Internet Addresses

- IP addresses (internet protocol) take the form **nnn.nnn.nnn.nnn** where nnn is a number between 0 - 255
- if you connect to the internet through an internet service provider (ISP) you are generally assigned a temporary IP address for the duration of your dial in session
- if you connect from a local area network (LAN) you have a permanent address or might obtain a temporary one from a DHCP (Dynamic Host Configuration Protocol) server

Protocol Stack

- the protocol stack is built into the OS and used to communicate on the internet
    - the protocol stack is referred to as the TCP/IP protocol stack

Protocol Layer	Comments
Application Protocols Layer	Protocols specific to applications such as WWW, e-mail, FTP, etc.
Transmission Control Protocol Layer	TCP directs packets to a specific application on a computer using a port number.
Internet Protocol Layer	IP directs packets to a specific computer using an IP address.
Hardware Layer	Converts binary packet data to network signals and back. (E.g. ethernet network card, modem for phone lines, etc.)

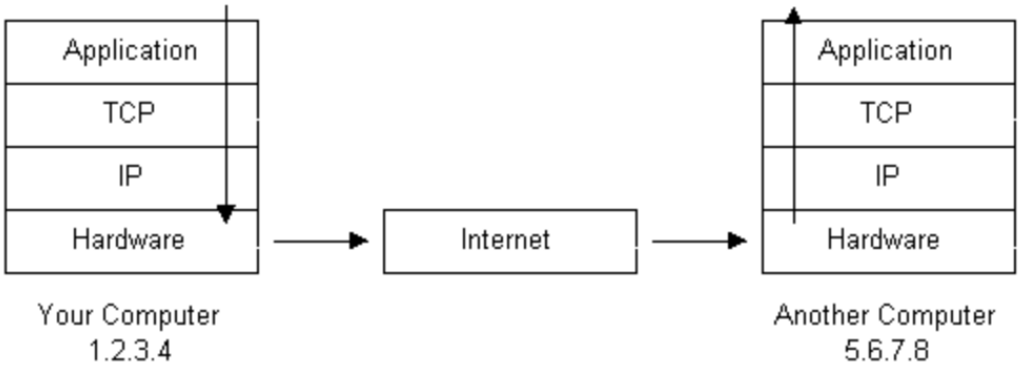
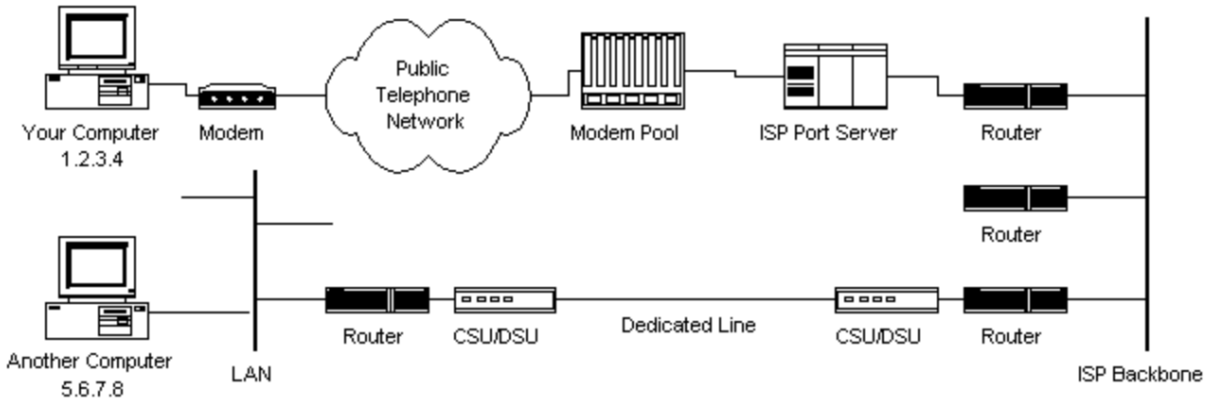


Diagram 2

- data transferred across the internet is broken up into smaller chunks called packets
- in the TCP level each packet is assigned a port number
- in the IP layer each packet receives its destination IP address
    - once packets have a port and IP destination they are ready to be sent via hardware
- the hardware layer turns data / text into binary electronic signals (1s and 0s) and transmits these signals over radio waves / phone line / fiber optic / etc
- on the other end your ISP has a direct connection to the internet

