**Lecture 6: IP Addressing**

**6.1 Network Layer**

largely involves routing of datagrams from source to destination which involves the IP protocol and routing protocol

* transport segment from sending to receiving host
  + sending side encapsulates segments into datagrams
  + receiving side delivers segments to transport layer
* network protocols exist in every host and router
* intermediate routers examine header fields in all IP datagram passing through it to see the IP address so that it knows where to send to

**6.2 Key Network-layer functions**

2 main functions:

1. Forwarding: move packets from router’s input to appropriate router output (which router to go next?)
2. Diagram

   Description automatically generatedRouting: determine route taken by packets from source to destination (what is the end-to-end routing path from sender to receiver?)

\* Note that routing protocol must be ran before forwarding to obtain the forwarding tables

**6.3 Data Plane, Control Plane**

6.3.1 Data planes

* local, per-router function that takes care of forwarding from router to router
* determines how datagram arriving on router input port is forwarded to router output port based on forwarding table
* typically operates in small time scales (milliseconds)

6.3.2 Control Planes

* network-wide logic that determines end-to-end routing path
* determines how datagram is routed among routers along end-to-end path from source to host
* 2 common approaches:

1. traditional routing algorithms: data plane and control planes both implemented in routers
2. software-defined networking (SDN): implemented in remote servers, provides a zoomed-out overview of the network, don’t have to rely on router-to-router communications

**Text

Description automatically generated6.4 Overview of network layer**

**6.5 IP Addressing**

* IP addresses: 32-bit unique identifier for host or router interface
* Interface: connection between host/router and the physical link
  + routers typically have multiple interfaces
  + host typically have 2 interfaces (1 for wired ethernet and 1 for wireless)
* IP address is associated with each interface

6.5.1 How are interfaces connected

* Wired ethernet interfaces are connected by Ethernet switches (switches do not have interfaces)
* Wireless WiFi interfaces are connected by WiFi base station

**6.6 Subnets**

* network formed by a group of “directly” interconnected hosts
  + hosts within same subnet can physically reach each other without intervening routers (within subnet don’t require IP protocol)
  + for hosts to send data to host from another subnet, must pass through router
  + Diagram

    Description automatically generatedhosts in the same subnet have the same network prefix (Note: same prefix same subnet)

6.6.1 CIDR

* IP address comprises of 2 parts:
  + Network (subnet) prefix (Most significant bits)
  + Host ID: used to uniquely identify the host (least significant 32 – bits)
  + Diagram, timeline

    Description automatically generated# of interfaces we can hold is
* CIDR: Classless Inter-Domain Routing
  + arbitrary length for the network prefix
  + A picture containing text

    Description automatically generatedaddress format: a.b.c.d/x, where x is # of bits in subnet portion of address

6.6.2 Subnet mask

* made by setting all network prefix bits to “1”s and host ID bits to “0”s
* Graphical user interface

  Description automatically generated with low confidenceused to determine which network and IP address belongs to (using bitwise AND operation)

**6.7 How to get IP address**

6.7.1 How do ISPs obtain block of IP address

ICANN: Internet Corporation for Assigned Names and Numbers is an organization that allocates addresses, manages DNS, assigns domain names and resolves disputes. ISPs registers a block of addresses from ICANN and distributes it to their clients.

Graphical user interface, text, application, email

Description automatically generated6.7.2 Special/Reserved IP Addresses

Private addresses are used within a local network (e.g. within a company/university, local area network) and are not used to communicate with devices outside of the local network

6.7.3 How do Organizations obtain a block of IP addresses?

Table

Description automatically generatedBigger organizations like Google or Meta may buy IP addresses directly from registry. Smaller organizations who are customers of ISPs will rent from ISP’s address space.

6.7.4 Hierarchical Addressing: Route Aggregation

Diagram

Description automatically generatedAllows for efficient advertisement of routing information

Diagram

Description automatically generatedSuppose that an organization wants to switch ISP, we don’t have to renumber all router and hosts

6.7.5 Longest Prefix Matching

Steps to find out which router to route to:

1. Convert dotted representation of IP address to binary
2. Timeline

   Description automatically generated with medium confidenceSee which destination address range matches with the prefix of the given IP address

6.7.6 How do end hosts get IP address?

* hard-coded by system admin in a file
* DHCP (Dynamic Host Configuration Protocol): dynamically get address for server, “plug-and-play”

**6.8 DHCP**

**Goal**: allow host to dynamically obtain IP address from network server (follows a client-server model)

* can “borrow” and renew lease on address from network
* allows for reuse of address (only hold address while connected)
* support for mobile users who want to join network

**Overview**:

* host broadcasts “DHCP discover” msg [optional]
* server responds with “DHCP offer” msg [optional]
* host requests IP address: “DHCP request” msg
* server sends address: “DHCP ack” msg

\* Note: 1st 2 steps can be skipped if the hosts already know the IP address of the server (e.g. when it renews lease)

**DHCP returns**:

* address of first-hop router
* name and IP address of DNS server
* network mask

**DHCP runs over UDP** (Application layer)

* DHCP server port number: 67
* DHCP client port number: 68

**A picture containing text

Description automatically generated**If DHCP only exists on 1 subnet, the intermediate router between multiple subnets must be a relay agent to forward DHCP requests to subnet with DHCP server. Router is known as a relay agent.

