of computers and the connectivity or *networking* capability between these devices. All transactions between modalities, PACS, scheduling, billing, dictation, and reporting systems are made possible through specialized computer programs or *applications* that are executed by computers. Computer systems are quite diverse and are often designed to augment a specific task, whether it is to support image reconstruction for a modality such as computed tomography (CT) or digital radiography (DR) or rapid image display as in PACS workstations. Fundamentally, all computers are built around a similar base design with enhancements in specific areas to address certain needs such as rapid storage access and data transfer for file servers or improved video characteristics for PACS client display stations. The purpose of this chapter is to familiarize the reader with the fundamentals of computer architecture, networking, and computer applications.

6.2 Computers 101: Hardware

6.2.1 Hardware Elements of Computers

- There are five **core hardware components** of the modern digital computer system: the central processing unit (CPU), memory, input devices, output devices, and a bus. These are all part of a **physical machine** or a single computer.
- While some components are given greater emphasis for a particular computer design (e.g., a faster CPU for computationally intensive tasks), virtually all types of computers have these five key components represented. Most of the hardware components in the modern digital computer are contained within small modular semiconductor packages (**integrated circuits [ICs] or chips**) that, in turn, contain millions of discrete components.
- Numerous ICs are interconnected on a large circuit board, frequently referred to as the **motherboard**. The motherboard is interfaced with other outside compo-

KEY CONCEPT: Software vs. Hardware

The fundamental distinction between software and hardware is that hardware exists as the tangible physical components and connections inside of a computer. Software is a set of instructions that are performed on the hardware.

KEY CONCEPT: Core Computer Hardware Components

- CPU (central processing unit)
- Memory
- Input devices
- Output devices
- Bus

nents (e.g., disk drives, power supply, keyboard, network, etc.) using specialized couplers providing necessary power and connectivity to **peripheral devices** such as disk drives (storage), video displays, and keyboards.

- The **central processing unit (CPU)** or **microprocessor** is typically the largest integrated circuit on the motherboard, and its role is to execute specific commands or instructions/machine code dictated by a computer program and to orchestrate the movement of data and instructions through the entire computer system.
- Although the CPU is frequently personified as the "brain" of the computer, it has no innate "intelligence" or inherent ability to make decisions. The CPU's strength is in its ability to process instructions and manipulate data at amazing speeds. In this regard it is the perfect soldier; it follows all commands presented to it with blazing efficiency.
- The number of instructions that a CPU can perform per second is expressed as its **clock speed**. Typical personal computer CPUs can perform over three billion instructions per second or three gigahertz (3 GHz). Modern CPUs actually contain 2–12 CPUs or **cores** in 1 IC (**multicore CPU**). This provides unparalleled computational speed as each core shares the processing tasks formerly assigned to one CPU.
- While the strength of the CPU is in its ability to process instructions, it has limited capability to store data before or after execution. The CPU relies on **physical memory** to store this information and provide it to the CPU on demand.

6.2.2 GPUs/TPUs

- A graphics processing unit (GPU) is a specially designed microprocessor that handles graphics and display operations that are sent to the video display with the capability for high volume and low precision computing.
- Early computers did not require GPUs because the demand for graphics operations and calculation was small and could be shared by the CPU.
- Today's high-resolution 3D graphics require off-loading the computational burden for graphical display on these specialized microprocessors that are integrated into a separate graphics card specifically designed to manage all graphics operations.
- A side benefit of GPU architecture is that they are intrinsically very powerful; they may contain hundreds of internal computing cores that are capable of extremely efficient number crunching. This made them a very inexpensive means to achieve supercomputing power and handle the demands inherent to the growing field of **machine learning**.
- A tensor processing unit (TPU) is a specially designed integrated circuit for processing related to neural networks and machine learning. It is a proprietary technology developed by Google Inc for their TensorFlow AI software platform.

6.2.3 Memory Types

- **Memory** is a computer component that is principally used to temporarily store data (and results) and applications or programs. In contrast to the CPU, a memory module has no capability to process instructions; instead memory is designed to reliably store large chunks of data and then release this data on command (often at the behest of the CPU).

SYNONYMS:

- Software
- Application
- Program
- Process

- Physical memory can exist in **solid-state** form as an integrated circuit or as **physical media** (spinning magnetic disk or hard disk drive [HDD], compact disc [CD], digital versatile disk [DVD], or solid-state drive [SSD]).
- **Nonvolatile memory** will retain data written to it until it is erased or overwritten. Examples include USB memory sticks and disk drives [HDD and SSD]. Since the inherent speed of nonvolatile memory is substantially slower than that of volatile memory, volatile RAM is typically employed on the motherboard to augment data processing.
- A solid-state memory module that can be erased and rewritten an unlimited number of times is generically referred to as **random-access memory or RAM**.
- **Read-only memory (ROM)** is memory that has pre-stored instructions/data that is nonvolatile (persists without power applied). EPROM or erasable programmable read-only memory is a type of ROM that can be erased and reused. This is often how patches/updates are applied to hardware devices.
- Memory that can only retain data with power applied to it is referred to as **volatile memory** – most RAM motherboard memory modules are of this type. These are rated by their **storage capacity** (given in mega- or gigabytes), **access speed** (in nanoseconds), **data rate** (DDR2), and **configuration** (single in-line memory module or dual in-line memory module SIMM or DIMM).
- Some forms of memory are designed for specific tasks. **Video memory (VRAM)** is employed on video graphics cards to store graphical information to improve video display performance. A specialized form of high-performance memory is found on most CPUs to help efficiently buffer data that moves in and out of the microprocessor core (**L2 cache memory**).