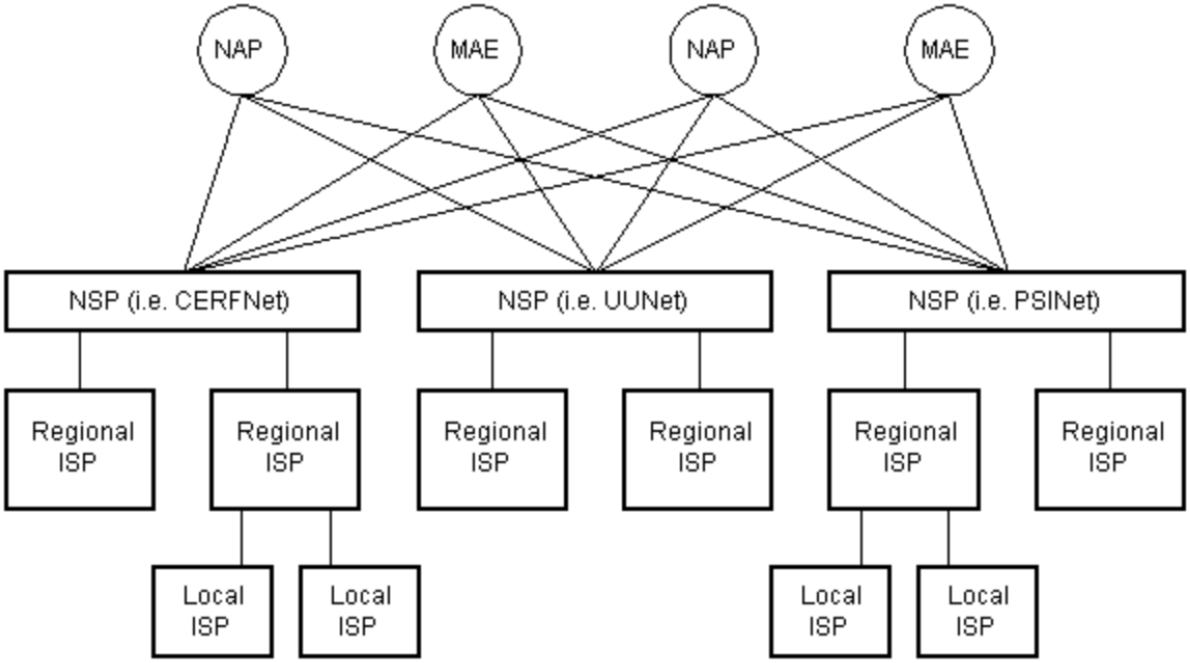
    - the ISPs router examines the destination address of each packet and sends packets with your IP address to you
    - packets normally go through multiple routers to get to you
- packets that you receive start at the bottom of the stack
    - they are stripped of IP and port addresses and then delivered to your application in a manner that is readable
    - they are then re-assembled into their original form



- the ISP maintains a pool of modems for customers which is managed by a dedicated computer
    - this computer controls data flow from the modem pool to a backbone or dedicated router
- after passing through your ISP's local equipment, packets are routed to the ISP backbone or a backbone that the ISP buys bandwidth from
    - from here packets go through several routers, backbones, and networks until they reach their final IP address destination
- `traceroute www.yahoo.com` or `traceroute "IP address"`
    - prints the route of routers, computers, and entities your packets must pass through to get to their destination

