Computer Networks

Networks intro

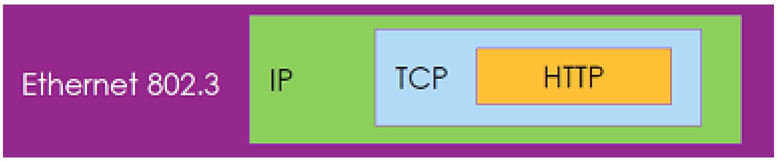
- Definition of Computer Networks
    - Components:

Distributed systems (applications)
Networks (messages)
Communications (bits)

    - Example Networks:
        - car key with car; sensors network with their controllers (either one-way or two-ways)
- Design principle:
    - Dumb network & Smart users
        - networks don’t store too much info but just pass the info
- Internet 'preferred' protocol stack
    - Application
        - deliver functionality
    - Transport
        - ensure end-to-end performance
    - Network
        - send packet over multiple links
    - Physical & Link
        - transmit frames
- Messages in Layers
    - Overview

Layer	What it transports (Protocol Data Unit)	How they connect
Application	Messages/Data	Proxy, gateway
Transport	Segments/Datagrams	
Network	Packets (!!)	Router
Link	Frames (cells, circuits)	Switch, Bridge
Physical	Bits	Hub (repeater)

    - Messages Encapsulation on Each Layers



- e.g. Ethernet frame's payload contains IP packet(s)

- Relations between Layers
  - Network layer is the commander on router (may sit on switch)
    - configure the link layer protocol...
  - Application layer is the commander on host
    - configure the network layer protocol...

Information Transmission - Communication

- Problems:
  - Attenuation - loss of energy
  - Noise - gain of energy
  - Delay distortion - smearing
  - Frequency cut-offs - loss of information
  - Frequency-specific attenuation
- Approaches:
  - Circuit switching
    - Communication oriented
    - Pros:

      hardware level guarantee fixed (reliable) quality during the communication
    - Cons:

      lots of waste of capacity (no sharing) & explicity resouce allocation $\Rightarrow$ expensive to scale

      networks need to store state (info of connection) $\Rightarrow$ multiple single points of failure

      hard to recover from failure
  - Multiplexing
    - Spatial division multiplexing - (more wires)
    - Time division multiplexing - (take turns)
    - Frequency, Amplitude, Phase multiplexing
- Analog vs. Digital
  - Digital
    - easy to represent, store and regenerate
  - Analogue
    - represent the natural world
    - Sine wave appears every where $\Rightarrow$ Fourier transformation
  - $\Rightarrow$ Measureing & Creating Sine wave
    - encoding the feature of sine wave

e.g. frequency, amplitude and phase

- use Sine wave as carrier

  (especially in wireless communication, yet constant voltage is easier in wired transmission)

- Encoding of Bits into Signal
  - Modulation & Demodulation
    - Modulation: turning bits into signals
    - Demodulation: turning signals into bits
  - Single Bit Encoding in Modulation
    - Amplitude modulation (AM)
    - Frequency modulation (FM)
    - Phase modulation (PM)

      (detect phase shift: need sync ⇒ clock line can be represented by freq or in other forms)
  - Symbol Encoding in Modulation
    - Symbol: bit pattern

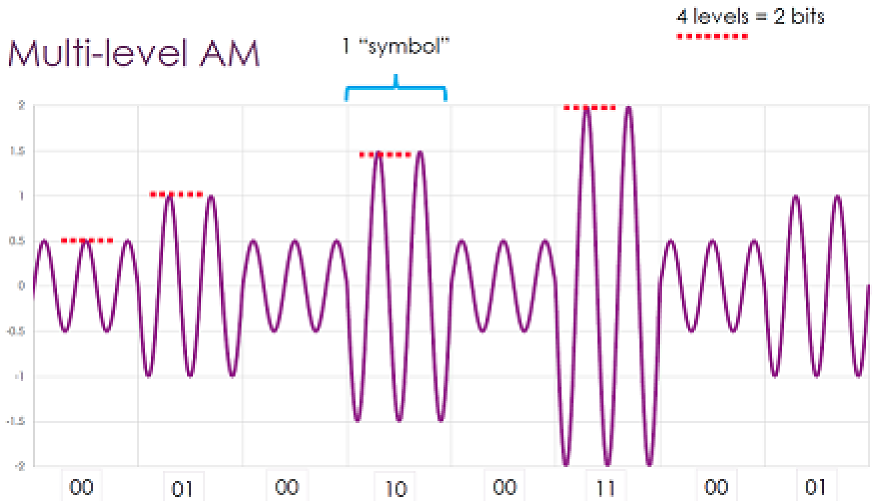
      ⇒ 1 symbol / second > 1 bit / second
    - Multi-level modulation

      Multi-level AM = Amplitude Shift Keying (ASK)

      Multi-level FM = Frequency Shift Keying (FSK)

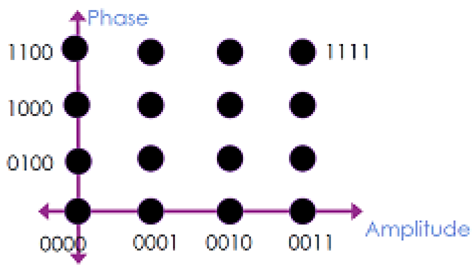
      Multi-level PM = Phase Shift Keying (PSK)

      e.g.:

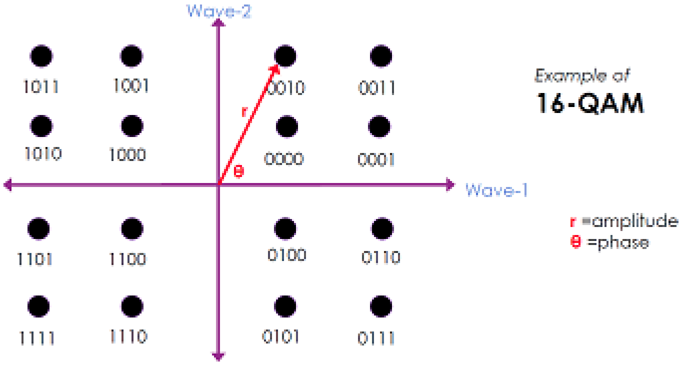


    - Phase + Amplitude Modulation:

      ⇒ Constellation diagram:



⇒ Quadrature Amplitude Modulation (QAM):



Other Modulation:

256-QAM CableTV system

4096-QAM Powerline data

65535-QAM ADSL

x-QAM depnding on the needs and techniques available

Note:

phase-amplitute-frequency modulation (on 3 axis) not commonly used

Because: frequency usually used to denote channel (using carriers)

⇒ frequency to avoid interference & harder to change

- Bands
  - Baseband: constant voltage
    1. Baseband signal: lowpass signal, using <u>constant voltage as carrier</u>

       ⇒ non-modulated signal

       ⇒ only non-zero near the origin of frequency spectrum

       e.g. ASK, OOK (On-off keying)
    2. Baseband channel: lowpass channel, typically an unfiltered wire
    3. Baseband transimission: <u>transferring bit steam in line coding on typically an unfiltered wire</u>
  - Passband: the range of frequencies that can pass through a filter
    1. Passband signal: use <u>single frequency as carrier</u>

       ⇒ a signal with energy only in a passband, up-converted to higher frequency

       ⇒ digital modulation employed

       ⇒ integrate low-frequency wave (info wave) into a higher-frequency carrier wave
    2. Passband channel: channel of range of frequency after bandpass filters employed
    3. Passband transmission: (carrier-modulated transmission)

       <u>using passband signal to transfer info, typically in wireless transmission</u>
  - Broadband:
    1. Broadband signal: <u>use multiple frequency carries across a range</u>

$\Rightarrow$ FSK

        - Bandwidth: a specific range of frequencies

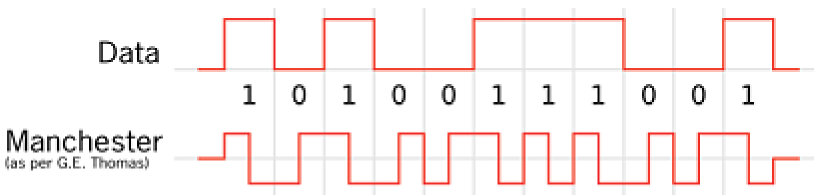
          can be divided at your choice & capacity of the technology allowed

- Limitation in Transmission Quality:
    - Shannon–Hartley theorem (Capacity Limit)
        - $C = B\log_2\left(1+\frac{S}{N}\right)$, where
          C is channel capacity in bits / seconds;
          B is bandwidth in Hz;
          S is expectation of received signal power over the bandwidth in volts^2;
          N is average power of noise and interference over the bandwidth in volts^2.
        - lowest sampling frequency of twice as the incoming signal to get a perfect reconstruction
    - Expressing Transmission Quality
        - Signal:Noise Rate (SNR) = Signal Energy : Noise Energy

          $\Rightarrow$ SNR in deciBel = $10 * \log_{10}(\text{Signal}/\text{Noise})$ dB
- Encoding Bits Sequence into Bits Patterns (regardless of modulation)
    - Key Concepts:
        - Map bits into patterns to reduce repetition
        - Signal each pattern with a transition
    - Bits Pattern Example: 4b/5b Code:
        - Mapping Table:

Given	Send	Given	Send
0000	**11110**	0100	**01010**
0001	**01001**	1000	**10010**
0010	**10100**	1101	**11011**
0011	**10101**	1111	**11101**

        - Features:
            1. avoid runs of 0, but can have maximal 6 1's in a row...
            2. trade bandwidth for reliability $\Rightarrow$ enable self-checking
    - Transition Example: Manchester Code:
        - Example:



        - self-clocking:
            1. a sync pattern in the front to denote the start (sync)
            2. the receiver can then identify if it is misaligned by half a bit period (prevent phase shift)

               (need to have rising/failing edge in every time unit, otherwise phase shift happens)