6.2.4 Storage

- There are additional forms of computer memory that are classified simply as **storage**, principally because they are characterized by slower speed compared to solid-state memory and nonvolatile

CHECKLIST: Types of Data Storage

- Online
- Near-line
- Off-line

characteristics (data persists indefinitely until erased/overwritten). These are made up of spinning media (disk drives, CDs, and DVDs) and linear media (tape). Strategy for storage is a balance of cost versus demand for access; data that has a high probability for immediate future use (e.g., recent relevant prior imaging studies) should be kept in an online state.

- **Online storage** refers to high-performance, nonremovable media that requires no human or mechanical intervention to retrieve. Data on spinning hard disk arrays is an example of online storage. As storage costs have decreased, most data is kept in online storage.
- **Near-line storage** consists of removable media (e.g., tapes, CDs, or DVDs) that are made available through mechanical means such as a robotic tape or optical disk jukebox. The efficiency of data retrieval with a near-line system is dependent upon the mechanical speed of the robotic system and the queuing mechanism of the media.
- **Off-line storage** is removable media that requires human intervention to load and retrieve data. As a result, performance is lowest for off-line storage. While off-line storage is the least expensive storage strategy, it is otherwise quite inefficient and is therefore reserved for data that has a low probability for future use.

6.2.5 Input/Output

- **Input/output devices** are hardware extensions that allow humans (or other devices) to interact with a computer. Examples of input devices include the keyboard, touch screen, mouse, microphone, and camera. Typical output devices include the video display, printer, plotter, and speaker.

6.2.6 Data Bus

- The **data bus** is the physical data chain built into the motherboard that allows for efficient data transfer. This is supported by several integrated circuits, known as the **chipset**, which coordinates uninterrupted data transfers through the bus. The chipset performs an essential role as the typical microprocessor can execute several billions of commands per second; it is highly dependent upon an efficient mechanism for delivering instructions and data to it. This requires that there is a well-orchestrated method for moving data between motherboard components and the CPU. Multiple different designs have been developed; the most common in use today is PCI (peripheral component interconnect) and PCI Express. The data bus is defined by a **data width** (typically 32 or 64 bits), which specifies how much data is delivered across the bus per cycle and a **clock speed** (given in megahertz).

6.2.7 BIOS

- Another key component to the typical computer motherboard is the **BIOS (basic input/output system)**. The BIOS is comprised of a non-erasable ROM chip that contains the minimal amount of software necessary to instruct the computer how to access the keyboard, mouse, display, disk drives, and communications ports.
- When the power is first applied to the computer, the motherboard relies on the BIOS to tell it what additional components are available to the motherboard for input and output (e.g., disk drives, memory, keyboard, etc.) The motherboard "becomes aware" of what is available and how to access it, each and every time the computer is restarted.

KEY CONCEPT: Booting Up

The motherboard, CPU, and memory retain no previous information about how the computer is configured. Every time the computer turns on, it pulls itself up by its bootstraps ("booting up").

- The BIOS also provides information to the motherboard on where to find the first piece of software to load during the startup process. The startup process is also known as the **boot process**. The first piece of software to load is usually a portion of the **operating system** that will coordinate the other software programs.

6.2.8 Virtual Machine

- While all of the components are components of a single computer, the outdated notion of a one-for-one relationship between CPU, memory, and storage has evolved into an abstraction known as a **virtual machine**.
- A **virtual machine (VM)** is a software representation of hardware. It is an **image file** of a single computing environment that is all encapsulated in software yet performs exactly like a physical machine. Multiple virtual machines can co-exist and run on a single physical computing environment sharing resources. The user of these **guest** systems has the identical experience as that of using an independent physical device.

KEY CONCEPT: Virtual Machine vs. Physical Machine

Multiple virtual machines can be supported on a single piece of hardware allowing for economies of scale with decreased maintenance, backup, and recovery costs. Each virtual machine acts like a distinct computer, but several virtual machines can exist on a single piece of hardware.

- A VM environment offers tremendous economies of scale, allowing multiple operating systems to run simultaneously on a single physical device. VMs are easier to maintain, manage, back up, and restore and can be cloned at will. Most modern data centers use VMs to support vendor software solutions rather than installing physical devices for each company.

6.3 Computers 101: Software

- Hardware can be seen and handled. **Software**, on the other hand, is a virtual concept. While we can handle the media that software is written on, we cannot actually "see" the software.
- The term "software" applies both to **application programs** and **data**.
- Software at its lowest level (the level at which it interacts with the CPU) consists of a long series of **bits** or binary digit (ones and zeros). All data written to physical media, whether it is magnetic disk, USB stick, CD, DVD, or RAM, is stored as an orderly series of bits. A **byte** of data is eight sequential bits.
- Software is divided into operating **system software**; **application software**, programs which help users perform specific tasks; and **programming or development software**, programs that aid in the writing (i.e., coding) of other software.
- All software consists of individual procedures that command the computer to follow a precisely orchestrated series of instructions. The number of individual instructions specified in any one program varies depending upon the type and complexity of the software – from ten to one hundred million lines of code. (The Windows X operating system, for example, contains approximately 50 million lines of code.) By comparison all of the code hosted by Google approximates two billion lines of code!