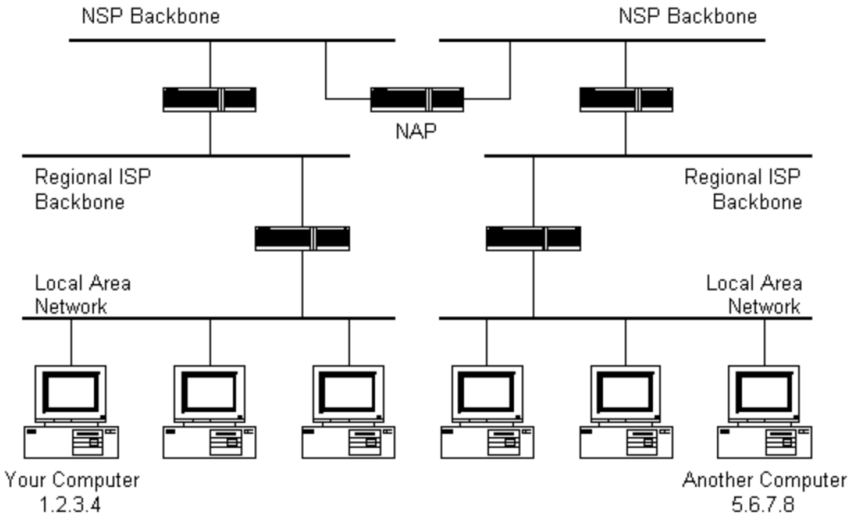
Internet Infrastructure

- the internet backbone is made up of many large networks which interconnect
- these large networks are called network service providers (NSPs)
    - these networks peer to peer with each other to exchange packet traffic
    - NSPs connect to network access points (NAPs) where packet traffic may jump from one NSPs backbone to another NSPs backbone
- NSPs also interconnect at metropolitan area exchanges of MAEs
    - MAEs serve the same purpose as NAPs but are privately owned
- NSPs also sell bandwidth to smaller networks such as ISPs



Internet Routing Hierarchy

- routers are packet switchers, they take in packets sent to them and point the packets in the correct direction
- routers generally know which IP addresses are below it, but not which IP addresses are above it
- when a packet arrives at a router, the router examines the IP address put there by the IP protocol layer on the original computer
    - the router checks its routing table, if the network containing the IP address is found the packet is sent to that network



    - if the IP address is not found then the router sends the packet on a default route, usually up the backbone hierarchy to the next router
    - if the IP address can't be found the packets eventually reach the NSP backbone which has the largest routing tables
- web servers store web page content (images, videos, etc), host applications and databases, and respond to DNS queries

Domain Names and Address Resolution

- domain names service (DNS) is a distributed database the keeps track of computer's names and their corresponding IP addresses
    - many computers connected the internet host part of the DNS database and the software that allows others to access it
    - these computers are known as DNS servers
    - no DNS server contains the entire database, only a subset of it
    - if a DNS server does not contain the domain name requested by another computer, the DNS server redirects the request to another DNS server
- DNS lets you know where on the internet the computer that houses www.anothercomputer.com lives
    - when you enter a web address into your web browser, the browser first connects to your primary DNS server
    - after obtaining the IP address for the domain name you entered, the browser then connects to the target computer and requests the web page you wanted.

HTTP and World Wide Web

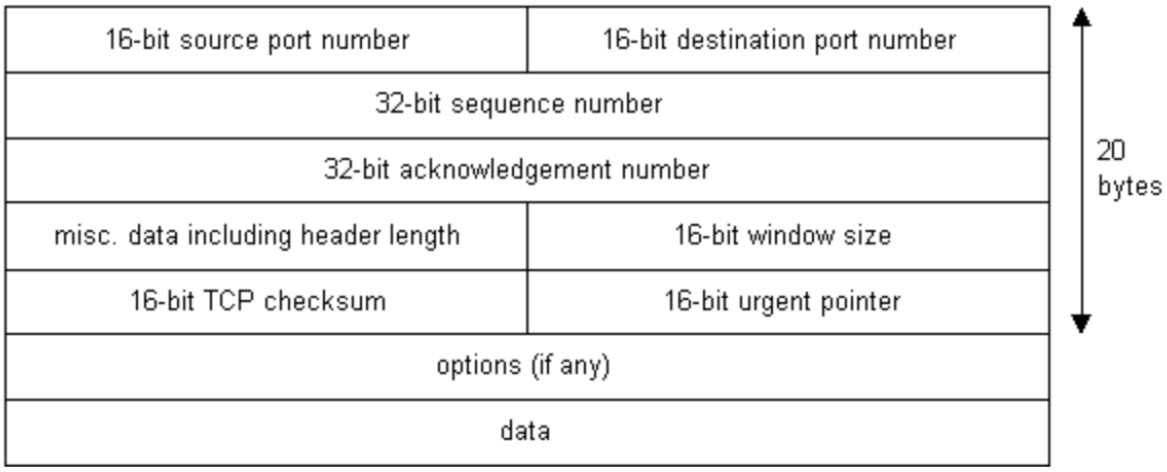
- world wide web (www) is one of the most commonly used services on the internet
    - whereas internet is the network of connected computers, world wide web is a global connection of documents and resources linked by hyperlinks and URLs that are access via HTTP and HTTPs protocols
- HTTP is the protocol that web browsers and web servers use to communicate with each other over the internet
    - used by web browsers and web servers to communicate with each other
    - connectionless text based protocol
    - clients (web browsers) send requests to web servers for web pages and images
    - after a request is serviced, the connection between client and server is disconnected and a new connection must be made for each subsequent request
- most protocols are connection oriented, when you communicate with another computer you keep the connection open. HTML does not do this

