# Networks

A network is a system of computers that can communicate with each other. A small network, such as the computers in a home, or a business, is called a Local Area Network (LAN), a network made up of many LANs, covering a large area, is called a Wide Area Network (WAN). The largest WAN is the internet, which connects computers all over the world.

## Internet

The internet consists of *architecture* — the physical devices that make up this giant communication system — and *protocols* — a set of rules for using the architecture to communicate. The two main protocols for the internet are the Transmission Control Protocol (TCP) and the Internet Protocol (IP).

Any computer can join the internet if it connects to the architecture and follows the protocols. Each computer on the internet is sometimes called a node, this includes personal computers, mobile devices, "smart" devices, etc, as well as computers that are part of how the internet works.

Some computers on the internet are *servers*, which provide some service, such as a web page or access to email, used by other computers. When a computer is getting a service from some server, it is called a *client*.

One important element of the architecture of the internet is *redundancy*, which means that there is more than one path information can follow to get from one computer to another. This

means that if one path is damaged or slow, information can take one of the other paths, making the system more reliable

## IP Addresses

Since every computer on the internet must be able to communicate with every other computer, we must have a unique way to identify every computer. This is done by assigning each one a unique number called an *IP address*.

The rules for these addresses are set out in the Internet Protocol. IP version 4 provided shorter addresses (32 bits long) such as 192.0.2.0 or 198.51.100.0 but these are in the process of being replaced with longer (128 bit) IP version 6 addresses such as 2001:0db8:85a3:0000:0000:8a2e:0370:7334. This provides many more addresses, so that many more devices can be on the internet.

Since more than one program on a computer may be communicating over the internet at the same time, we need a way both to address a specific computer and a specific program on the computer. For this, we add *port numbers* to IP addresses, using a colon to separate them. This combination is called a *socket address*. For instance, port 80 is used for web communication, while port 110 is used for email, so a computer at IP address 198.51.100.0 could have web pages sent to socket address 198.51.100.0:80 and emails sent to 198.51.100.0:110.

IP addresses are provided to a computer when it connects to the internet, usually by an Internet Service Provider (ISP), an organization that provides such connections. An address can be assigned *statically*, in which case the ISP guarantees that you will always be given the same address[1], or *dynamically*, in which case

[1] and generally charges somewhat more money

you will simply be given one of the addresses they own, not necessarily the same one as when you last connected. A computer called the *DHCP server* provides the IP addresses by following a set of rules called the Dynamic Host Configuration Protocol.

For personal computers on the internet, almost everything sent to their address is sent immediately in response to a request. For instance, when you want to view a web page, your computer sends a request (including your current IP address) to the server that has the page, and the page is sent back (to that address) usually within a second. So it does not matter if the address of a personal computer is dynamic; it is unlikely to change in the middle of a communication.

A static address would be important if other computers frequently needed to be able to contact your computer, for instance, if it were a server.

# The Internet Model

The Internet Model[2], also known as the TCP/IP model, describes how information travels over the internet by breaking the process into four layers. At each layer we consider different parts of the process of communication, rather than trying to keep track of everything at once.

- The Application Layer covers how a program sends a message to another program.
- The Transport Layer covers how a computer sends a message to another computer.

[2] The Open Systems Interconnection Model describes a way to break up communication into seven layers to make network communication well organized, but we will stick with the simpler Internet Model.

- The Internet Layer covers how pieces of a message are moved across the internet.
- The Link Layer covers how nodes are actually connected so they can communicate, and is outside what we need to cover.

Separating into layers like this means that we don't have to build information about how the internet works into our programs, and we don't have to build specifics of one program's messages into the internet.

## The Application Layer

In the application layer, a program on the sending computer prepares a message for a program on the receiving computer to use. To be certain that the message will be understood, programs create sets of rules for how the message is formatted, their own protocols

So the sending program creates a message according to its protocol, and the receiving program gets the message, and uses it, based on the protocol.

Neither program knows anything about what happens to the message in between.

## The Transport Layer

In the transport layer a computer is sending a message from one of its programs to a program on another computer.

To do this, it breaks the message into small pieces called *packets*, which can be sent individually. The receiving computer eventually reassembles them into the original message.

Each packet includes part of the data being sent, and a header with information such as the sending socket address, receiving socket address, and which packet it is out of how many in total.

The packets may arrive out of order, and some might not arrive at all. If a packet is damaged (the data has been messed up) or missing (after some amount of time), the packet will be re-sent. This is part of TCP.

Once all the packets have arrived and are reassembled into the original message, the message is given to the program the packets were addressed to, based on port number.

This layer ignores the contents of the message (that's the Application Layer's problem) and how these packets will actually move across the internet (that's the Internet Layer).