FURTHER READING: Core Computer Components

How Computers Work (10th Edition). White R, Downs TE. Que Publishing, Indianapolis, 2014.

- All computer software must be moved into storage (i.e., disk drive) or physical memory (RAM) before it can be **executed** by the microprocessor. The instructions are passed through a series of software layers where they ultimately reach the microprocessor. Executing an instruction causes the computer to perform one or more operations.

6.3.1 Computer Operating System

- The **operating system (OS)** is the underlying software that integrates the hardware with software applications. It is distinguished from the essential hardware components in that it consists entirely of *software* – millions of lines of machine commands that are understood and obeyed by the microprocessor. The OS actually consists of hundreds or thousands of individual programs **(executables and libraries)** that are bundled together. Many of these individual programs are designed to work cooperatively with each other (libraries), whereas single executable files may be run on demand by the user of another **process** on the system.
- The OS is automatically executed each time the computer is started, and it is the *most* important software component running on any computer. A modern computer cannot operate without an operating system.

- Although the CPU is frequently personified as the "brain" of the computer, it is really the OS software and the CPU acting together that provides the underlying "intelligence" of system. The OS and the CPU are inexorably linked; therefore, the distinction between software and hardware is sometimes blurred.
- The OS is designed to automatically manage nearly every task (or **process**) including maintenance of the files on the disk, tracking input from peripheral devices like keyboards or network cards, displaying output on printers and video displays, and controlling **memory allocation**. Memory allocation is crucial to maintaining stability of the system because if two programs try to use the same area of memory, both programs will usually fail.
- Two of the most critical jobs of the OS are ensuring that programs do not unintentionally interfere with each other, adjudicating limited resources (e.g., memory), and maintaining security.
- A function paramount to the modern OS is the support of the **graphical user interface (GUI pronounced "gooey")**. A GUI replaces typed computer commands with a graphical representation of the task (e.g., moving a file). This is accomplished by creating a visual representation of the computer file system (the desktop), icons, and windows and linking them to the movements of a pointing device such as a mouse or trackball.
- The OS also provides a foundation or **software platform** for all other software (application programs). Therefore, the choice of operating system to a large extent determines which application software can be used on a particular system.
- There are several operating systems in use today. The most popular is the Windows OS (Microsoft, Redmond, Washington) which runs on the majority of computers worldwide, especially business computers. Other choices include UNIX, Linux, DOS, and the macOS (Macintosh).
- A **multiprocessing OS** supports use of more than one CPU. A **multitasking OS** allows more than one program to run simultaneously. A **multithreading OS** allows different parts of a program to run concurrently, and a **multiuser OS** allows two or more individuals to run programs concurrently on the same computer system.
- An OS includes hundreds of small programs called **drivers**. Drivers enable software to interact with the ubiquitous hardware devices attached to the motherboard and between components on the motherboard itself. In other instances, drivers allow one software component to safely interact and exchange data with another piece of software.

KEY CONCEPT: Drivers

Drivers are small programs that enable the operating system and application programs to interact with each other and with peripheral hardware devices. They require periodic upgrades, especially when the OS changes.

- From the user perspective, the OS provides the framework in which the application software runs. All application software runs **on top of** the OS; the OS, in turn, exchanges data/instructions directly with the hardware. In general, applica-

tion software cannot interact directly with the hardware; it must be brokered through the OS. The OS essentially adjudicates allocation of resources to meet the demands of the user and applications. The modern OS is intentionally designed to sustain itself automatically with minimal user interaction. The software that is designed to perform real work for users is the application software.

6.3.2 Application Software

- OS software is designed to run autonomously with little interaction from the individual user. The OS monitors all internal functions of the computer, maintains stability of the hardware components, and regulates the processing of data in the microprocessor.
- **Application software** is a program designed to do *real work* for a user. Application software does not supplant the base OS software. Instead, application software runs *on top of* the OS such that an application is written (or coded) to work with a specific OS. The application is coded with OS-specific instructions to request specific actions such as opening a window, writing text, drawing objects, etc.
- Examples of application software include word processors, Internet browsers, PACS viewers, dictation systems, and spreadsheets.

6.3.3 Software Containers

KEY CONCEPT: Virtual Machines vs. Containers

A virtual machine (VM) is an abstraction of the hardware layer of a computer, whereas a container is an abstraction of an application.

- **Containers** are often compared to VMs as both allow multiple kinds/types of software to be run in a contained or isolated environment.
- While VMs are an abstraction of the hardware layer of a computer, a **container** is an abstraction of an **application**. One or more containers can share a single OS.
- **Containerizing** an application involves bundling the application, its configuration files, all needed libraries, and dependencies into an isolated self-sustaining file running inside of an OS.
- The application container allows specific inputs to go in and specific outputs to come out. This provides for a very secure and stable environment.
- Unlike a VM, a container does not include an entire OS to function, only the essential components (i.e., dependencies).
- Two of the most popular container systems are **Docker** and **Kubernetes**.