- The following example demonstrates the functioning of a web browser when accessing a page at the URL http://example.org/home.html. The browser resolves the server name of the URL (example.org) into an Internet Protocol address using the globally distributed Domain Name System (DNS). This lookup returns an IP address such as 203.0.113.4 or 2001:db8:2e::7334. The browser then requests the resource by sending an HTTP request across the Internet to the computer at that address. It requests service from a specific TCP port number that is well known for the HTTP service, so that the receiving host can distinguish an HTTP request from other network protocols it may be servicing. HTTP normally uses port number 80 and for HTTPS it normally uses port number 443

Transmission Control Protocol

- TCP is responsible for routing application protocols to the correct application on the destination computer
    - ex you want packets from an html going to your internet application while packets pertaining to an email go to your email application
- this is done via port numbers
- when sending packets, the TCP breaks data into chunks (packets) then adds a TCP header with specific TCP information to each chunk
    - maximum packet size is 65,535 bytes
    - includes the port number of the application the data needs to be sent to
- when reading packets the TCP layer receives the packets from the IP layer below it
    - the TCP strips the header from the data, does data reconstruction if necessary, and then sends the data to the correct application using the port number
- for each package received, the TCP sends an acknowledgment to the sender to confirm delivery
- TCP also checks for errors in receiving the data

Below, a TCP header:



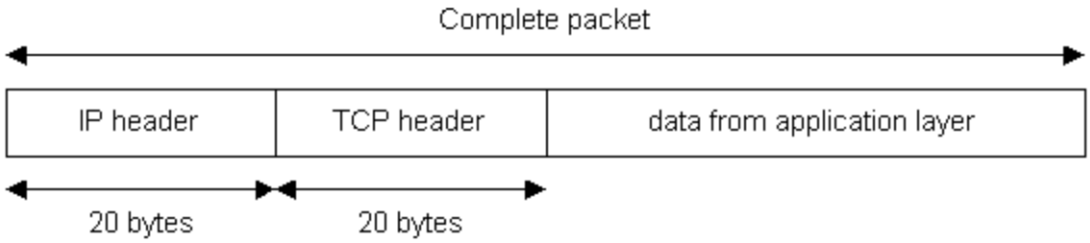
- TCP's job is to get application level data from application to application reliably. The task of getting data from computer to computer is the job of IP.

Below is an example of specific TCP port numbers:

Listed below are the port numbers for some of the more commonly used Internet services.

FTP	20/21
Telnet	23
SMTP	25
HTTP	80
Quake III Arena	27960

Below, a complete packet:



Transport Layer Security (TLS)

- used to encrypt data being passed between IP addresses
- IP addresses initiate a TLS connection and establish a cypher suite of algorithms that specifies details such as which shared encryption keys, or session keys, will be used
    - public key cryptography is used
- the TLS connection also verifies authentication by making the server prove its identity to the client with public keys
    - public keys are encryption keys that use one-way encryption, meaning that anyone can unscramble data encrypted with the private key to ensure its authenticity, but only the original sender can encrypt data with the private key
- once data is encrypted and authenticated, it is then signed with a message authentication code (MAC). The recipient can then verify the MAC to ensure the integrity of the data
- to keep TLS from slowing down transfer too much, TLS false start allows the server and client to begin transmitting data before the TLS handshake has been completed
- TLS session resumption also allows servers and clients that have previously communicated to use an abbreviated handshake
- HTTPS is an implementation of the TLS protocol on top of HTTP

what happens when you load a webpage

- DNS query: When your browser starts to load a webpage, it likely first makes a DNS query to find out the IP address of the website you are searching for.
- TCP handshake: Your browser opens a connection with that IP address.
- TLS handshake: Your browser also sets up encryption between the web server and your device so that attackers cannot read the data packets that travel between those two endpoints.
- HTTP request: Your browser requests the content that appears on the webpage.
- HTTP response: The server transmits the content in the form of HTML, CSS, and JavaScript code, broken up into a series of data packets. Once your device receives the packets and has verified that it has received all of them, your browser interprets the HTML, CSS, and JavaScript code contained in the packets and renders the web page. The whole process takes only a second or two.