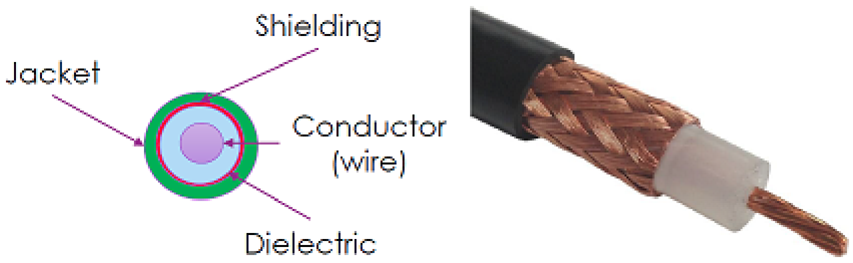
Physical Layer

Copper

- Characteristics
  - Physical:
    - Soft & bendable around the corner
    - Light; Malleable; Easy to make thin wire
    - Easy to add insulation & protection; Reasonably robust to oxidation
  - Social:
    - Cheap, compared to fibber (yet price is increasing)
  - Electrical
    - Shared medium (one voltage over the whole line)
    - Receive (RX) & Transmit (TX) on the wire:
      1. Half-duplex - each side takes turns to transmit & receive - Time Division Multiplexing
      2. Full-duplex - both ends can transmit & receive in parallel - Frequency Division Multiplexing
    - Reference of 'zero' ⇒ cables tend to have a pair of wires
    - Resistance:
      1. impedance, inductance (hate frequency change), etc...
      2. frequency related resistance: skin effect

         ⇒ in alternating current, higher frequency, higher resistance, more current close to skin

         ⇒ frequency attenuation
      3. Varies from cross-section: thinner wires, bigger resistance
    - Attenuation:
      1. loss of energy (in the form of heat, light and etc.)
      2. loss of frequency and etc...
- Noise in Signalling:
  - Random Wire Antenna: straight wires on the ground as an antenna
    - receiver for other signals
    - transmitter of its own signals
    - Electro-magnetic Interference （EMI, 电磁干扰）& Radio-frequency interference (RFI)
    - Coupling with adjacent wires ⇒ cross talk (expecially at near & far end - NEXT&FEXT)
  - Solving Antenna Problem
    - Protection - "Coaxical" cables



Pros: well sheilded - protection from noise & security (much less sending out), robust

Cons: single RX/TX, expensive

- Spatial Division Multiplexing ⇒ more wire in a cable (robust)

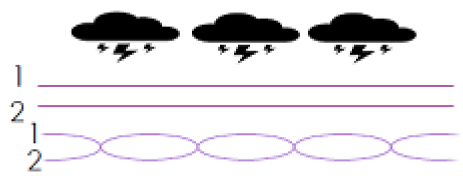
  Pros: full duplex, inverse multiplexing - multiple path to share (one-to-many & many-to-one)

  Cons:

  too many adjacent wires ⇒ cross talk

  long straight unsheilded wires ⇒ antennas problems remains

- Differential signalling
- Twisting wires



  Assumption: noise source has a direction

  ⇒ twisting to make sure noise added evenly

  ⇒ use the reference line to record the noise and then fileter it out

  Example: UTP (unshielded twisted paris - 网线)

  ⇒ combin with shielding: STP/FTP (shielded twisted / foiled twisted pair)

- Skew between Pairs:
  - Different lenghts between multiple pairs can result in un-aligned signal

    ⇒ affects inverse multiplexing

    ⇒ have to be in the same length within tolerance
- Resistance (Inpedance) Mismatch
  - Results in signal bouncing back to the sender

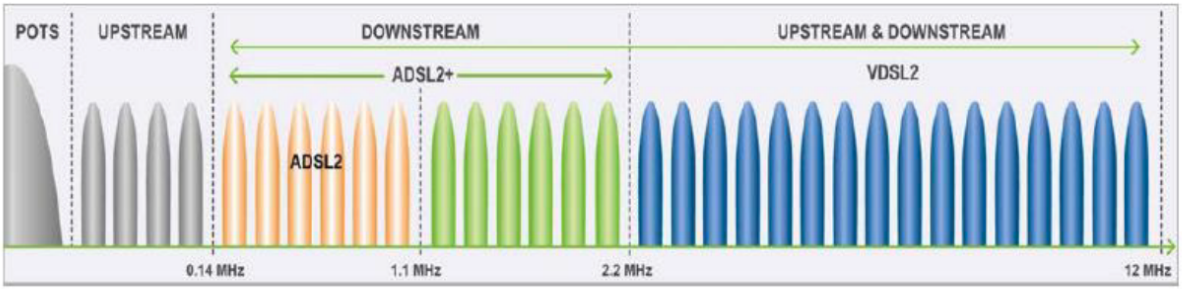
- Transmission on Copper
  - Speed:
    - kHz to MHz, enhanced by different keying & multiplexing technology
  - Distance:
    - Low data rate (< 1Mb/s) for longer distances (km)
    - High data rate (~100Mbs) for short distances (~500's m) E.g. DSL+
  - Downside:
    - Propagation delay (speed of electricity in copper = ~3us / km)

$\Rightarrow$ collision of two sender signalling at the same time

- Costs
  - Deployment
  - Protect Damage
    - easy to have a shared backbone
    - last mile exposed in the real world - insects, weathers, stealing, etc...
  - Last Mile Trad-off (last mile also refered as local loops sometimes)
    - cost of exchanges, distance for the final cable, quality of signal through the wire

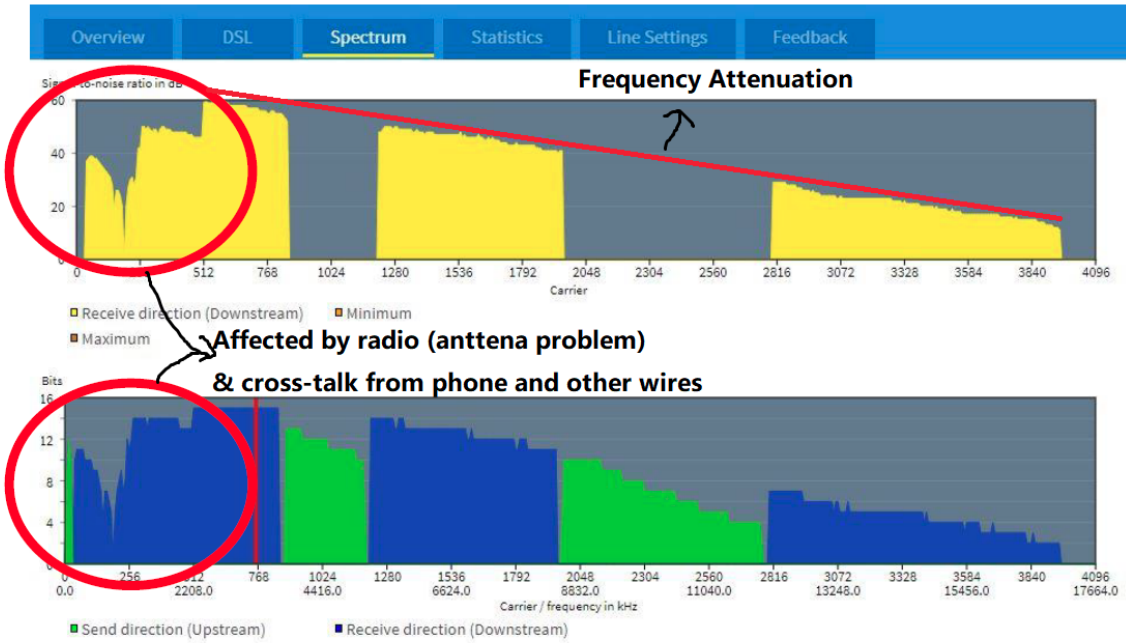
      Note: up to 4+ km from their exchanges
    - scalability
- Existing Last Mile Technology
  - DSL - based on existing telephone wires
    - evolving from ADSL to VDSL etc...

      later, DSL+: 16-bits (65535) QAM, FDM, ...
    - Assymetric: more on downstream performance



    - Pros: using the existing telephone line; co-exit with POTS (plain old telephone service)
    - Cons: limited performance; performance decreases over the distance

      $\Rightarrow$ may deploy more DSLAM to make average distance shorter
    - DSL Example:
      1. Computer -> modem (add info onto carrier)
      2. modem (s) -> DSLAM (aggregate signal from modems)
      3. DSLAM -> switch (decide which LAN it is in)
      4. switch -> router / switch (go to outer internet / transfer to another switch)
  - NBN - National Broadband Networks
    - Spectrum in Real World

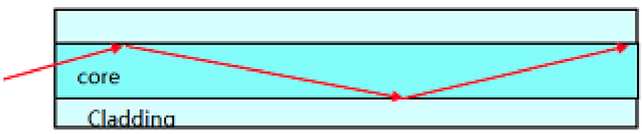


- Mixed-Technology
  1. NBN FTTx (Fiber to the x)
  2. Hybrid Fiber Coax and etc...