**Lecture 7: Network Layer**

Table

Description automatically generatedNetwork layer delivers packets to receiving hosts. In this layer we are only concerned with the IP header fields of the IP datagrams passing through it.

**7.1 IPv4 Datagram Format**

Fields (total of 20 bytes):

* Version number: IPv4 or IPv6
* IP datagram length: length of IP datagram including IP header
* Flags: 3-bit value
* Fragment offset: 13-bit value (reason why we express offset in units of 8 bytes)
* TTL: used to prevent packet from circulating in the network forever, once TTL is 0 packet is dropped
* Header checksum: checksum for IP header (only)

**7.2 IP Fragmentation, Reassembly**

* Different links have different MTU (Max Transfer Unit) – the maximum amount of data a link-level frame can carry
* Datagrams that are “too large” needs to be fragmented by routers
* Reassembly of packets is done at the destination host and will only push it up the network layer once all it has received ALL the fragments
* IP header fields of each fragment are mostly the same except for the part responsible for fragmentation/assembly

7.2.1 Fragmentation/Assembly bits in IP header

Flag (frag flag) set to:

* 1 if there is next fragment from same segment
* 0 if it is the last fragment

Offset: starting byte of payload, expressed in units of 8 bytes. If offset is not divisible by 8 then we must add padding (\0 bytes)

Diagram

Description automatically generatedID: 3-bit field that helps to identify whether different fragments belong to the same datagram

\* Note that if any fragment is lost, it would mean that the entire datagram is lost requires upper layer for recovery mechanisms

**7.3 NAT (Network Address Translation)**

Internet has 2 types of IP addresses:

* Public IP addresses
  + globally unique and routable
* Private IP addresses (reserved IP ranges)
  + e.g. 10.0.0.0/8, 172.16.0.0/12, 192.168.0.0/16
  + not globally unique (NUS and NTU can use the same IP addresses within their respective subnets)
  + used in organizations, universities, homes
  + not routable on the backbone internet, but are routable within an organization (e.g. NUS uses 172.26.x.y and any device that is connected to NUS WiFi can communicate to each other)
  + ISPs will drop any packet with src IP that are part of the private networks

NAT routers must:

* Replace (src IP address, port #) of every outgoing datagram to (NAT IP address, new port #)
* Remember (in NAT translation table) the mapping from (src IP address, port #) to (NAT IP address, new port #)
* Diagram

  Description automatically generatedReplace (NAT IP address, new port #) in destination fields of every incoming datagram with corresponding (src IP address, port #) stored in NAT translation table.

Diagram

Description automatically generated\* Max number of connection to a NAT router is limited by the # of port numbers available =

\* WAN side port number must be different, but LAN side port number can be the same

Motivations and benefits:

* No need to rent a large range of public IP addresses from ISP: 1 public IP for NAT routers can service 65536 devices within the subnet
* All hosts use private IP addresses. We can change addresses of hosts in local network without notifying the outside world
* We can change ISP without changing addresses of host in local net
* Hosts inside network not explicitly addressable and visible by outside world (security benefit)

Challenges

Peer-to-peer applications don’t work directly since many residential computers use NAT. 2 devices who are in 2 different private networks cannot communicate directly

**7.4 Routing Algorithms**

The Internet exists as a hierarchy of Autonomous Systems (AS) which are organization units e.g. Facebook, Google, Singtel Starhub etc. Each AS will then have multiple subnets within them. Routing on the Internet is hence done hierarchically due to the decentralized administration.

2 types of routing:

1. **Intra-AS routing**
   * Finds least cost path between 2 routes within AS
   * Single admin, no policy decisions are needed
   * Focused on performance
   * Commonly used protocol: RIP, OSPF
2. **Inter-AS routing** (NOT COVERED)
   * Handles interface between AS
   * Admin wants control over how traffic is routed
   * Policy may dominate performance
   * Uses BGP

7.4.1 Distance Vector Algorithm

* Routers know physically-connected neighbours and link costs to neighbours
* Routers exchange local views with neighbours and update own local view based on neighbour’s view
* Iterative process
  1. Swap local view with direct neighbours
  2. Update own local view
  3. Repeat until no change to local view
* Follows Bellman-Ford algorithm:

Diagram

Description automatically generatedwhere is the cost of the least-cost path from x to y, is the cost of link between router x and y, min is taken over all direct neighbours of x

7.4.2 RIP (Routing Information Protocol)

* Implements DV algorithm but uses hop count as the cost metric (i.e. insensitive to network congestion)
* Entries in the routing table are aggregated subnet masks (routing to destination subnet)
* Exchange routing table every 30 seconds over UDP port 520
* “Self-repair”: if no update from a neighbouring router for 3 minutes, assume neighbour has failed
* Properties:
  + Distributed: Each node received info from one or more of its neighbours, performs calculations then distributes it back to neighbours
  + Iterative: Process continues until no more updates available
  + Asynchronous: does not require all node to operate in step with each other

**7.5 ICMP (Internet Control Message Protocol)**

* used by hosts & routers to communicate network-level information
  + error reporting: unreachable host/network/port/protocol, TTL reach 0
  + echo request/reply (ping, traceroute)
* Carried in IP datagrams (ICMP header starts after IP header)
* Table

  Description automatically generatedTypes and Codes:

**Lecture 8: Link Layer Part 1**

IP datagram may travel through multiple routers and links before it reaches its destination.

* Link layer sends datagram between adjacent nodes (hosts or routers) over a single link
  + IP datagrams are encapsulated in link layer frames for transmission
  + Different link-layer protocols may be used on different links
    - each protocol provides a different set of service (e.g. PPP, 4G/5G, Ethernet)
    - however, there is a basic set of services that the link layer provides to network layer

Diagram

Description automatically generated**8.1 Routing Overview**

Mainly 3 scenarios:

* sender and receiver in same subnet (same IP prefix): use ARP protocol
* receiver outside of subnet: send data via default gateway, use RIP/OSPF protocol to route within AS
* receiver outside AS: use BGP protocol

**8.2 Link Layer Services**

1. **Text

   Description automatically generated with low confidenceFraming**: Encapsulate datagram into frame, adding header and trailer
   * Trailer and header changes at each link
2. **Link access control**: coordinates which nodes can send frames at certain point of time when multiple nodes share a single link
3. **Reliable delivery**: error detection and correction, often used on error prone link e.g. wireless link but not used on low bit-error link e.g. fibre
4. **Error Detection**: detect errors caused by signal attenuation or noise and possibly signal sender for retransmission or drop corrupted frame
5. **Error Correction**: receiver identifies and correct bit error(s) without retransmission

**8.3 Link Layer Implementation**

* Link layer is implemented in “adapter” (aka Network Interface Card [NIC]) or on a chip
  + e.g. PCIe card, 802.11 (WiFi card)
* Typically implemented as hardware for faster speed
* Adapters are semi-autonomous, implementing both link and physical layer

**8.4 Error Detection and Correction**

Popular error detection scheme:

* Checksum (used in TCP/UDP/IP)
* Parity checking
* CRC (Cyclic Redundancy Check): typically faster than checksum due to using bitwise arithmetic, often used in link layer
* Error detection schemes are not 100% reliable
  + may miss some errors but rarely
  + larger error detection and correction (EDC) fields yields better detection but uses more overhead
    - e.g. redundant information, additional calculation etc.

8.4.1 Parity Checking

1. Single bit parity
   * detect single bit errors in data (unable to correct)
   * parity bit is 1 if # of “1” bit is odd else 0 if # of “1” bit is even
   * only has 1 bit overhead
2. Two-dimensional bit parity
   * can detect and correct single bit errors in data
   * Chart, scatter chart

     Description automatically generatedcan detect any 2-bit error in data

8.4.2 Cyclic Redundancy Check (CRC)

* Text

  Description automatically generatedPowerful error-detection (high error detection rate) used in ethernet and WiFi
  + D: data bits, viewed as binary number
  + G: generator of bits agreed by sender and receiver
  + R: generate CRC of bits (the higher the *R* value, the higher the detection rate)
* CRC calculation done in bit-wise XOR without carry/borrow, done in link layer since it has access to hardware. Not done in upper layer since it would have to rely on software which is slow
* Receiver divides (*D, R*) by *G* and would know if an error occurred if non-zero remainder detected

**8.5 Types of Network Links**

1. Point-to-Point Link
   * Diagram, schematic

     Description automatically generatedsender and a receiver connected by a dedicated link
   * Example protocols: Point-to-point (PPP), Serial Line Internet Protocol (SLIP)
   * e.g. RJ45 Ethernet, point-to-point link between ethernet switch and host
2. Broadcast link (shared medium)
   * A picture containing text, printer

     Description automatically generatedmultiple nodes connected to a shared broadcast channel
   * when a node transmits a frame, the channel broadcasts the frame and each other node receives a copy
   * e.g. 802.11 WiFi, Satellite (StarLink), Ethernet with Bus Topology