6.3.4 Programming Languages

6.3.4.1 Low-Level Programming Language

- **Low-level programming language** is the software language that is **directly** understood by a microprocessor and is termed **machine code or machine language**. Every CPU model has its own native machine code or **instruction set**. The instruction set consists of a limited number of relatively primitive tasks, like adding or subtracting data in specialized memory placeholders called registers or moving data from one register to the next.
- Despite their enormous processing speed, the intrinsic mathematical capabilities of a microprocessor are quite limited; a CPU cannot perform simple multiplication or division on its own – it has to be *taught how to do it.* By stringing a series of machine codes together, more complex processing (e.g., multiplication) is possible.
- Both machine code and its symbolic representation (**assembly language**) are considered low-level languages because they are the closest command analog to the actual functional details of the microprocessor. Low level does not imply diminished quality or efficiency; in fact, programs written directly in machine code or assembly language are very efficient.
- Although low-level programming instructions produce efficient programs, programming in machine code or assembler is difficult, tedious, and very time-consuming.

6.3.4.2 High-Level Programming Language

- **High-level programming language** is an abstraction of machine code programming because it uses natural language elements instead of arcane numbers and abbreviations. This makes the process of programming simpler, intuitive, and more understandable to the human programmer.
- High-level programming is the foundation of most software development projects. There are many high-level languages in common use today. Some of the popular languages currently include C, BASIC, Python, JavaScript, Java, Go, R, PHP, and Swift.
- Using high-level programming languages, programmers (or "coders") type out individual lines of the **source code** for an application, using a development software program.

6.3.4.3 Integrated Development Environment

- An **integrated development environment (IDE)** is a toolset that facilitates programming by providing a workspace that makes coding more efficient.

- The lines of the source code need to be translated into **machine code** before the program can be understood and tested on the microprocessor.
- This conversion process is known as **compiling** a program, and the software that converts the source code to machine code is known as a **compiler.**
- Most software development platforms include one or more compilers. The compiler turns the source code into an **executable** program which is customized for the specific OS/microprocessor combination for which the program was developed.
- The compiler saves the programmer a substantial amount of time and effort by constructing the sequence of machine codes that accurately represents each source code command.
- Programmers must follow a tedious sequence of compiling, testing, identifying errors, correcting errors, re-coding, and re-compiling a program in a process known as **debugging** the program. Most of the time devoted to programming is spent debugging the code.
- **Scripting languages** differ from compiled languages in that the source code is **interpreted** and converted into machine code at the time of execution – obviating the compiling process. The development process with scripted languages is typically more rapid than with compiled code since it can be tested while written; however, because scripting languages are interpreted at the time of execution, they are typically slower to execute. Therefore, scripted language is often reserved for smaller programs that are not computationally intensive. Scripting languages include JavaScript, VBScript, Python, and ASPX.

6.4 Computer Networking

- A **computer network** is a group of two or more interconnected computers that are capable of sharing data and resources. Networking allows multiple independent users to share the same resources (i.e., applications and data) and work with this data simultaneously.
- Fast, reliable networks form the backbone of a digital radiology department and allow large quantities of imaging data to be efficiently transported between modalities, archives, and viewing stations.
- Computer networks can be classified on the basis of scale (i.e., size, complexity), scope, topology, architecture, and connection method.
- The most common network is the **local area network (LAN)**. A LAN is characterized by serving computers in a small geographic area such as a home or office.
- A network which is comprised of two or more LANs is termed a **wide area network (WAN)**. Although the term is somewhat ambiguous, it is more commonly used to describe networks with a broad geographic coverage – metropolitan, regional, or national. The largest WAN is the public **Internet**, which is a global system of interconnected computer networks.

- A typical radiology department network would consist of at least one LAN which may be interconnected to a larger WAN (e.g., a hospital or enterprise network).
- Connection of two or more networks (i.e., **internetworking**) changes the scope of network resources available to any computer on the network. An **intranet** is one or more networks that are under control of a single administrative authority. Access to any external or unregulated networks is either not provided or is limited to authorized users.
- An **extranet** is an internally managed network (intranet) that maintains limited connectivity to networks that are neither managed, owned, nor controlled by the same entity. An extranet is typically isolated from the public Internet with security measures such as **firewalls** which regulate connectivity to outside or unmanaged networks. Most hospitals and business organizations configure their internal network in this way.
- Many home networks (wireless or wired) are extranets that consist of a LAN with access provided to the public Internet (WAN) via an **Internet service provider (ISP)**.
- A **virtual private network (VPN)** is a method for including a distant device into an intranet with nearly the same level of security as if it were on the premises.

FURTHER READING: Networking

Computer Networking: A Top-Down Approach (7th Edition). Kurose JF, Ross KW. Addison Wesley, 2017.

6.4.1 Physical (Hardware) Networking Components

- Basic components of a computer network include the network card, cabling, and a point of connection (e.g., hub, repeater, bridge, router, or network switch).
- The **network interface card (NIC)** is the piece of computer hardware that provides the capability for a computer to communicate over a network. Every NIC possesses a unique number, its **medium access control (MAC) address**. This number can be used to help route data to and from other computers.
- The physical connection of the computer to the network is usually accomplished through specialized cabling that contains four pairs of simple copper wires (twisted pair) in a configuration known as category 5 or **Cat5**, or its enhanced version Cat5e. Cat5 cabling frequently terminates in special rectangle plastic connectors that resemble oversized telephone connectors.