Fiber

- Characteristics
  - Physical:
    - Light weight, very robust to oxidation and water
    - Easy to make thin cable
    - Fragile when twisting & bending, hard to connect (need to melt it)
    - Good at distance (several km is trivial)
  - Social:
    - Expensive, compared to copper (yet price is decreasing)

      Note: fiber itself is okay, yet the end-point is expensive (⇒ usually use FTTx)
  - Electrical
    - Robust to electircal interference
    - High throughput: much higher freqeuncy (light) signal - start at THz
- Noise in Signalling:
  - Oblique Light Leaks
    - use another layer of glasses to reflec the light (with in a 'critical angle'), wrapped with jacket

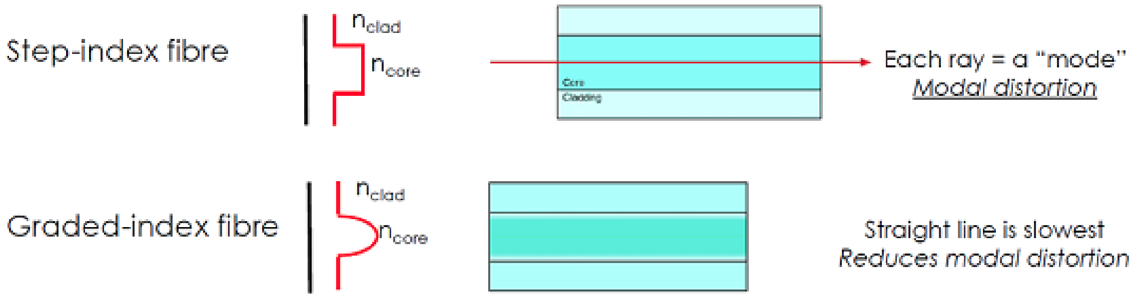


- Modal Distortion (varying distance for light to travel because of reflection)
  - use graded-index (缓变折射率) fiber instead of step-index (阶跃折射率) fiber

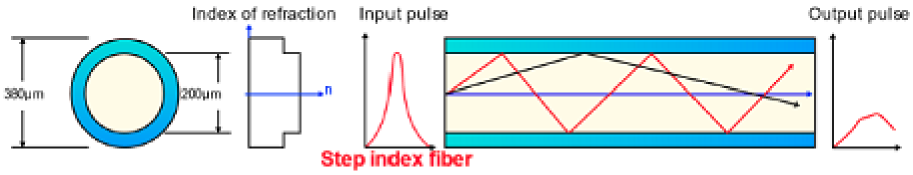
    ⇒ different kind of glass at different layer so that...

    ⇒ speed up the light bouncing in the fiber & slow down the light going straight

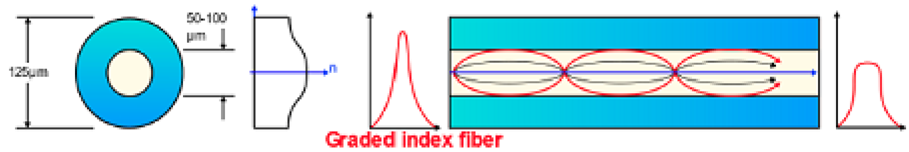
    ⇒ receiver can line up the light more easily



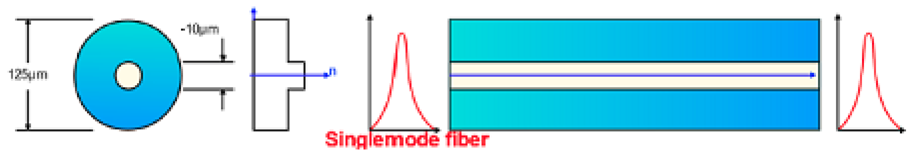
  - Multi-mode vs. Single-mode (each ray = a 'mode')
    1. multimode fiber (MMF) step-index: more bandwidth, significant modal distortion



    2. multimode fiber (MMF) graded-index: a few bandwidth, less modal distortion



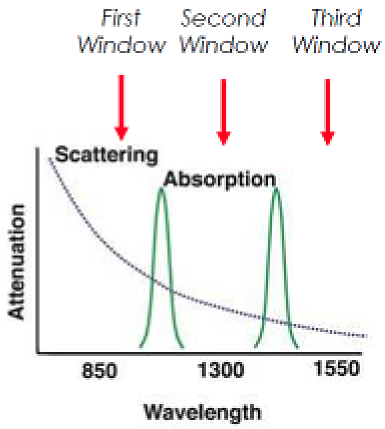
    3. singlemode fiber (SMF): less bandwidth, good at travelling on long distance



    Note: from 1. → 3. the performance increases, so does cost

    does not mix up different cables ⇒ performance suffers

- Fiber Connectors
  - Factors: dust; reflection at the end point
  - Solution: use curved faces at end point ⇒ focusing the light on one point
- Attenuation
  - Scattering: structures + materials in the fiber
  - Absorption: materials in fiber
  - Can be frequency dependent:



- Chromatic dispersion (色散):
  - Factors: refraction index varies with wave length; hard to have a pure single wavelength lazer
  - Solution: Soliton pulses
- Polarization mode dispersion (偏振模色散)
  - Core shape helps

- Setting up Fiber
  - Multi-core Cable Design
    - Factors
      1. individual fibers are fragile ⇒ cable bundles up to 1024
      2. costs the same to deploy one or a bundle of fibers
      3. people wants their own cable for security..., though one fiber can carry whole internet
- Transmission over Light
  - Electronic Data to Optical Signal
    - Keying in optical signal:

      E.g. OOK (on-off keying), QPDM (quatrature polarization division multiplexing)
    - Pulse an LED: cheap, yet broad wavelength range, no nice pulse ⇒ though used in MMF
    - chop a laser: can be small & at a high rate
      1. Cut the light using thin pins ⇒ noise at the edge of square wave (pysical effect)
      2. use the inveretd wave to cancel out the info wave

      ⇒ wavelength tunable on the fly ⇒ used in SMF
  - Speed
    - starts at THz, able to carry the whole internet traffic in on fiber (device can't catch up)
  - Distance
    - Normally...MMF: 1-2 km; SMF: 50-100 km
    - Regenerate/repeat: every 50-100 km

      using expensive optics & electronics for OEO interfaces (optical-electronical-optical)
    - Amplify: every 50-100 km

      using cheap electronics OR optics ⇒ amplifies both signal AND noise
  - Downside:

        - Not easily a shared medium ⇒ point-to-point

          ⇒ crosstalk still exists when sharing, in connectors and within fiber

        - Can use one fiber for RX & TX

          ⇒ need optical splitters at both ends; crosstalk effects

- Last Mile with Fiber
    - Costs
        - existing copper vs. deployment fiber
        - deployment copper vs. deployment fiber
        - maintain copper vs. maintain fiber
    - (G)PON Passive Optical Network
        - Technology used in the fiber part of the backbone, just before the last mile
        - Active network (for comparison)
            1. traffic from backbone splitted by splitters into cabinet depending on their destinations
            2. cabinet starts the last mile, sending only your info to you
        - Passive network
            1. traffic from backbone not splitted, sends all traffic on this fiber to all ends of this fiber (BFS)
            2. use TDM (time division multiplexing) at all fiber ends, to check if this piece of info is yours
            3. potentially RX&TX on the same fiber, using WDM (wavelength division multiplexing)
            4. security may suffer, yet business gains
    - General Approaches:
        - Push fiber as near as can afford / achieve
        - FTTx (Fiber to the x)

          ⇒ FTTP/B/C/N: Fiber to the Premises/ Building/ Curb/ Node

          ⇒ fiber node -> FDU (fiber distribution unit) -> copper cable into house

          Note: the position of FDU differes between 'x'

        - Combine with copper
            1. using DSL
            2. HFC (hybrid fiber coax): share coax copper

               Note: though coax affords 10 Gb/s, yet is shared

               ⇒ fiber node -> trunk coax -> trunk amplifier -> cable into house

               ⇒ peak speed might influenced

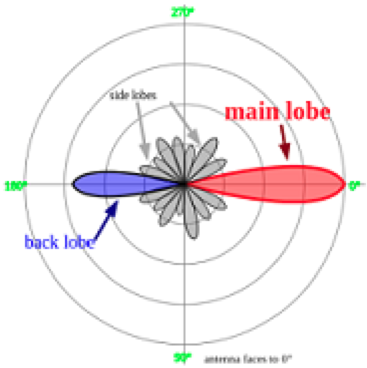
Wireless Communication

- Characteristics
    - Distance
        - can go a very, very long way (satellite transmission)
    - Electronic

        - sensitive to atmospheric conditions and EM interference
    - Unguided transmission
        - on a broadcast & shared medium (free space)
- Noise in Signalling
    - Absorption
        - Gases, dusts
        - Structure & terrains
    - Reflection, Refraction (折射) & Diffraction (衍射)
        - Temperature difference
        - Turbulance
        - Structure and terrains

          $\Rightarrow$ causes multipath reception (multiple delayed refelcted waves interferes)
        - Even varies with time and different wavelengths
    - Noise
        - Extraneous signal in the free space
- Transmission in Wireless: Improvements
    - Transmiter & Receiver $\Rightarrow$ Antennas
        - Omnidirectional (Broadcasting) antenna

          $\Rightarrow$ broadcasting to all direction, yet poor coverage for directly under the antenna
        - Directional antenna $\Rightarrow$ more focus



      Note: generally, $O[n]$ in size, with $n = \text{wavelength}$
    - Clearer Signals
        - More power - shout louder
        - Decrease bitrate - slow down
    - Smarter to Deal with Environment
        - Frequency hopping ($\Leftrightarrow$ channel changing)
            1. detect traffic jam (lost / wrong messages)
            2. ask for re-allocation & try re-association (either actively or after the connection is lost)
            3. re-enter the session with the AP (access point), using the same credential info

               (Note: hopefully the APs reserve the same IP and session for a while)

    - Beam-shaping (directional antennas)
  - Select the Right Wavelength (Frequency) & Power
    - Long wavelength (low frequency):
      1. Go around corner, through walls and waters ⇒ long distance & through obstacles
      2. Low data rate as a trad-off
    - Short wavelength (high frequency)
      1. high data rate
      2. need line of sight (easily blocked)

    ⇒ Consider requirement of point-to-point vs. area coverage; obstacles; effective distance
  - Use the Right (Allowed) Spectrum
    - Spectrum allocation sets the rule of using the shared free space

      (some are reserved for special use, e.g. military use)
    - Channel allocation with each spectrum

      E.g. FM radio (85-108 MHz) in Canberra has 0.8 MHz channel spacing
  - Covering Large Area with Wireless
    - Repeaters
    - Mixed with lined networks (link wireless to wired)
    - Coverage type
      1. fixed vs. mobile client ⇒ directional vs. broadcast
      2. point-to-point vs. cell coverage
    - APs networks (mobile + cell coverage) ⇒ cell handovers
      1. negotiate with current APs to re-association (while the connection is still okay)
      2. APs aware the re-association - keep the same IP & session
      3. enter the same session with credential info
    - Space wireless
      1. Forms: satellite to satellite; satellite to/from ground
      2. Handing over needed: satellite orbits
      3. Potentially high delay: long distance
    - Wireless between earth and space (e.g. Google balloon)
      1. Pros: stable-ish location with greater coverage
      2. Cons: power & maintaining
    - Longe range wireless: MIMO (multiple input multiple output, for 5G), MUSA-MIMO

Link Layer

- Focus & Role
  - Message - Frame
    - various length:
      1. length specified in the frame
      2. start & end of the content denoted

        - targeted messages:
            1. destination address
            2. source address

LANs

- Definition:
    - LAN: local area network
    - WAN: wide area network
    - PAN: personal area network

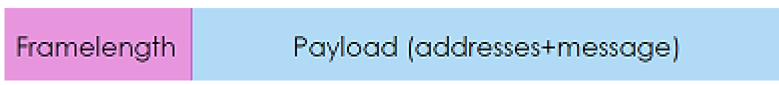
  ⇒ Start of any kind comminication
- Design Principles:
    - Simple
        - no guaranteed message delivery, correction or other specialized fedtures (real-time or etc.)