8.5.1 Multiple Access Protocols

* In a broadcast channel, if 2 nodes transmit simultaneously, every node will receive multiple frames at the same time frames collide at nodes and data will be invalidated
* Have multiple access protocol which is a distributed algorithm that determines how nodes share channel, i.e. when a node can transmit
  + coordination about channel sharing must use channel itself no out-of-band channel signalling
* 3 broad classes:
  + Channel partitioning
    - Divide channel into fixed, smaller pieces (e.g. time, slots, frequency)
    - allocate piece to node for exclusive use
  + “Taking Turns”
    - nodes take turns to transmit
  + Random Access
    - channel not divided; collisions still possible
    - must have ways to recover from collision

8.5.2 Channel Partitioning Protocols

No collisions, but low efficiency of bandwidth

* TDMA (time division multiple access)
  + Access to channels in “rounds”
  + Each node gets fixed length slot (length = frame transmission time) in each round)
  + Chart

    Description automatically generatedUnused slots are idle
* FDMA (frequency division multiple access)
  + Channel spectrum divided into frequency bands (e.g. radio uses this)
  + Each node assigned 1 frequency band
  + Unused transmission time in frequency bands go idle
  + Diagram

    Description automatically generatedNodes don’t have to wait but only uses of bandwidth

8.5.3 “Taking Turns” Protocol

* Polling

Diagram

Description automatically generated

* + master node “invites” slave nodes to transmit in turn (no idling)
  + **concerns**:
    - polling overhead: last node must wait for master to poll whole list before getting its turn (minor overhead)
    - single point of failure (master node)
* Token passing
  + control token passed from 1 node to the next sequentially (round robin)
  + **Concerns**:
    - single point of failure (token)
    - if client with token disconnect must ensure only 1 token is regenerated (lots of code to handle error)
    - token overhead

8.5.4 Random Access Protocol

* when node has packets to send
  + no prior coordination among nodes
  + if 2 transmitting nodes collision
* Random Access Protocol must specify:
  + how to detect collision
  + how to recover from collision (e.g. delayed retransmission)
* Slotted ALOHA
  + Assumes all frames are of equal size, time is divided into slots of equal length (length = frame transmission time), nodes start to transmit only at beginning of slot
  + **Operations**:
    - listens to channel while transmitting (collision detection)
    - if collision happens: node attempts to retransmit a frame in subsequent slot with probability *p* until it succeed
    - A screenshot of a computer

      Description automatically generated with medium confidence*p* small: more unused slots, *p* big: collisions
* Pure (unslotted) ALOHA
  + no slot, no synchronization
  + when there is a fresh frame transmit immediately
  + Chance of collision increase!
    - Frame sent at collides with other frames sent within 1 time unit:
* CSMA (Carrier Sense Multiple Access)
  + **Chart

    Description automatically generated**sense channel before transmission
    - channel sensed idle: transmit
    - channel sensed busy: defer transmission
  + collision still can happen e.g. when two nodes sense the channel idle at the same time (due to propagation delay) and both start transmission

* CSMA/CD (Collision Detection)
  + Diagram

    Description automatically generatedWorks similarly to normal CSMA but transmission aborted when collision detected (reduce channel wastage)
  + Retransmit after random time
  + If frame size is too small:
    - collision happens without being detected
    - no retransmission
    - must have minimum frame size: Ethernet uses minimum frame size of 64 bytes
* CSMA/CA (Collision Avoidance)

Diagram

Description automatically generated

**Lecture 9: Link Layer Part 2**

Diagram

Description automatically generated with medium confidence**9.1 MAC (Media Access Control) Address**

* Every NIC has a 48-bit MAC address (aka Physical or LAN address) e.g. 5C-F9-DD-E8-E3-D2
  + MAC address allocation administered by IEEE and first 3 bytes identifies vendor of adapter
  + used to send & receive link layer frames
  + On receiving a frame, NIC checks if destination MAC address matches its own address using hardware (very fast).
    - If yes, extracts datagram and passes to protocol stack.
    - Otherwise, discards the frame.