DEFINITION: Bandwidth

The maximum amount of data that can be transmitted over a medium, usually measured in bits per second.

- Other forms of physical connection used less often include **fiber-optic cables (optical fiber)** and **wireless** (802.11x). Fiber optic provides greater transmission capacity (**bandwidth**) than Cat5, and wireless affords greater access where physical connections are not readily available.

KEY CONCEPT: Network Devices

There is a one-to-one relationship between computers and network devices. That is, there is only one computer attached to each network cable.

- The term **Ethernet** describes the wiring and signaling schema for the NIC and the cabling between devices on the network.

6.4.2 Network Switches, Hubs, Bridges, and Routers

- The cornerstones of the computer network are **switches**, the devices that connect other devices together on the network. Switches vary in the degree of functionality by which they manage the data traffic that passes through them. The term *switch* is an imprecise term that refers to many types of network devices.
- The simplest and most inexpensive of network switches is the network **hub**. The hub provides a simple and passive method for all computers connected to it to transmit and receive data to each other. Each computer network cable has an individual connection to the hub. The hub creates a shared medium where only one computer can successfully transmit at a time and each computer **(host)** is responsible for the entire communication process.

CHECKLIST: Types of Network Switches

- Hub
- Bridge
- Router

- The hub is a passive device. The hub merely replicates all messages to all hosts connected to it and does not have any capability to route messages to a specific destination. A network hub is the most basic and inefficient means of connectivity. For this reason, simple hubs are rarely used today.
- The network **bridge** improves upon the design of the basic network hub by providing a level of active management of the communication between attached hosts. The bridge is capable of learning the MAC addresses of the connected host computers and will only send data destined for a specific host through the connection associated with a unique MAC address. By routing the data stream to the intended recipient, switching creates a more efficient method for network transmission.
- Since the bridge needs to examine all data sent through it, it creates some processing overhead which slows the data transmission rate. Bridges typically support data transmission rates of 10, 100, and 1000 megabits per second (Mbs).
- The network **router** offers yet another level of technical sophistication over the network bridge. Like the network bridge, a router is capable of examining the

contents of the data passing through it and is able to discern the identity of the sender and the intended recipient. However, instead of relying on the value of the hardware NIC MAC address (which is fixed and not configurable), the router is capable of discerning data based upon a software configurable identifier known as the **Internet Protocol address (IP address)**.

- The IP address is a configurable 32-bit numeric value (e.g., 192.123.456.789) that is used to uniquely identify devices and the networks to which they belong. Using this schema, a host that is accessible globally must have a unique IP address. With 232 possible combinations (IPv4), this provides for 4.3 billion unique addresses. Under IPv6, there are 2128 possible combinations allowing for far greater unique addresses.
- The **hostname** of a computer is a human-readable unique label for a computer on a network (e.g., mycomputer145.nowhereuniversity.edu). Every device on a LAN or WAN or the public Internet has a unique hostname.
- An IP address may be designated as **fixed** (unchangeable) or **dynamic** (modifiable, reusable). Every hostname has a unique IP address associated with it.
- Network routers maintain **network routing tables** that define the topology of a network, the relationship of devices on a network, and how to reach them.
- With the billions of computers in use today connected to the public Internet, there are not enough unique public IP addresses for each computer to have its own unique address.
- A computer that is hidden within a **private network** need not have a globally unique address (it only needs to be unique on the local network). This scheme allows for conservation of unique IP addresses. That is, two internal networks can use the same IP subaddresses, as long as those computers are not exposed to the rest of the Internet.
- A **subnetwork** is a small network of computers that is connected to a larger network through a **router**.
- The typical **broadband network router** used in home networking has additional features such as **DHCP (dynamic host configuration protocol)**, **NAT (network address translation)**, and a network **firewall**. These additional features provide a secure connection between the home LAN and the ISP WAN.
- **DHCP** is used to orchestrate how devices are automatically configured on a network whereby individual devices "negotiate" with the network controller or router to establish a path for communication.
- The router using **NAT** serves as a proxy that allows multiple computers to share a single public Internet IP address. The broadband network router assigns each computer in the home network its own IP address that is *only unique within the home network*.
- It is through this mechanism that a LAN with dozens or hundreds of computers can share a single IP address to the public Internet.
- The network **firewall** is primarily a security device (hardware and software) that filters traffic between the network, adjacent networks (WAN or LAN), and public networks. Multiple firewalls can exist within an organization to protect for unauthorized access.

6.4.3 Network Processes and Protocols

- To communicate effectively, each device must adhere to a specific set of rules for communication called **network protocols**. Networks are usually comprised of a heterogeneous group of devices of different make, model, vintage, and performance. The most ubiquitous network protocol over Ethernet is the **Internet protocol suite (IPS)** or **Transmission Control Protocol/Internet Protocol (TCP/IP)**.
- TCP/IP is a software abstraction of protocols and services necessary for the establishment of communication between two computers on a network. This network abstraction was set down by **the International Organization for Standardization (ISO)** and is referred to as the **ISO network model**. The model describes five to seven **information layers** that link computer software applications to the hardware that must perform the actual transmission and receipt of data.
- The layers in the network ISO model rely upon protocols to regulate how information is passed up through and down the ISO stack.
- The Internet protocol suite defines a number of rules for establishment of communication between computers. In most instances, the connection is a one-to-one relationship. Two computers go through a **negotiation process** prior to making a connection. The negotiations include request and acceptance of an initial connection, the type of connection, the rate of transmission, data packet size, data acknowledgement, as well as when and how to transmit missing data.