          ⇒ left to the software
        - focusing on transmitting one message from A to B
    - Efficient
        - multiplexing
- Multiplex in LAN
    - Fair Multiplexing vs. Statistical Multiplexing
        - not everyone talking at the same time
        - no one always spamming - don't need all the bandwidth or all the time
        - demand on capacity varies with time

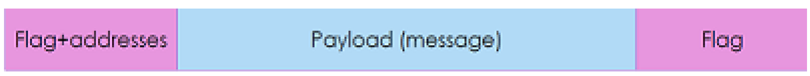
      ⇒ Statistical multiplexing reduces capacity waste

      (more time / channels / wires for the current users)

      e.g. random back-off on collision
    - Fair Access to Network
        - rules for trying to send
    - Example Designs
        - Simple Frame: need to be in synchronization

        - Frame with Flag: need an escape symbol to distinguish (e.g. the "\" to denote "\n" in C)

- MAC (Media Access Control) & Sharing the Media
    - Address Scheme
        - hardwired to the network **interface**
    - Access Scheme - Randomized Access on Shared Media
        - send and then detect

1. send the frame
2. detect collision on the wire
3. wait for acknowledgement
4. on collision or no acknowledgement $\Rightarrow$ back-off for a random time & re-send

Pros: simple, effective in low traffic networks

Cons: actual performance depends on back-off scheme, not scalable

- Carrier-Sense Multiple Access / Collection Detection (CSMA/CD)

  1. sense for carrier till no collision
  2. send frame
  3. detect potential collision - because transmit on wires takes time

     $\Rightarrow$ upper limit time for any potential collision to occur (bounded by wire length)

     $\Rightarrow$ can have a minimum frame size (need to wait for collision detection anyway)

  4. back-off for a random time in collision detected

  Pros: good for wired network

  Cons: not working in wireless

- Carrier-Sense Multiple Access / Collection Avoidance (CSMA/CA)

  1. sense for carrier till no collision
  2. wait for a random time $\Rightarrow$ reduce the possibility of sending frame at the same time
  3. send frame
  4. detect collision & re-try on detected

  Pros: better for wired network as wait before send

  Cons: not working in wireless either

- Back-off Scheme
  - Limitation: not too short & not too long
  - Ideal back-off time: depends on the number of devices in the LAN
  - Approximating the ideal time: Binary Exponential Back-off (BEB):

    1. counting the detected collision in a relatively recent history
    2. for the n^{th} collision, wait for a random number between $[0, 2^n - 1]$

  $\Rightarrow$ statistical multiplexing

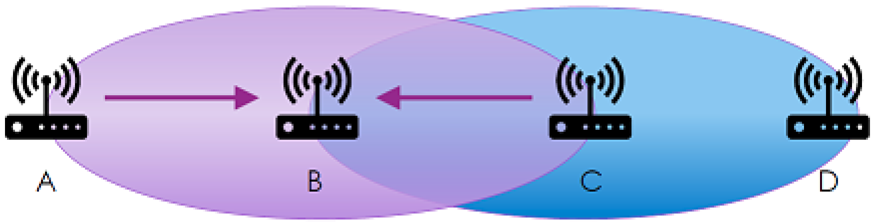
- Access Scheme - Wireless
  - Problem of wireless environment

    1. cannot detect the whole network from a corner

       $\Rightarrow$ because of limited coverage of each cell

       $\Rightarrow$ different Tx can transmit to one Rx without noticing interference

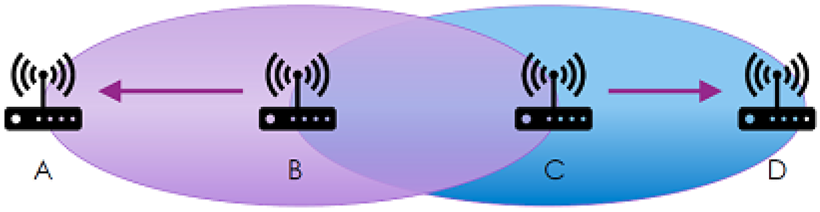
       $\Rightarrow$ hidden terminals: A, C are hidden from each other and can talk to B at the same time



2. local Tx (e.g. its own Tx or nearby Tx) are detected as collision

  ⇒ wasting bandwidth due to frequency hopping / back-up scheme

  ⇒ exposed terminal: C detects collision because of B talking to A

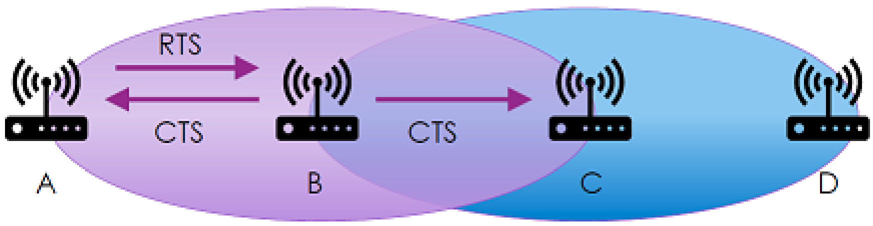


- Multiple Access Collision Avoidance (MACA) - handshake before yelling

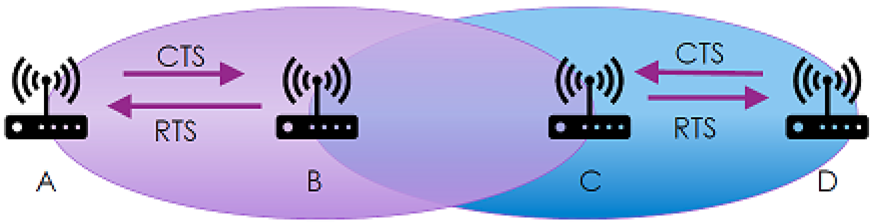
  1. sender: request to send (RTS), providing the frame length N
  2. anyone hears RTS stay silent for receiver's CTS
  3. receiver: clear to send (CTS), providing frame length N
  4. sender transmits the frame & everyone hears CTS stay silent for N

  Pros: now, the receiver decides the collision instead of the sender itself

  ⇒ fixing hidden terminals problem: C knows A is sending after CTS



  ⇒ fixing exposed terminal problem: B, C not influenced by others' CTS



- Access Scheme - Contention-free access

  - Token rings

    1. generates tokens rings (special frame)
    2. pass the token along the rings, under the path-selecting scheme
    3. only talk when token at hands

    Pros: time multiplexing ⇒ guaranteed no contention

    Cons: token may lost & hard to detect and re-generate & not adaptive to topologies change

    ⇒ fragile to error & not scalable

- Topologies

- Bus topologies
    - needs repeater if too long
    - too much collision if many devices

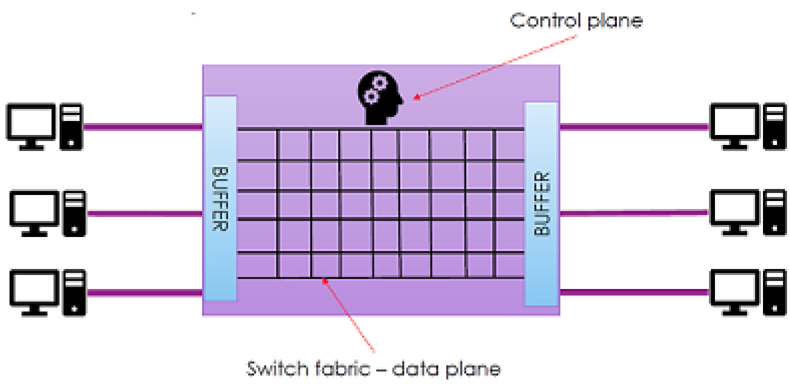
  ⇒ does NOT scale

- Switch
    - a device sitting in the center to learn the source / destination addresses from traffic
    - makes every link point-to-point: source -> switch -> destination

  ⇒ more scalable

  yet, people may employ policy on switch (slowing down the traffic)

- Different LANs design
    - General LAN (customer level)
        - bluetooth, 4G, Ethernet standards ...
    - Carrier-grade LAN (service level - guaranteed performance)
        - ATM (Asynchronous Transfer Mode), GPON (Gigabit-capable Passive Optical Networks), ...
    - Data-center LAN (specific for high volumn, short distance)
        - FibberChannel, ...
- Switches



    - Learn the Address on the Air

      (not all devices on the same media now ⇒ forward message to correct place)

        - Recording all source address of incoming message
        - New / Unknown address:
            1. broadcast to find the address & record the address ⇒ may suffer if cyclic
            2. hope that address show up (send incoming message)
    - Cyclic Swtiches Hierarchy
        - Reasons:
            1. spacial multiplexing - more wires
            2. redundancy
            3. short cuts
        - Broadcasting storm: with no global view, leads to recursive broadcasting
        - Spanning tree: disable some path ⇒ reduce to tree architecture

1. everyone think itself as root
2. broadcasting & forward its current info to select a root on set-up (flooding)

   deterministic selecting rules: e.g. lowest address wins
3. select the shortest path from root - using hop count
4. turn off ports not on the tree