Graphical user interface, text, application

Description automatically generated9.1.1 IP vs MAC Address

**9.2 ARP (Address Resolution Protocol)**

* Each IP node (routers, hosts) has an ARP table
  + acts as a caching mechanism
  + empty when computer starts up and slowly fills ups with each request
  + stores mapping of IP address and MAC address of other nodes in the same subnet

9.2.1 Sending Frames

* In the **same subnet:**
  + If *A* knows *B*’s MAC address from ARP table:
    - create frame with *B*’s MAC and send it
    - Only *B* will process this frame, other nodes may receive the frame but will ignore it
  + Else:
    - *A* **broadcast an ARP query packet** containing *B’s* IP address (Dest MAC address set to **FF-FF-FF-FF-FF-FF**)
    - Other nodes in subnet will receive packet but only *B* replies with its own MAC address
    - *A* will cache *B*’s MAC address to its ARP table
* To **another subnet**:
  + *A* cannot just put *B*’s MAC address as dest MAC since router will ignore the frame due to mismatch of MAC address
  + *A* create a link-layer frame with router’s MAC address and *B*’s IP address.
  + Diagram

    Description automatically generatedRouter will then move datagram to outgoing link, construct a new frame with *B’s* MAC address

**9.3 Local Area Networks (LAN)**

* computer network that interconnects computers within a geographical area (e.g. office, universities)
* Examples:
  + Ethernet: IEEE 802.3 standard
  + WiFi: IEEE 802.11 standard
* Exists other networks like MAN (Metropolitan area network), WAN (wide area network)
* Within LAN can have many subnets

**9.4 Ethernet**

9.4.1 Ethernet Topology

* **Bus Topology**: All nodes are connected and can collide with each other.
* **Star Topology**: switch in the centre and nodes are connected to that switch. They do not collide with each other.

Table

Description automatically generated with low confidence9.4.2 Ethernet Frame Structure

* **Preamble**: 7 bytes of 10101010 and 1 byte of 10101011
  + used to synchronize receiver and sender clock rate (allow receiver to know how long is 1 bit) so that it can know how long is 20 bits of 0 accurately
* **Destination MAC Address**: If the NIC receives a frame with either matching address or the broadcast address, it will pass the data in the frame to the network layer protocol. Else it will discard.
* **Type**: Higher layer protocol used, usually IP
* **CRC**: for corruption detection
* **Payload**: Minimum size of 46 bytes for collision detection, max size is 1500 (MTU).
  + maximal payload size makes it easier to implement efficient data buffer management algorithms in the NICs, switches and routers.
  + also helps to make sure that senders do not “hog" shared links and other senders get a chance to transmit as well.

9.4.3 Ethernet Data Delivery Services

* **Connectionless**: No handshaking between sender and receiver.
* **Unreliable**: No ACK or NAK. Data in dropped frames will be recovered only if the initial sender uses higher-layer rdt (e.g. TCP). Corrupted frames are also dropped.
* **Multiple Access Protocol**: CSMA/CD with binary (exponential) backoff

9.4.4 Ethernet CSMA/CD Algorithm

1. NIC receives datagram from the network layer and creates a frame.
2. If NIC senses that the channel is idle, it will start frame transmission. Else it will wait until idle.
3. If NIC transmits the entire frame without detecting another transmission, NIC is done.
4. If another transmission is detected, NIC aborts and sends a jam signal, which tells all other nodes that a collision has been detected
5. NIC enters exponential backoff
   * after collision, NIC chooses *K* at random from {0,1,2,…,} more collisions, longer wait time (attempt to adapt retransmission to estimated current load)
   * NIC waits K\*512 bit times and returns to step 2

9.4.5 Ethernet Switch

Link-layer device used in LAN that stores and forward ethernet frames. It examines incoming frame’s MAC address and forward frame to 1 outgoing links.

* **Layer 2 device**: Only implements physical and link layer protocols. No IP address.
* **Transparent to hosts**: hosts are unaware of presence of switches
* **Collision-free**: each host have dedicated (full duplex) connection to switch and switch buffers frames hosts can send frames to each other simultaneously.
  + Note that that may be collisions within the switch itself

9.4.6 Switch Forwarding Table

A switch has multiple interfaces, and it will need to know which nodes are reachable via which interface. This is done via switch forwarding tables, which have the following entry format:

< MAC address of Host, interface to reach host, TTL >

* **Self-learning**: Whenever the switch receives a frame from host A, it will record the interface that reaches A in its switch table, and send all frames for A to that interface.
* **Broadcast**: If the destination host is not found in the switch forwarding table, the switch will broadcast the frame to all outgoing links.
* **Hierarchical**

Diagram

Description automatically generated9.4.7 Switch vs Routers

|  |  |
| --- | --- |
| **Switch** | **Router** |
| Layer 2 Device | Layer-3 Device |
| Self-learning | Need Manual Configuration |
| Forwards link layer frames | Forward IP datagrams |
| Used in a subnet | Used to connect subnets |

**Lecture 10: Multimedia Networking**

There is a rise in use of multimedia applications where it is projected that 80% of traffic will be video in 2021. Videos are delivered **Over-the-Top (OTT)**

**10.1 Properties of Audio**

* **Sampling**: To convert audio analog signal to digital signal, we must sample the signal at some constant rate.
  + **Telephone**: 8000 samples/sec
  + **CD**: 44,100 samples/sec
* **Quantisation**: Each sample is rounded to one of a finite number of values. Number of quantisation typically in power of 2. e.g. 216 for CD.
  + Chart, histogram