6.4.4 Data Packets

- Data transmitted over a network is broken up into multiple small discrete chunks or **packets** before being sent over the network by the NIC. Packet size is variable and is part of the "negotiations" when establishing a network connection with another computer.

KEY CONCEPT: Data Packets

Since each packet is self-contained and auto-routable, different packets from a single message can travel over completely different routes to arrive at the same destination. This offers redundancy to networks.

- Since a network segment can only be used by a single computer at any one instant and the physical parts of the network (i.e., cabling and switches) are shared by many computers, splitting data streams up into smaller parcels in a shared network model improves network efficiency dramatically.
- Switching and assigning resources on a shared network is a complex process – one which needs to occur in microseconds to maintain efficient communication between thousands of devices that are potentially competing for these resources. This is a precisely managed and timed process whereby unique data packets are

embedded into network traffic in shared routes and redirected based upon predetermined routing rules.

- Despite the refined sophistication of the system, there are instances where two or more computers attempt to send data along the same segment simultaneously. This phenomenon is termed a **collision**. Optimum network design mandates minimizing collisions and maximizing **collision detection** to maintain fidelity of data transmission. In these instances, the controller may request the originating site to resend the lost packet(s).
- Additional metadata is automatically married to each data packet based upon protocols specified in IPS and contains information such as the data type, packet number, total number of packets, as well as the IP address of the sender and receiver. This is analogous to placing a letter (packet) in an envelope with delivery information (sender and return address). Data packets with this additional data **wrapper** are referred to as **data frames**.
- Since each frame of transmitted data contains information about where it originated and where it is supposed to go, routers can then examine each packet and forward it through the relevant pathway that is pre-configured to reach the recipient network. *Moreover, since each packet is self-contained and auto-routable, packets from a single message can travel over completely different routes to arrive at the same destination*. Routers instantaneously analyze and balance network traffic and will route packets over segments that are currently under a lighter load.
- At the receiving end, the ISO model also details how to reassemble the individual packets back into the original file.
- Each packet bears both an identifier and sequential number that tell what part of the original file each packet contains. The destination computer uses this information to re-create the original file.
- If packets are lost during the transmission process, TCP/IP also has methods for requesting retransmission of missing or corrupt packets.

6.4.5 Bandwidth and Latency

- **Network bandwidth** is defined as the rate at which information can be transmitted per second (bits/sec) through a network. In the parlance, a wider "pipe" can provide greater bandwidth. This can vary tremendously depending upon the type of physical connection, switches, and medium (i.e., cabling versus fiber versus wireless).

KEY CONCEPT: Theoretical Bandwidth

In general, actual bandwidth is approximately one-half of theoretical values. Other infrastructure factors that can reduce bandwidth include use of a firewall.

- Theoretical bandwidth of Ethernet, for example, varies from 10 to 1000 megabits per second and is generally twice that of actual bandwidth due to infrastructure constraints (e.g., use of a firewall) and network load.
- Another technology, known as asynchronous transfer mode **(ATM)**, can support bandwidths ranging from 155 Mb/sec to 2488 Mb/sec.
- **Latency** is another network parameter that is often used to gauge performance of a connection. It describes the time for a data packet to take a round trip from sender to receiver and back to the sender expressed in milliseconds (ms).
- Under optimal circumstances a network should achieve high bandwidth and low latency.

CHECKLIST: Theoretical Bandwidths

- Wired bandwidths
  - Ethernet 10 Mbps
  - Ethernet 100 Mbps
  - ATM (OC3) 155 Mbps
  - ATM (OC12) 622 Mbps
  - Ethernet 1000 Mbps
  - ATM (OC48) 2488 Mbps
- Wireless bandwidths
  - 802.11b 11 Mbps
  - 802.11g 54 Mbps
  - 802.11n 600 Mbps
  - 802.11ac 600 Mbps
- Cellular bandwidths
  - 3G - HSPA 7.2 Mbps
  - 3G - HSPA+ 21 Mbps
  - 3G - DC-HSPA+ 42 Mbps
  - 4G - LTE 100 Mbps
  - 5G 20,000 Mbps

- The term **broadband** is often used to describe high-speed Internet access with a minimum of 25 Mbps download and 3 Mbps upload speed. This type of network performance is achievable today through a number of mechanisms (e.g., Wi-Fi, cable, satellite, and cellular).
- Current **wireless (Wi-Fi)** bandwidth speeds rival many hardwired network protocols allowing for streaming media to handheld devices without performance degradation.
- It is important to recognize that there can be a substantial difference between the values of a theoretical bandwidth and **actual** achieved bandwidth. While packets of data move at the speed of light, other factors such as quality of cabling and efficiency of network switches and firewall contribute to **network overhead** that can impede actual performance.