Compared to flooding: maintain the reduced topologies instead of the whole map

- Virtual Lan
  - Reasons
    - Separation of traffic : logically separated network on the same infrastructure
      1. protect confidential info
      2. ensure devices in communication are compatible (computer cannot talk to phone)
      3. easy re-configure the LAN Structure on the demand
    - Prioritization of traffic
      1. drop frames accordingly when busy
  - Implementation
    - Tagging the port address into groups on switches
    - Tagging the frames accordingly

LAN - Ethernet

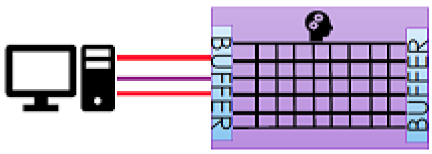
- Advantages:
  - scalablity:
    - plug-and-play
  - backward compatbility
    - negotiate on connection
- Auto-Negotiation
  - Topologies Change - devices connect / disconnect
    - Heartbeat: device sends out a "normal link pulse" to remind the network of itself
  - Capability Negotiation
    - both ends communicate in "fast link pulse", containing requirement of:
      1. Speed
      2. Duplex
      3. Rx & Tx Detection
- Ethernet Frame

Preamble	Start of Frame	MAC dest	MAC src	802.1Q tag [opt]	Type / Length	Payload	Checksum
7 byte	1 byte	6 byte	6 byte	4 byte	2 byte	42-1500 byte	4 byte

  - Preamble:
    - 1-0 bits sequence
    - wake up the receiver & synchronize

- MAC Addressing
  - originally plan to offer globally unique address
  - some address for sepcial use, e.g. "all ones" for broadcasting
  - have special bit for: multi-cast ⇒ send / receive messages from a group
- Tag:
  - virtual lan tags
- Type / Length:
  - different types of frame denoting the purpose

- Bigger Frames
  - Overhead of Ethernet Frame
    - ~30 bytes meta-data / 1500 bytes data ⇒ 3~5% bandwidth lost
    - more bandwidth ⇒ more frames ⇒ more read/write ⇒ traffic jam
  - Jumbo Frames
    - 9000 bytes payload
- Protocol:
  - Listening all frames on the wire until destination is my address
  - Can collect all frames transfering on the wire
- Link Aggregation / Trunking



  - Advantages:
    - performance
    - redundancy
  - Restriction:
    - need to use identical link for each port
    - frames order not changed
    - no partial frame (independent interfaces / devices / netwrok cards at other ends)
  - Protocol:
    - on set-up, checking if using aggregation
  - Model for Aggregation: selecting the path for frames
    - Round-robin: using each path in turns
    - Active back-up: use one path till broken
    - Random: randomly choose

LAN - Wireless LAN (WLAN)

- Interference
  - Dealing noise
    - Adapt power: shout louder

    - Adapt rate: slow down - e.g. 1b/10b (encoding 1 bit into 10 bits)
  - Statistical Multiplexing and Frequency Hopping (⇔ channel changing)
    - choose the frequency to change using statistical random method
  - Beam-Forming and Spaical Multiplexing
    - multiple input multiple output (MIMO) ⇒ multiple antennas for beam-forming
- Channels
  - 2.4GHz
    - most channels overlapped
  - 5GHz
    - larger spectrum space
    - channels does NOT overlap
    - can bind channels into a wider channel ⇒ higher bandwidth for each channel
    - built-in freqeucy hopping in the standard
- Frames in WLAN

X Mb/s *Y Mb/s*

Preamble	Start Frame	PLCP	**Header**	**Payload**	Checksum
7/16 byte	1 byte	6 byte	30 byte	**0-2312 byte**	4 byte

Frame Control	**Duration/ Connection**	**Addr 1**	**Addr 2**	**Addr 3**	**Seq Ctrl**	**Addr 4**
2 byte	2 byte	6 byte	6 byte	6 byte	2 byte	6 byte

  - Preamble
    - wake up the receiver
    - need to negotiate the frequency for EACH frame

      ⇒ to start the negotiation, Preamble is sent under a standard specific speed
  - Frame Control
    - denote the encoding / meaning of the rest of this frame
      1. control frame: control the communication with AP (acess point)

         e.g.request to send (RTS), clear to send (CTS), acknowledgement (ACK), ...
      2. management frame: manage relations with AP (acess point)
      3. data frame: sending the data
  - Reliability
    - detect error & drop frames
    - detect error & fix frames at receiver
    - detect error & sender sends again (WLAN default, as using acknowledgement)
      1. when resending, need to tag the frame as "resending", because acknowledge may lost
      2. may delay the performance because of delay

- Association with AP (Acess Point)
    - need to know:
        1. connection service (service set identifier - SSID)
        2. APs that accept this SSID
    - beacon / probe-request
        1. beacon: AP broadcast
        2. probe-request: client broadcast
    - authentication

      AP: connect to service database to check the ID-key (instead of storing info in local)
    - associate on to AP
        1. resource allocation
        2. re-associate
        3. dis-connect (free resource)

Network Layer

- Focus & Role
    - Message - Packets
        - Definition: fragments of message & smallest unit of data in network layer
        - Reasons: more spacial multiplexing $\Rightarrow$ more parallel
        - Targeted message: address in IPv4/6
    - Traffic Control
        - Optimized routing $\Rightarrow$ no order guaranteed
        - Prioritization
        - Compared with LAN:
            1. LANs focus on simplicity, instead of optimization
            2. Spanning Tree can NOT guarantee optimal topology
    - Scaling Problems
        - Internet accross the world
        - Compatible to diffrent underlying LANs
    - Compatible for Different LANs structure
        - Routing as a layer upon LANs $\Rightarrow$ network layer
    - Adaptive to Change
        - Coping the evolving network topology
    - Simplicity & Best-effort
        - Connection stage stored at ends
        - Minimal service level agreement $\Rightarrow$ no guarantee but best effort

          (reliability provided only where needed)
- Router
    - Forwarding

        - Happens within each router, based on its forwarding table
        - Distributed decision making
    - Routing
        - Happends on the globale level (in routers network)

          ⇒ optimizing routing causes each router optimize its forwarding table
        - Focus on packet ⇒ packets usually arrive in different order than that when it's sent
    - Forwarding Table
        - packet forwarding table
            1. forward packet based on its destination address
            2. more robust to router failure (find another path)
            3. learn / optimize forwarding table on the fly
        - circuit forwarding table
            1. forward packet based on the tag on packet
            2. storring states & policy in networks ⇒ virtual network

               ⇒ separate traffic, guaranteed performance (bandwidth, path, delay...), prioritized routing
            3. overhead of setup / tear down the circuit (resource allocation / cleanup
            4. more guaranteed performance
            5. more fragile ⇒ not able to recover from nodes failure automatically
- Netwrok
    - Routing on Packtes
        - statistical multiplexing - sharing links
        - decision made in destributed routers
        - no guarantee on arrived packets' order (or dependency)
    - Connectionless vs. Connetion-oriented
        - Connectionless ⇒ packet forwarding - network makes all decisions, in each distributed router
        - Connetion-oriented ⇒ circuit forwarding
- Hosts
    - Sending Packets
        - send to local LANs service (switches)
        - switche decides the forwarding direction

          ⇒ router - outer internet

          ⇒ local hosts - in my LANs
    - Hosts Routing Table
        - longest matching prefix + broadcasting address
- Communicating with Link Layer
    - Reason

- link layer deals with only MAC address

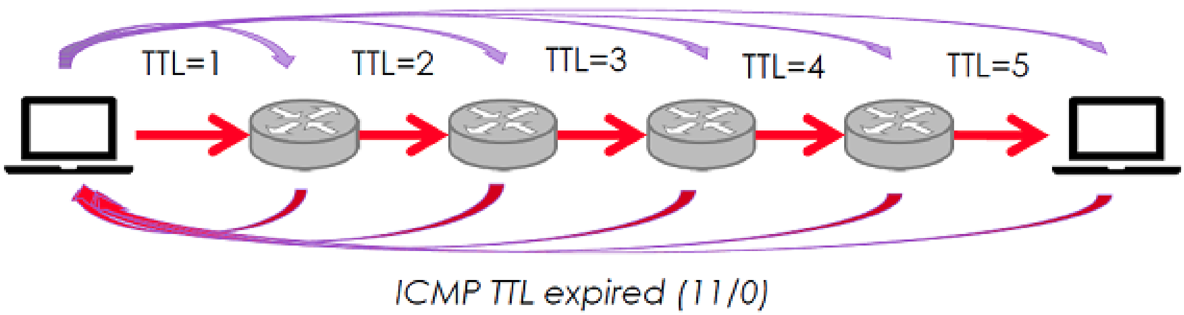
  ⇒ communication across layers (IP ⇔ MAC)

- link layer sends only its frames

  ⇒ inter-changing of IP packet and link frames, especially address

- The Address Resolution Protocol (ARP)
  - source MAC: read from local hardware
  - destination MAC:
    1. sender broadcasts LAN frame to call for corresponding IP address
    2. receiver replies with its MAC address
  - optimizations
    1. caches the MAC address (with time-out)
    2. cache passing IP address (when others broadcast & reply)
    3. upon connection on LAN, broadcast my IP address (MAC address in frame's address field)

- Allocation of Address
  - Consideration
    - globally-unique address
    - address aggregation
  - Authorities
    - allocate regional IP addresses blocks to regional internet registries
    - registries allocate IP addresses blocks to ISP
- Internet Control Message Protocol
  - Aims
    - special packet for router to inform the hosts (usually senders, including routers)
  - Traceroute

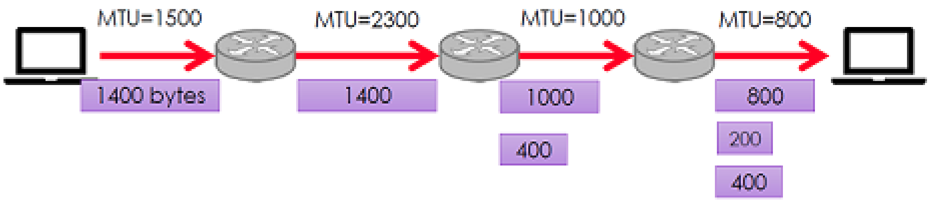


    - Sending message with increamenting TTL (Time to Live - in hops count)