    Description automatically generatedNyquist-Shannon sampling theorem:

, = sampling frequency, = highest signal frequency

* **Concatenation and Decoding:** All values represented as bits and all samples will then need to be concatenated back together. To decode, we convert it back to analog signal. This signal is an **approximation of the original** since some sounds are lost in sampling and quantisation.
  + **Speech Encoding:** 8000 samples/sec, 256 quantized values (8bits) = 64kbps
  + **CD:** 44,100 samples/sec, 16 bit quantized value = 705.6kbps for mono and 1.411 Mbps for stereo
* **Compression:** lossless (e.g. zip) and lossy (mostly used for media)

**10.2 Properties of Video**

* Video is just a sequence of images displayed at a constant rate e.g. 30 images/sec (fps)
* each image is an array of pixel and each pixel is represented by RGB bits (8 bits for each of R,G,B)
* Redundancies:
  + **Spatial:** instead of sending N values of same colour, send (colour, N)
  + **Temporal:** only send difference between 2 consecutive image instead of complete frame
* **Bit Rate:**
  + **CBR (constant bit rate):** video encoding rate. Might lead to variable quality whereby we have low quality for complex frames and high quality for simple frame
  + **VBR (variable bit rate):** video encoding rate changes according to spatial and temporal coding.
* **Compressors:**
  + **MPEG 1 (CD-ROM):** 1.5 Mbps
  + **MPEG2 (DVD):** 3-6Mbps
  + **MPEG4/H.264:** used in internet, < 2Mbps

**10.3 3 Application Types**

* **streaming, stored audio, video**
  + streaming: can begin playout before downloading entire file
  + stored (at server): transmit faster than audio/video will be rendered (need to buffer at client)
  + e.g. YouTube, Netflix, Hulu
* **conversational (“two-way live”) voice/video over IP**
  + interactive nature limits delay tolerance
  + e.g. Zoom, Skype, Teams
* **streaming live (“one-way live”) audio, video**
  + can be a few seconds late
  + e.g. live sporting event

**10.4 Streaming Stored Video**

10.4.1 Challenges

* **Continuous playout constraint:** once client playout begins, playback must match original timing
  + hard to achieve as network delays are variable (jitter) so will need client-side buffer to match playout requirement
* **interactivity**: pause, fast-forward, rewind, jump through video etc.
* video **packet may be loss**

10.4.2 Client-side Buffering

Diagram

Description automatically generated with medium confidence

When video arrives at client, we do not immediately play the video, instead we keep a reserve of video in the buffer of client. Only play when we have certain amount of video in buffer (typically a few seconds worth).

* **Absorbs variation in delay**: If a certain piece is delayed, as long as it arrives before the reserves are depleted, user will not notice delay
* **Bandwidth change resistance**: If bandwidth briefly drops, user can continue to continuously playback as long as buffer not depleted
* If **fill rate < playout** rate: buffer eventually depletes (cause freezing of video until buffer fills again)
* If **fill rate > playout** rate: less likely to deplete buffer but larger delay until user begins watching

**10.5 UDP streaming**

* **Pushed-based streaming**: send rate = encoding rate = constant rate. Transmission rate can be oblivious to congestion level since no rate-control protocol in UDP
* **short playout delay**: 2-5 seconds to remove network jitter
* **UDP may not go through firewall**
* **Uses RTP (real-time transport protocol)**
* **Small buffer:** usually buffer < 1s

**10.6 HTTP Streaming**

* **Pull-based streaming**: retrieve multimedia file through HTTP GET request
  + **bigger length of data**: each segment can be a few MiB
  + sends at max rate under TCP
* **Fill rate fluctuates:** due to TCP congestion control, retransmission
* **Larger playout delay:** wait till buffer fills up
* HTTP/TCPpasses through firewall more easily

**10.7 VoIP**

* Requirements:
  + delay of < 150ms is considered good, any delay > 400ms would be noticeable and may impair interactivity
* Characteristics:
  + **Talk spurts**: We don’t always talk during a conversation. Hence we can just generate packets during talk spurts at 64kbps. 20ms chunks at 8KB/s = 160 bytes/s
  + application-layer header added to each chunk and encapsulated in UDP or TCP segment and then sent at 20ms interval during talk spurts
* Losses/delays:
  + **network loss**: IP datagram loss due to network
  + **delay loss**: IP datagram arrive too late for playout, typically 400ms max tolerable delay
  + loss tolerance: loss between 1-10% is okay
* fixed playout delay: receiver attempts to playout each chunk at exactly q ms after chunk was generated
  + any chunk that arrives at chunk t+q will be too late
  + tradeoff in choosing q
    - large q: less packet loss, more delay
    - small q: low delay but more loss
  + Solution: **Adaptive playout Delay**
    - aim to estimate network delay and adjust playout delay at beginning of each talk spurt
    - silent periods are compressed and elongated
    - Diagram, timeline