6.5 Client-Server Architecture

- The **client-server computing model** is one of interdependency between two or more computers where one computer provides data or services to the other.
- Early networks were used primarily to back up data to a central

DEFINTION: Server-Client

A server is a computer that provides application services or data. A client is a computer or software application that receives those services and data.

location during off-hours. Each user kept complete versions data and the applications used to create that data on their local devices. This model is expensive to deploy and maintain.
- Cost to deploy and maintain applications and data on separate computers in a large organization has become prohibitive.
- Centralized **data center** with redundant servers that provide services to thousands of remote clients has become the standard architecture in most healthcare systems, largely due to the improvements in network speeds. This allows for data and applications to be centrally managed.

6.5.1 Cloud Computing

- Current considerations include a shift to a **cloud computing** infrastructure whereupon an organization leases computing resources from large, redundant, and geographically disparate locations to obtain access to servers, storage, databases, software, and business intelligence resources.
- The primary advantages of cloud computing are **scalability** and low maintenance costs since an organization is not required to support a physical data center.

6.5.1.1 SaaS, PaaS, and IaaS

- The three types of cloud computing services are **software as a service (SaaS)**, **platform as a service (PaaS)**, **and infrastructure as a service (IaaS)**. Each has variable local maintenance and management requirements. SaaS, for example, is characterized by all management maintained by the cloud provider.
- **SaaS** provides access to software that runs on client computers without installing the software locally. Examples include Google apps, Dropbox, DocuSign, etc.
- **PaaS** provides a development platform in the cloud to create applications, databases, and services as a replacement for local development servers in a data center. Examples include Google Cloud Platform (GCP) and Amazon Web Services (AWS).
- **IaaS** consists of infrastructure such as storage, networking, and virtualization services in the Cloud. Examples include Amazon Elastic Cloud Services (EC2).

6.5.2 Web Services

- As technology has continued to evolve, there has been a growing convergence of desktop computing and network computing. In the past, maximizing computing efficiency required application software and data to reside on the client computer.

- A **fat client** (thick or rich client) is a host application that performs the bulk of data processing operations for the user with minimal to no reliance on network resources. It is typically installed on each local computer.
- By leveraging the power of faster network services, real-time transfer of data and application resources to the client desktop computer is afforded. The client makes requests from a dedicated, powerful networked computer (a server) which stands ready to provide application services or data to the client over the network.
- While any computer can be configured to act as a server, most servers have additional hardware capacity to support the increased demands of multiple simultaneous users (i.e., faster multicore CPUs, large memory stores, and larger hard drives).
- This close interrelationship of multiple clients and a server is known as **client-server architecture**. Almost all of the structure of the Internet is based upon the client-server model. This infrastructure supports delivery of web pages over the World Wide Web and email.

KEY CONCEPT: Client-Server Architecture

In its purest form, client-server architecture concentrates on maximizing virtually all of the computational power on the server while minimizing the computational requirements of the client stations. This affords great economies of scale without loss of functionality.

- The most basic client application is the web browser, which interacts directly with the server to render data, images, or advanced visualizations. Any application that is accessed via a web browser over a network that is coded in a browser-supported language (i.e., JavaScript, Java, HTML5, CSS, etc.) is called **web application or web app**.
- A **thin client** (lean or slim client) is an application that relies primarily on the server for processing and focuses principally on conveying input and output between the user and the server.
- The term thin-client application is often misused by industry to refer to any function or application that runs within a web browser – however, this is an incomplete definition. Even if the application is used inside of a web browser, if additional software or browser plug-ins are required or local data processing occurs, the term **hybrid client** is more appropriate. Most PACS client viewing software that runs within a web browser is classified as hybrid client; there is a local installation of viewer software, but there are tight interdependencies with servers to render the images. Similar relationships exist for post-processing platforms.
- Modern PACS systems are designed to leverage this configuration where the majority of the image management is controlled by a powerful central server which responds to multiple simultaneous requests for image data from relatively inexpensive, less-powerful client viewing stations.

DEFINITION: Thin Client

A software application that does not depend upon any local software components and does not perform any processing on the local host.

- Software applications that are designed to operate principally over a network in a client-server configuration are grouped collectively into something known as **web services**. There are established profiles and specifications which define how these services are supposed to interoperate with service providers and service requesters. Web services differ from web applications in that web services need **not** run inside of a browser or be constructed with web elements.

6.5.3 Protocols

- Clients and servers rely on mutually agreed upon **protocols** for the exchange of information. These are set up by international standards bodies such as the Internet Engineering Task Force and the World Wide Web Consortium (W3C).
- In addition to specifying the ubiquitous medical imaging format, the **DICOM** protocols specify how to discover and exchange images between a PACS client and an archive (see **Chap. 12**).
- There are many protocols in use for the exchange of information. The most common is **Hypertext Transfer Protocol (HTTP)** which is the basis for data exchange on the Internet and in web browsers.
- The **Simple Mail Transfer Protocol (SMTP)** is a standard for email clients to negotiate, transmit, and receive email messages.
- **Simple Object Access Protocol (SOAP)** is used to exchange structured information and is often used to negotiate PACS clients with servers.
- **Representational State Transfer (REST)** is a client-server architecture that promotes stateless retrieval of information from a server by a client. **Statelessness** refers to the condition where each data exchange has no dependencies on the prior exchanges of information; they are all independent.
- **Application Programming Interfaces (API)** allow other software to send commands to the application and are often built using REST transactions between the client and server.