      ⇒ the i^{th} router sends back with corresponding exceptions (via control messge protocol)

  ⇒ host can find out the path its packet is taking

    - TTL: time to live (in hops count)
- Fragmentation in IP
  - Slicing Packets
    - packet bigger than LAN's payload ⇒ sliced packets



- Realizing the need of slicing:

  packet size > MTU (maximum trasmission unit)

- Flags to inform the next router:

  1. Identification field: key to identify an unique packet (which sliced packets share)
  2. Fragment offset: the offset of this packet in the original big packet
  3. MF: more fragment flag ⇒ more fragment of packets after me
  4. DF: don't fragment flag ⇒ no more fragment after me

- Trasmitting sliced packets

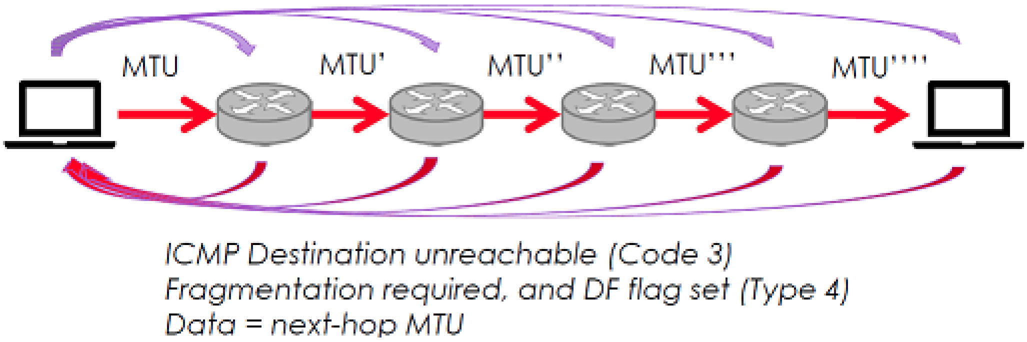
  1. copy IP Header, including identification (each sliced packet belongs to the original packet)
  2. adjust Length, Checksum & TTL (time to live) feilds of each sliced packet
  3. set fragment offset & MF/DF flags
  4. receiver re-assemble accordingly

- Potential problem

  1. more work for routers
  2. more potential internal packet loss
  3. security issue (injection within packet)

- Avoiding Fragment in IP ⇒ Path MTU Discovery

  - Using internet control message protocol - similar to traceroute



⇒ sender send at the lowest MTU

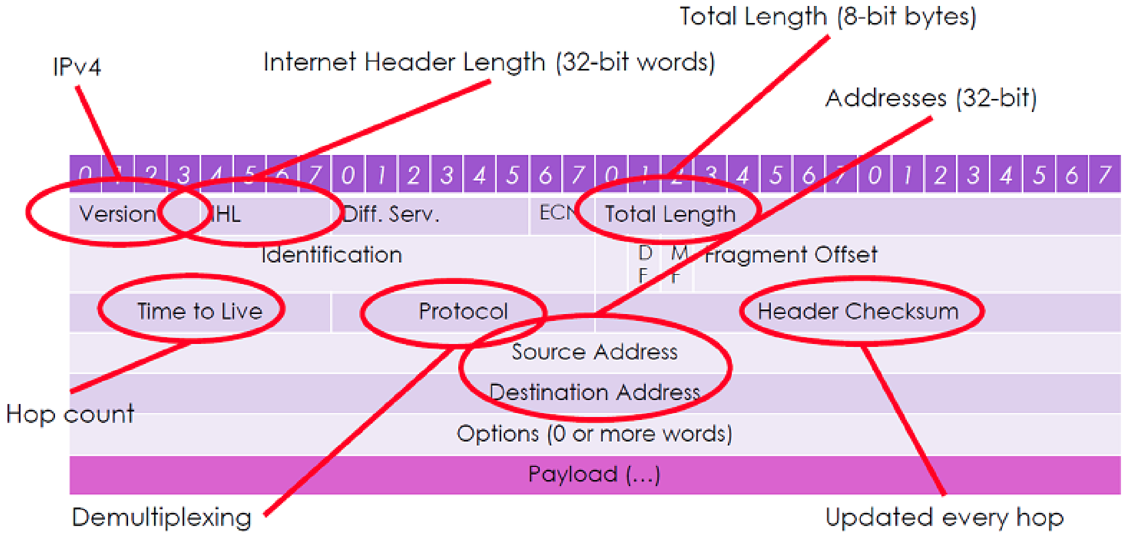
⇒ router focuses on sending packets

- IP multicasting
  - Definition
    - mutilcast to only a group of users, compared to broadcast
  - Challenge
    - sender may not be able to handle thousands of requests & data streams
  - Approach
    - usres subscribes to the sender, using special message / packet

⇒ all routers on the path know the subscription

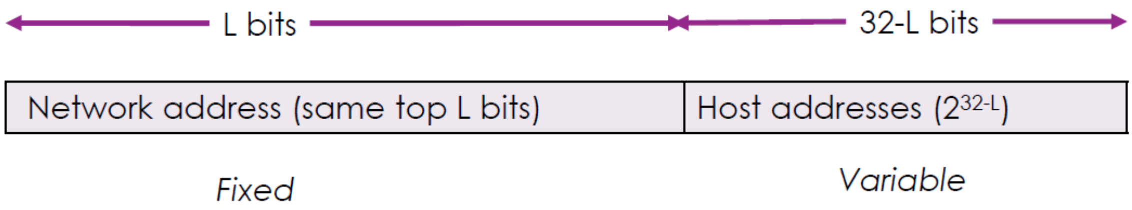
potential problem: states stored in network

- Internet Protocol - IP
  - IP - v4
    - Protocol overview (top-left -> bottom-right)



Note: checksum needs to be update at every router (hop counts changing)

    - Addressing



1. total length - 32 bits
2. prefix denoting a netwrok, containing a range of the address

   ⇒ fewer entry in forwarding table
3. host addresses denoting the subnet under this network (denoted by prefix)
4. "/x" - x bits for host addresses; (32-x) bits for prefix

    - Special addesses:

1. private networks, multicasting, broadcasting, experimental, local interface, ...
2. convention: sub-net wires, sub-net broadcast, local router, ...

    - Classes

1. denoted by first few bits of the address
2. denoting different function of current packet (broadcastin, ...)

    - Forwarding by prefix

1. Prefix:

   network address
2. Assumption:

   addresses aggregated into ranges

3. Benefits:

   aggregation benefit of hierarchical addresssing ⇒ less entries, routing efficiency

   more flexible for directing specific trafic

4. **Forward by longest matching prefix**

   default behavior for shorter (less-specific) prefixes

   specialized behavior for longer (more-specific) prefixes

   ⇒ allow to route sub-chunks (some specific) of address to other hops / routers

   choose the one that match the most ⇒ "best-effort service"

- Addresses exhaust

  1. problem ⇒ no more available addresses & large amount of wasted addresses

  2. current solutions

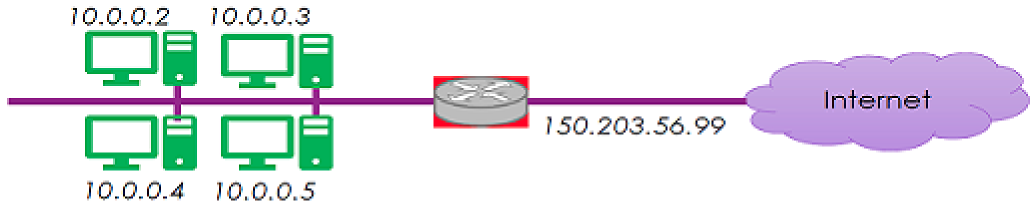
     re-allocating smaller chunk of addresses

     ⇒ addresses aggregation damaged

     ⇒ larger forwading table & more updates

     ⇒ routing efficiency ↓

     NAT (Network Address Translation) - use private address space behind a public IP address



  3. future solution: IPv6

- Potential & exisiting problems

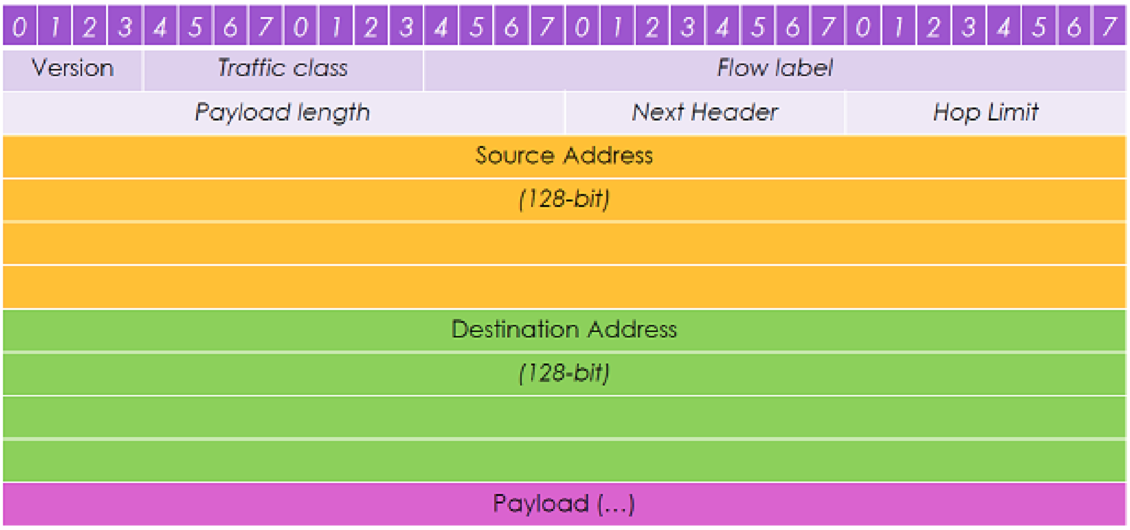
  1. designed in a smaller & more trusting world