### UDP

Suppose that we are doing something like video chat. Do we want to wait for all the parts of each image in the video before we show the next? Probably not, we want the video to keep going so we can keep talking, even if one frame doesn't look as good. For purposes like this, we might use a different protocol. Instead of TCP, UDP, the Universal Datagram Protocol, is a protocol that still uses packets, but focuses on sending and receiving as fast as possible, rather than checking for missing or damaged packets.

## The Internet Layer

In the internet layer, the packets of a message move across the internet.

Routers[3] are computers on the internet which are connected to several other computers; their job is to look at the receiving

[3] for our purposes, we don't distinguish between routers, switches, and hubs, which do similar jobs

socket address each packet and decide, based on that address, which direction to send the packet.

Since the internet has a lot of redundancy, there are many places where more than one direction could be chosen and still move in the right general direction.  So, routers look not only at the IP address the packet is being sent to but also the current traffic (all the other packets currently being sent) in each direction.  They try to balance this traffic so that one direction isn't dealing with more packets (and thus going slower) than it has to.

### Packet Switching

We break our messages into packets and then have routers choose which way to send each packet of a message independently.  This is called packet switching.  Because we do packet switching, potentially each packet in a message might take a different route to reach the destination.

Packet switching is more reliable and potentially faster than trying to send the whole message together.  If the message were sent whole, instead of packet switching, then any problem affecting part of the message would affect all of it.

For instance, suppose that one path between the sending computer and the receiving computer is getting a lot of traffic, so packets on that path are slowed down; other packets from the same message may be sent by the routers along other paths, even if these are longer, so that most of the message still gets there fast.

Suppose that one path goes out completely — e.g. a router loses power before it can send the packet on.  As long as at least one packet takes a path that gets it to the destination computer, TCP would mean we eventually send requests to resend all the others, and the whole message would eventually get through.

## The Link Layer

This layer covers the details of actual connections between machines in a network, for instance how one computer can actually cause a packet to move over a cable to the next computer that needs to see it[4]. We don't need to worry about the details of the link layer for this course.

## Internet Communication Process

Let's go through the process in chronological order (disregarding the link layer).

[Application Layer] The sending program on the sending computer does any necessary formatting using its own protocols to create the message to send to the program on the other computer.

[Transport Layer] The message is broken into packets on the sending computer. Each contains part of the data, and a header including socket addresses and how many packets there are.

[Internet Layer]. The packets are moved over the internet by routers choosing which way to send each packet based on both destination IP and traffic.

[Transport Layer] On the destination computer, the packets are checked and any lost or damaged packets requested to be resent (assuming TCP). The packets are reassembled into the original message and given to the destination program (based on port number).

[Application Layer] The destination program uses the data, based on its own protocols.

---

[4] Some versions of this model include physical details in this layer, others include an extra Physical Layer.

# Internet Architecture

The internet covers the planet, so it must be able to send information very long distances very fast.  For this we mostly use fiber-optic cables, which use light to send information.  These high-speed fiber optic lines are called *backbones*, and the system of these lines working together is also called the *internet backbone*.

Various governments and companies provide parts of the backbone, connected with lots of redundancy.  It is in everyone's interest that information freely move from one system to another.

An Internet Service Provider is a company that is connected to the backbone system and provides internet access to others.  The physical place where an ISP has its computers is called their *Point of Presence* (POP).  An ISP would usually have many POPs to provide access to customers all over a large area, and connect these in an *ISP Regional Network* which should have redundancy as well.

Companies, schools, and other organizations often maintain LANs, on which they might have computers that provide email — mail servers — computers that provide web pages — web servers — and individual computers.

# Net Neutrality

The internet architecture has historically always treated all packets equally, no matter where they are coming from or going

to, because it is more efficient for routers to only check what they need to: the destination of the packet and the current traffic. This is called Net Neutrality.

Since everyone's data moved at the same speed over the internet, Net Neutrality has meant that small new companies could expect their data to be treated the same as that of large established media conglomerates, and controversial material could be widely distributed.

Some companies who own parts of the backbone, however, would prefer to have to option to slow down or speed up certain packets, and would argue that since they own the routers, they should be able to choose how the packets passing through them are handled.

Where net neutrality is not the law, companies who control parts of the internet architecture can, for example, charge online entertainment companies extra for full speed[5], and smaller companies who cannot afford to pay this premium could find their data so slowed down they would lose customers. These controlling companies can also choose to slow down or stop entirely data associated with a rival company, or data from sites whose politics they disagree with, or data using a protocol (e.g. bittorrent) they disapprove of.

[5] Costs that would invariably be offset by increases in subscriber costs.