IPv4 v IPv6

- the original internet protocol (IPv4) is running out of IP addresses
- IPv6 addresses this by using a 128 bit IP address vs the 32 bit addresses used in IPv4
    - IPv4 can support only 2^32 (4.29 B) unique addresses which have all been allocated
- IPv6 achieves greater scale by using a hexadecimal address format
    - instead of using a base 10 number system with numbers 0-9 a hexadecimal system uses numbers 0-9 to represent 0 to 9 and A-F to represent 10-15, making it a base 16 number system
- IPv4 example: 192.159.252.76
- IPv6 example: 3FFE:F200:0234:AB00:0123:4567:8901:ABCD
- IPv4 records are in DNS address A while IPv6 records are in DNS address AAAA
- IPv4 minimum packet size is 576 bytes, IPv6 minimum packet size is 1208 bytes

Private IP addresses

- not routed on the internet and traffic cannot be sent to them
- intended for closed local area networks where computers and the closed network communicate with each other

DNS Records

- **A** record (address record) points a domain or a subdomain to an IP address
- **CNAME** (canonical name) point one domain or subdomain to another domain, ex: points www.example.com to example.com
- **AAAA** record, similar to A record but points to an IPv6 address

- **MX** (mail exchange) directs email to a particular server, like CNAME it must point to a domain and not to an IP address
- **NS** (name server) points to the name server where your IP address is stored

TCP Ports

- a port number is always associated with an IP address
- port numbers are identified by a 16-bit number (thus ranging 0-65535) and are reserved for specific services so that an arriving packet can easily be forwarded to a running application
- common port numbers:
    - 22 SSH (secure shell login)
    - 23 Telnet remote login service
    - 53 Domain Name Services (DNS)
    - 80 HTTP Hypertext Transfer Protocol
    - 443 HTTP Secure (HTTPS) over TLS/SSL

SSL / TLS / HTTPS

- SSL (Secure Sockets Layer) and its successor, TLS (Transport Layer Security), are protocols for establishing authenticated and encrypted links between networked computers
- although SSL protocol was deprecated with the release of TLS 1.0 in 1999, it is still common to refer to these related technologies as SSL or SSL/TLS. The most current version is TLS 1.3
- SSL/TLS works by binding the identities of entities such as websites to cryptographic key pairs, each key pair consists of a private key and a public key. The private key is kept secure, and the public key can be widely distributed via a certificate
- to receive SSL/TLS website certification HTTPS:
    - generate a pair of public and private keys, preferably on the server that is to be protected
    - the public key and domain name are used to generate a certificate signing request (CSR)
    - the CSR is sent to a publicly trusted CA (certificate authority) such as SSL.com, the CA validates the information in the CSR and generates a certificate to be installed on the requestor's web server
- with an SSL/TLS certificate traffic can be directed through port 443 and a website is HTTPS secured
- users visiting an HTTPS website are assured of the following:
    - authenticity, the server presenting the certificate is in possession of the private key that matches the public key in the certificate

    - integrity, documents signed by the certificate (e.g. web pages) have not been altered in transit by a man in the middle
    - encryption, communication between the client and server is encrypted

localhost

- localhost is a hostname that refers to the current computer being used to access it
- it is used to access network services running on the host (access whatever the computer is displaying to the internet)
    - it uses the loopback network interface to do this and creates a website at the URL http://localhost that can only be accessed by the computer displaying the website
- the localhost DNS points to the IPv4 address: `127.0.0.1`
    - it also points to the IPv6 address: `::1`
- IPv4 network standards reserve the entire address block 127.0.0.0/8 (more than 16 million addresses) for loopback purposes
    - any packet sent to any of those addresses is looped back
- example: `http://localhost:8080/` means that you are explicitly targeting port 8080 on the localhost