     ⇒ lack of security, mobaility and compatibility concern

  2. out of addresses & routing efficiency problem

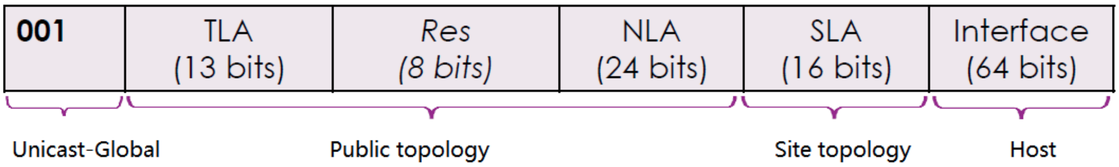
- IP - v6

  - Protocol overview



1. larger address sapce: 128-bits
2. specific meaning in addressing scheme

- Addressing



1. 3-bit header: unicast, multicast or anycast (no broadcast)
2. TLA: top level aggregator - global ISP
3. Res: reserved
4. NLA: next level aggregator - site
5. SLA: site level aggregator - subnet
6. Interface: address in local subnet - host

Advantage: explicit addresses aggregation

- Transferring to IPv6

1. dual stack: routers run both (2 separate pathways)
2. translate: convert IPv4 $\leftrightarrow$ IPv6
3. tunneling: pack IPv6 packet inside IPv4 packet until a router recognize IPv6 (v4 everywhere)

Transport Layer

- Focus & Role
  - Message - Segment
    - Definition & components:
      1. functionality & quality (including reliability) for applications
      2. host-to-host & port-to-port message, for application use
  - Main Services
    - Reliability
    - Communication between hosts (on their ports)
  - Port and IP addresses

        - IP address for host
        - Port for applications ⇒ port binding:
            1. port allocated on memory;
            2. client connects to an exposed port;
            3. server maintain the concurrency from inside
    - Service Types
        - Reliability:
            1. reliable: segments loss repaired at transport layer
            2. unreliable: reliability offload to applications
        - Communication froms:
            1. messages: self-contained command and response
            2. byte-stream: generic flow of bytes, chunked into segments

    ⇒

	Unreliable	Reliable
Messages	UDP (datagrams)	
Byte-stream		TCP (Streams)

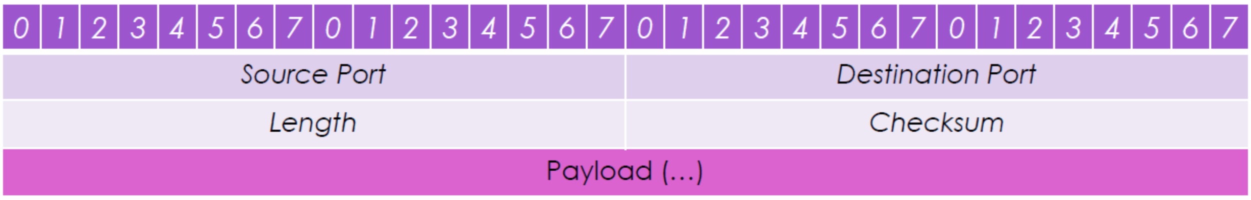
- TCP vs. UDP
    - Comparison
        - TCP: many features, able to negotiate
            1. large content transfer
            2. longer, more complex sessions
            3. reliable
        - UDP: enhanced packet
            1. short messages, light server touch
            2. simple request-response transaction
            3. unreliable, yet compensated by ARQ (Automatic Repeat Request)

TCP	UDP
Connection-oriented *(significant state in transport layer)*	Connectionless *(minimal state in transport layer)*
Delivers BYTES: once, reliably, in order	Delivers MESSAGES: 0-n times, any order
Any number of bytes	Fixed message size
Flow control (sender/receiver negotiate)	Don't care
Congestion control (sender/network negotiate)	Don't care

    - Situation for UDP - multicasting
        - connectionless
        - Replicate segments or packets are fine

- Missing (some) segments or packets are fine
- Sending Byte-stream - TCP
  - Chunks of byte-stream in segments - message boundaries not reserved
  - Read / write on buffer

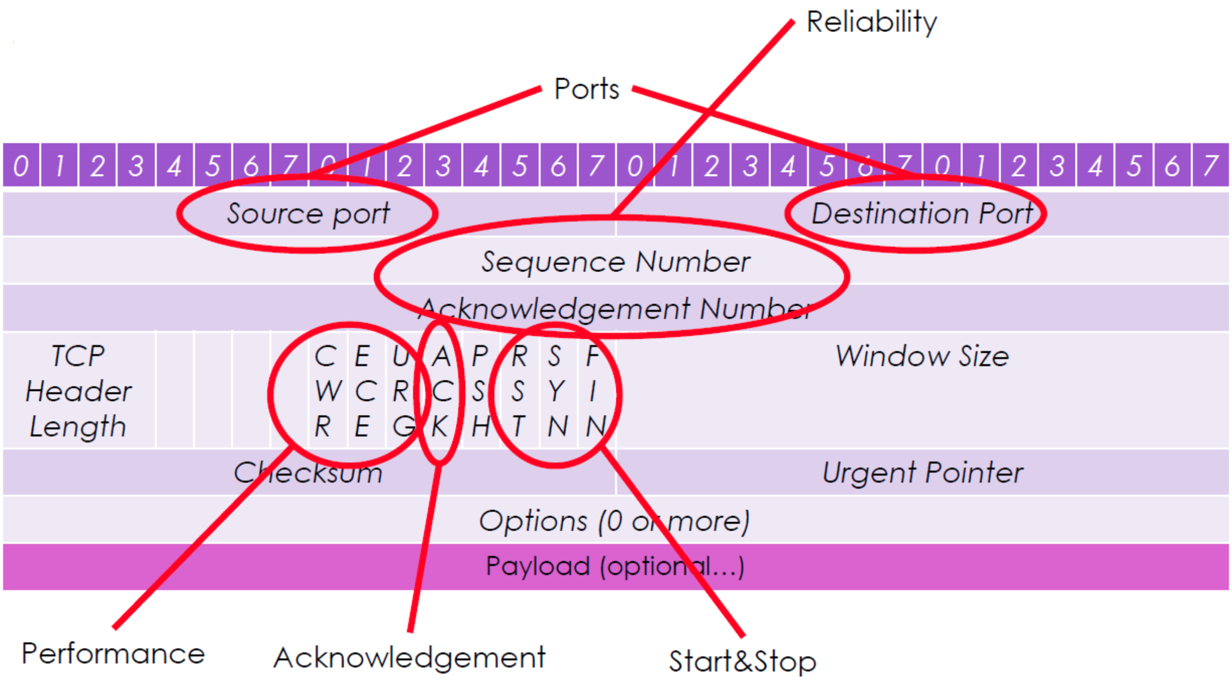
- UDP Segment



  - Ports
    - associate segments with applications / sessions

      note: application can use multiple ports

- TCP Segment



  - Options
    - Ports
    - Maximum segment size
    - Window scale - for selective repeat
    - Time stamp
    - Selective acknowledgement - advanced acknowledgement
  - Reliability
    - Important components: sequence number, acknowledegement
    - Sequence number: byte count in a stream (in $\bmod n$ space) $\Rightarrow$ can be used as relative time stamp

      Note: does NOT start from 0 $\Rightarrow$ security reason
    - Acknowledgements: with sliding windows of size w & selective repeat

$\Rightarrow$ more parallel (size w depends on bandwidth and delay)

1. sender

   allows w segments to be outstanding (no ACK provided) - sliding window

   a timer for every unACKed segment - re-send after time-out

2. receiver

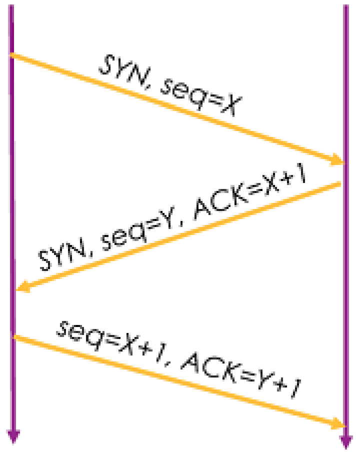
   buffers many segments $\Rightarrow$

   ACK received segments

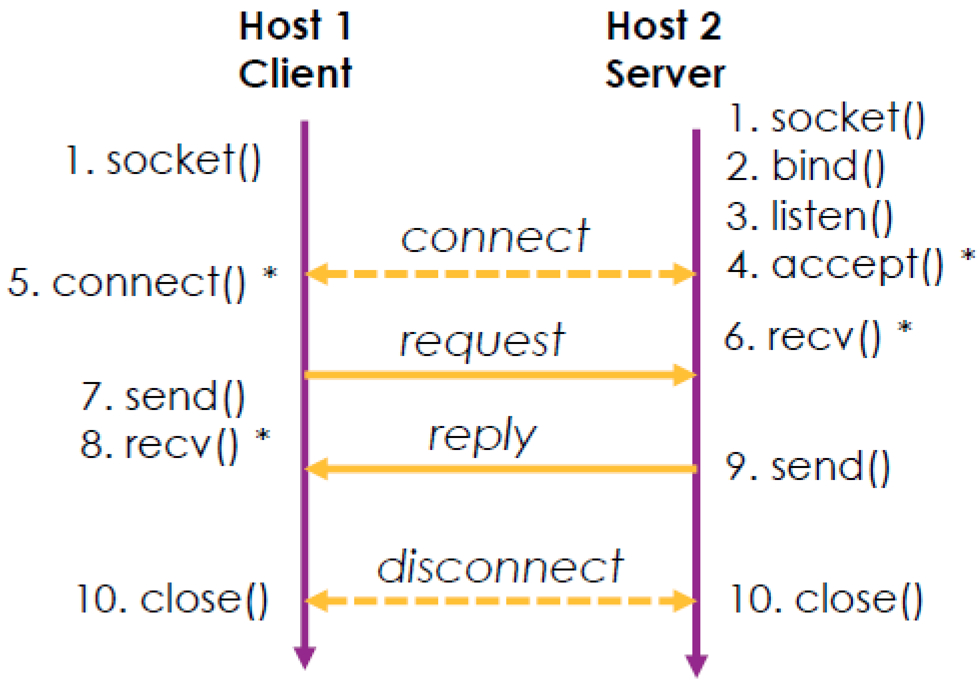
   request missing segments - which is in the gap of segments stream & the future