      Description automatically generatedchunks still playout at every 20ms
* Recovery from Packet Loss
  + Forward Error Correction (FEC): Send enough bits to allow recovery without retransmission
    - **Simple FEC:** for every N chunks, we create a redundant chunk by XOR-ing the original N chunks and send N + 1 chunks (1/N overhead). Allows us to reconstruct 1 chunk from N+1 chunk.
    - **Piggybacking**: Send lower resolution stream as redundant information
  + Diagram

    Description automatically generatedInterleaving:

**10.8 Real-time Protocol**

RTP specifies packet structure for packets carrying audio, video data. RTP packet provides: payload type identification, packet sequence number numbering, time stamping.

* RTP runs in end systems.
* **UDP**: RTP packets are encapsulated in UDP segments. RTP libraries provide **transport-layer** interface that extends UDP
  + port#, IP address
  + payload type identification
  + packet sequence numbering
  + time-stamping
* **Interoperability**: if 2 VoIP applications both incorporate RTP, they may be able to communicate with each other
* Diagram

  Description automatically generated**RTP Suite**:
* audio chunk + RTP header forms a RTP packet which is encapsulated in UDP segment
* **RTP Header**:
  + **payload type** (7bits): indicate type of encoding used. If sender changes encoding during call, sender informs receiver via payload type field (refer to lect notes)
  + **Sequence#** (16bits): increment by one for each RTP packet sent. Used for detecting packet loss and restoring packet sequence
  + **timestamp** (32bits): sampling instant of first byte in RTP packet (refer to lect notes for more info)
  + **SSRC** field (32 bits): identify source of RTP stream. e.g. Used to distinguish which skype stream if you have multiple skype client opened

**10.9 WebRTC**

WebRTC is an open framework for the web that enables Real-Time Communications (RTC) capabilities in the browser, via simple JavaScript APIs. It makes use of RTP and RTCP .

**10.10 DASH (Dynamic Adaptive Streaming over HTTP)**

**Advantages:**

* **Simple server**: regular web server (no state, scalable)
* **No firewall issue**: uses standard port 80, TCP
* Standard (image) web caching works

**Disadvantages:**

* DASH based on media segment transmission, typically 2-10s in length
* **Does not provide for low latency** for interactive 2-way communication
* **High Overhead:** original media need to be split into different streamlet and transcoded to different quality

10.10.1 How it works?

**Idea:** Split video into streamlets, where each streamlet is of a different quality (e.g. low, medium high)

* Webserver provides a playlist (typically of .mpd format) which is a file that specifies all available qualities and all streamlets available
* As client plays video, execute adaptive bitrate algorithm (ABR) to determine which segment to download next

**Lecture 11: Network Security**

**11.1 What is Network Security**

**Properties:**

* **Confidentiality**: Only sender should be able to read and understand the message.
* **Integrity**: sender, receiver want to ensure message not altered (in transit, or afterwards) without detection
* **Availability/Availability**: Services must be available to authorised parties who need it.
* **Authenticity**: Ability for sender and receiver to confirm the identity of each other.
* **Non-repudiation**: Sender cannot convincingly deny having sent something.

**Potential Attacks**:

* **Eavesdrop**: intercept message
* **Insert** message into connection (compromise integrity)
* **Impersonation**: can fake (spoof) source address in packet (or any field in packet)
* **Hijacking**: “take over” ongoing connection by removing sender or receiver, inserting himself in place
* **Denial of Service (DoS)**: prevent service from being used by others (e.g., by overloading resources)

**Terminologies:**

* **cipher-text-only attack:** attacker has ciphertext that he can analyse. Broken either by brute force or statistical analysis
* **known-plaintext-attack:** attacker has plaintext corresponding to ciphertext (e.g. attacker know pairing of letters in Caesar cipher)
* **chosen-plaintext-attack:** attacker can get ciphertext for plaintext

**11.2 Symmetric Key Cryptography**

Both Alice and Bob share same key. Challenge is how to communicate said key to the other party securely.

* **Substitution Cipher:** Replace one character for another. The encryption key is a one-to-one mapping of the characters used.
  + Mapping is static
  + Key size is huge
  + Weak against frequency attack
* **Caesar Cipher**: Left or right rotation of the alphabets or characters used. Encryption key is the shift number.
  + Still has static mapping but key size is only 1 int
* **Multiple Substitution Cipher**:

1. Choose *n* substitutions ciphers.

2. Choose a random cyclic pattern of numbers 1 to

*n*.