6.6 Database Applications

- Many useful web services and web applications provide direct access to databases.
- There are a number of **database models**; however, the **relational model** is used most often. In the relational model, data is abstracted into **tables** with rows and columns. Each row is an individual

DEFINITION: Database

A structured collection of data. Data which is housed in a database is more amenable to analysis and organization. Databases are ubiquitous and are the essential component of nearly every computer application that manages information.

record, and each column is a separate **attribute or field** for each record. One or more tables are linked logically by a common attribute (e.g., an order number, serial number, accession number, etc.).

- Databases also support an **indexing** mechanism which confers greater speed to the system when accessing or updating data. Indexing comes at some cost since it adds some processing overhead to the system.
- The most common programmatic operations on a relational database include reading or selecting records for analysis, adding records, updating records, and deleting records.
- **Structured query language (SQL)** is a database-specific computer language designed to retrieve and manage data in **relational database management systems (RDMS)**. SQL provides a programmatic interface to databases from virtually any development platform.
- Databases are integral to the infrastructure of most business systems including information systems in radiology. Virtually every aspect of radiology services is tied to relational database functions from patient scheduling to transcription.
- For more information on databases, see **Chap. 11**.

PEARLS

- Although the microprocessor is frequently personified as the "brain" of the computer, it has no innate "intelligence" or inherent ability to make decisions. The microprocessor's strength is in its ability to process instructions and manipulate data at amazing speeds.
- All application software runs *on top of* the operating system, and the operating system, in turn, is directly integrated with the hardware. In general, application software cannot interact directly with the hardware; all interactions are brokered by the operating system.
- A computer that is accessible globally must have a unique IP address; however, a computer that is *hidden* within a private network need not have a globally unique address (it only needs to be unique on the local network). This scheme allows for conservation of unique IP addresses.
- A thin client (lean or slim client) is an application that relies primarily on the server for processing and focuses principally on conveying input and output between the user and the server.
- Software applications that are designed to operate principally over a network in a client-server configuration are grouped collectively into the term "web services."

Self-Assessment Questions

1. The core hardware components of a digital computer include everything **except**:

   (a) Microprocessor
   (b) Memory
   (c) Bus
   (d) Keyboard
   (e) Operating system

2. Volatile memory is distinguished from nonvolatile memory by:

   (a) Poorer performance of volatile memory
   (b) Flammability of volatile memory
   (c) Inability of volatile memory to retain data with power loss
   (d) Greater expense of volatile memory
   (e) None of the above

3. Which is **not** true about storage?

   (a) Online storage is readily available.
   (b) Near-line storage requires human intervention.
   (c) Off-line storage is not accessible by robotic devices.
   (d) Data is stored on media such as tape, compact disk, and DVD.
   (e) None of the above.

4. Which statement is true regarding virtual machines (VMs)?

   (a) A VM doesn't really exist.
   (b) A VM is a software representation of computing hardware.
   (c) Using a VM is not the same as working with a real computer.
   (d) A VM is more difficult to control and maintain.
   (e) None of the above.

5. How does a container differ from a virtual machine?

   (a) Containers are more isolated than VMs.
   (b) Containers only allow one user, whereas VMs allow many.
   (c) A container is an abstraction of a software layer, whereas a VM is an abstraction of the hardware components.
   (d) Containers are more stable than VMs.
   (e) None of the above.

6. Which is the best statement regarding the motherboard data bus?

   (a) It connects to the keyboard.
   (b) It connects to the power supply.
   (c) It interconnects the components on the motherboard.
   (d) It connects to the disk drive.
   (e) None of the above.

7. What is the fundamental distinction between software and hardware?

   (a) Price.
   (b) Hardware is a physical entity.
   (c) Packaging.
   (d) Complexity.
   (e) None of the above.

8. The purpose of the operating system (OS) is:

   (a) To manage memory allocations
   (b) To copy files to disk
   (c) To manage the user interface
   (d) To manage computer resources
   (e) All of the above

9. Computer drivers are:

   (a) Names for a specific type of golf club
   (b) Large programs that take control of the operating system
   (c) Small programs that provide a bridge or interface between hardware and software
   (d) Similar to computer viruses
   (e) None of the above

10. Low-level programming languages are:

    (a) Fairly simple to learn and use
    (b) Are primarily used by human computer programmers to create applications
    (c) Are not as costly as high-level programming languages
    (d) Are used primarily by the CPU
    (e) All of the above

11. The most complex network switch is the:

    (a) Network hub.
    (b) Network router.
    (c) Network bridge.
    (d) They are all similar in complexity.
    (e) Not listed.

12. What statement is true regarding networking addresses (IP)?

    (a) All IP addresses are a fixed unique number.
    (b) We are running out of unique IP addresses.
    (c) IP addresses can by dynamic or fixed in value.
    (d) The computer MAC address is the same as the IP address.
    (e) IP addresses are recognizable names and places.

13. Which is true of thin-client applications?

    (a) They require a third-party web browser to run.
    (b) They do not need software.
    (c) They require a networked server.
    (d) They require an internal database.
    (e) All of the above.