3. Repeat this cyclic pattern until it’s the plaintext

length.

4. For each character in plaintext, find the corresponding number, then use that substitution for that character.

Key need not be just a n-bit pattern which makes it **more resistant to frequency attack**

* **Table

  Description automatically generatedDES & AES**

**11.3 Public Key Cryptography**

Sender and receiver has a pair of private and public key. They do not share their private key but public key can be known to anyone. Often **used in HTTPS** protocol

**Requirements:**

Given public key, it should be computationally impossible to compute private key.

11.3.1 RSA

1. Choose two distinct large prime numbers and (e.g. 1024 bits each)
2. Compute and
   * used as the modulus for both public and private key
3. Choose such that and is coprime with (no common factor)
4. Choose d such that is exactly divisible by i.e.
5. Public key is () and private key is ()

11.3.2 RSA Encryption and Decryption

1. To encrypt , compute
2. To decrypt , compute

11.3.3 Why is RSA Secure?

Factoring which is a huge number is computationally difficult without knowledge of and

11.3.4 Combining Public and Symmetric Key Cryptography

Using Public key cryptography to encrypt messages is too slow (DES is 100 times faster than RSA). Hence, we can first use RSA to conduct symmetric key exchange and then use the symmetric key to secure future communications.

Table

Description automatically generated**11.4 Recommended Key Lengths**

**11.5 Digital Signature**

We want to have a signature so that we know who is sending the message and that that person cannot refute having sent it, i.e. **authenticity** and **non-repudiation.**

The easiest way to achieve this is to use private key of RSA to sign the message then the other person can use public key of RSA to decrypt and verify that it is indeed from the authentic sender. However this is **computationally expensive**.

**11.6 Hash Functions**

**Properties**:

* many-to-1 (pigeon hole principle)
* produces fixed size message digest
* Given a digest , it is computationally infeasible to find m such that =
* **Graphical user interface, text, application, email

  Description automatically generated**Diagram

  Description automatically generatedInternet checksum has some properties of Hash but it is easy to find another message with same hash function
* Common hash function:
  + **MD5**: Computes **128-bit** message digest in 4-step process. Is actually obsolete now as collisions can be found within a minute.
  + **SHA-1**: 160-bit message digest
* **Password Hashing:**
  + Passwords are not stored in plaintext but rather hashed and stored.
  + To make it hard to find similar plaintexts by simply checking the hashes, a salt i.e. random string of characters is added to the front of the password before being hashed.
* **Public Key Certification**
  + We need a trustworthy source to get the public keys from
  + Public keys are provided by an organisation known as **Certificate authority (CA)**
  + An entity, e.g. a host or a router, needs to register its public key with a CA, often requiring to provide some proof of identity.
  + CA creates a certificate binding that entity to its key (along with a lot of other information) and this certificate is digitally signed by the CA.
  + A sender will thus get the receiver’s certificate from a CA and apply the CA’s public key to the certificate to get the receiver’s public key.
* VPNs:
  + Institutions often want **private network** for security but its very costly (separate routers, links, DNS, infrastructure)
  + **Diagram

    Description automatically generated**VPN encrypts inter-office traffic before sending out into the public internet and is logically separate from other traffic

**Graphical user interface, text, application, email

Description automatically generatedText

Description automatically generatedGraphical user interface, text, application

Description automatically generatedFull Path Taken After Entering** [**www.facebook.com**](http://www.facebook.com)

**Miscellaneous**

Modern routers are capable of many functions:

* acts as DHCP server for local network
* acts as DHCP client for ISP (gets dynamic IP address from ISP)
* acts as NAT gateway
* routing

**QnA**

**Why not use MAC instead of IP for routing?**

An IP address logically comprises two parts: network prefix and host ID. This is designed to facilitate routing: routers check prefix and deliver a packet to an

aggregated destination network. If MAC address is used instead, hierarchical

routing cannot be achieved. For example, MAC address is burnt in ROM and usually cannot be changed. When people carry their laptops around the world, devices in a subnet won’t have common prefix in MAC addresses. This makes routing difficult as routers must remember routing for every single MAC address.

**FDMA vs TDMA**

* **FDMA:** Good for sending messages that requires short latency since channel will be readily available. However, if big files are sent then FDMA might take longer than TDMA since it only has 1/N \* Bandwidth per channel
* **TDMA:** Good for sending large files in a short time since it has dedicated bandwidth usage for 1/N of the time. Bad if you want to send message with low latency since you have to wait for the next slot.

**Applications that won’t function properly from a computer connected to a NAT router**

* Web servers
* Email servers

**Things that works:**

* Messaging client
* Networked printer
* Email client

**Utilization of Link in Sliding Window**

where U = utilization, L = packet length, R = link bandwidth, N = window size