3. Pros: no need to suspend on every segment $\Rightarrow$ more parallel

- Connection in TCP - 3 Way Handshakes
  - Reasons:
    1. TCP is full-duplex $\Rightarrow$ need to connect two independent paths for each direction
    2. need to start together $\Rightarrow$ negotiate synchronization & initial Seq (sequence number)
  - Procedure:



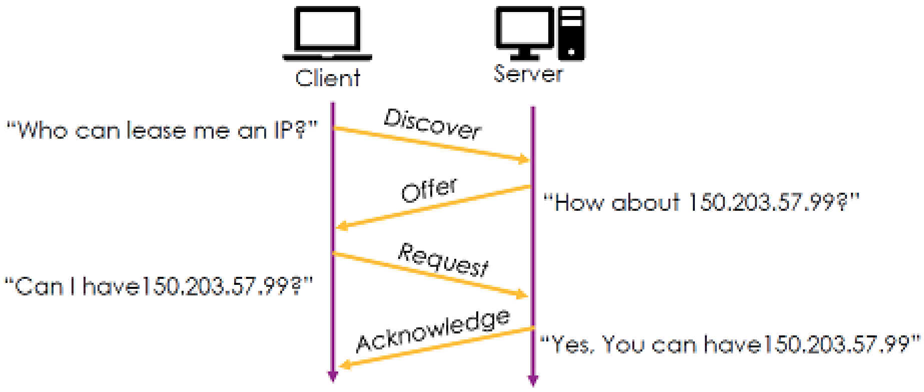
- Connection between Applications
  - Necessary components:
    - source & destination IP addresses
    - source & destination ports
    - transport layer protocol (TCP/UDP)
  - Socket API:



        - Note: * denotes potential blocking calls
    - NAPT (Network Address and Port Translation)
        - NAT: hiding behind a public IP address
        - NAPT: hiding behind a public IP + translate outbound port into host's actual port

Application Layer

- Focus & Goal
    - Build Sessions
        - Sessions: a series of interactions
        - based on TCP reliable byte-stream or UDP unreliable messages, or combination / extension ...
    - Presentation of Content
        - Interpret content: interpret message/byte-stream inside TCP/UDP segments' payload
        - Handle Command: handle request & control from both ends
    - UDP-based Application
        - Short messages ⇒ light server touch ⇒ simple request-response transaction
    - TCP-based Application
        - Large content change ⇒ longer & complex sessions
- Dynamic Host Configuration Protocol (DHCP) - Getting IP Address
    - Goals:
        - allocate IP address
        - automatic configuration, instead of manual
    - Negotiation procedure



- need to broadcast to discover
- need to request the IP address after being offered

  (backward compatibility)
- can directly request the IP address in hand, when it is close to expired

- DHCP relays (转接):
  - DHCP server in the middle;
  - Relays on router/switches...
  - Relays forward the request to the server (broadcst → unicast)
- Multiple DHCP
  - Reasons: performance, robustness
- DHCP service with orther services
  - on the default router (to the internet)
  - with DNS service

- Domain Name Service (DNS)
  - Goal & Reasons
    - Changing IP address
      1. IP address expired
      2. LANs re-configure
      3. physical movement to other places
    - Human readable name
  - Definition
    - Names - for humans
    - Addresses - for underlying protocols and applications
    - Resolution - finding the right servers (authority) to find the requested IP - names

    Note: 1 name can have multiple addresses; 1 addresses can have multiple names
  - Design
    - Distributed control
    - Hierarchical namespace - delegate to authorities (for them to be responsible for, legally)
    - Automated protocol & handling
  - Namespace
    - Root: '.' (usually dropped)

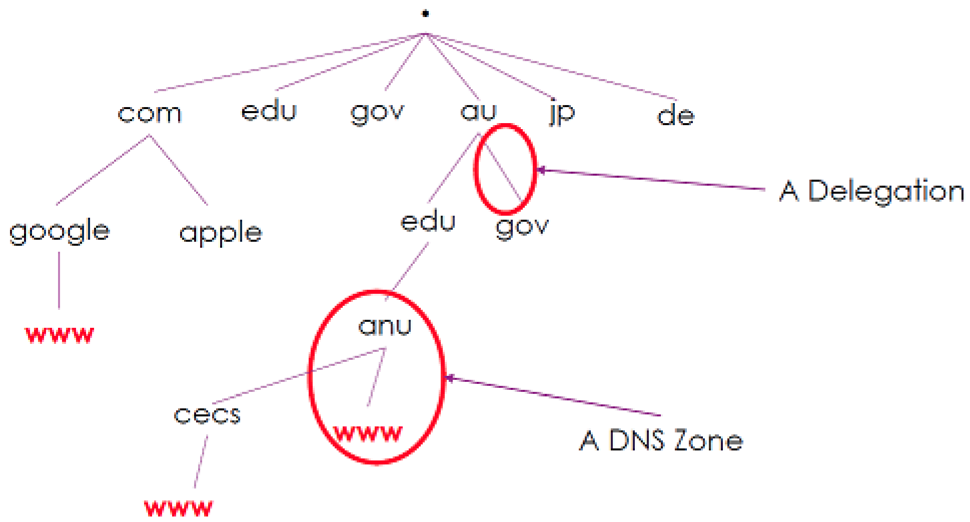
- Top level domain (TLD):
    1. classification - com, edu, org, net
    2. contry code - au, us, jp, cn, ...
- running down the hierarchical to the local host

- Domain
    - Delegated to authorities; authorities hold legal responsibility
    - Responsibility covers the subtree starting form the delegated point

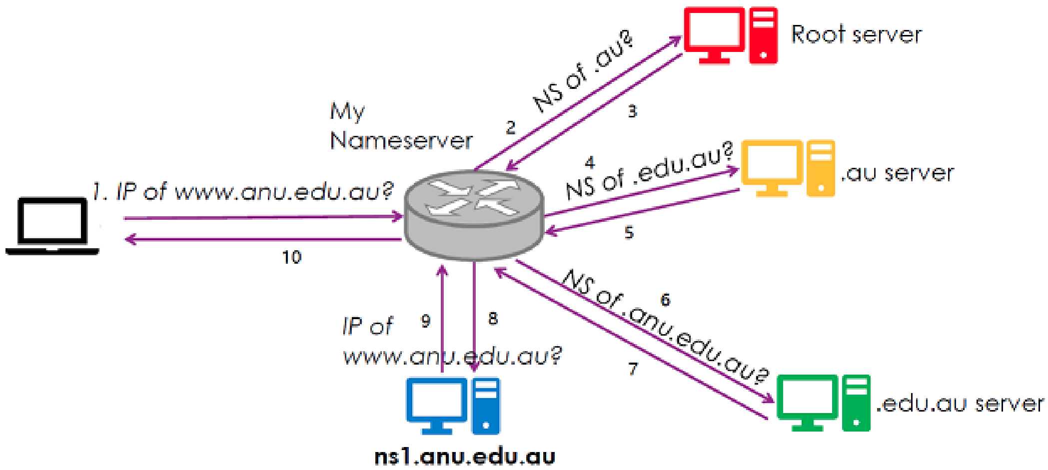
- Zone
    - Shared pieces of DNS database - through technology
    - Recording the information about:
        1. meta data of the the zone
        2. further relations (delegations)
        3. resource records: addresses and corresponding DNS names, services & other meta data

           (includes time stamps - used for cache)

- Example



- Resolving the IP address (Resolution)
    - Iterative query to resolve IP address
        1. example:



2. Improvement - Caching

   ⇒ cache information with an expired time

   ⇒ dirrectly knows the authorities / IP address for the next query (within a time) - shortcut

- Nameserver, name & IP addresses replication:

  1. ask more than one server ⇒ spread the workload & risks
  2. more IP ⇒ more hosts, prevent nodes fail-over
  3. more names ⇒ less typo

  Note: enable prioritization, with other addresses

- Security issues

  can change the cache in router - man-in-the-middle

  1. make a query by yourself;
  2. draft a reply, including a valid source address & guess the query's ID (enumeration will do);
  3. (router usually check only destination addr, matching ID, is it answering query)

- More security: signature, public-private keys, ...

  ⇒ validate along the way ⇒ building a trust chain

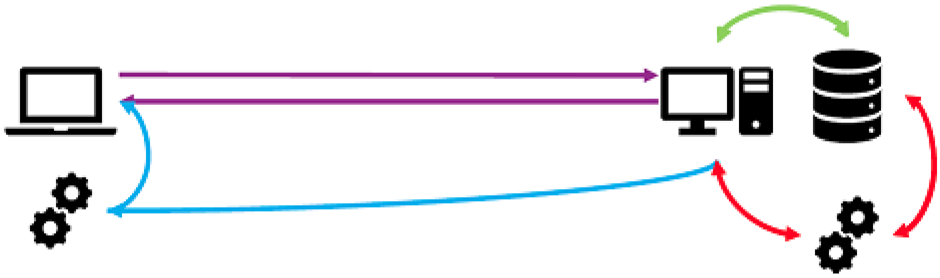
- Policy: can change the query on the fly ⇒ enable policy, e.g. content check, DNS polluting

- Dynamic DNS - NAPT

  1. register a DNS name on DNS server for my host ⇒ server handle public IP address change
  2. host address change: local IP address (private address) change ⇒ router port change
  3. message sent to me will find the right port through DNS (DNS will query my router)

- HTTP Protocol - Most Common Used Web Application
  - Focus
    - Deliver associated content
    - Linking semantic related content on the web, instead of fetching everything to local
    - Light weight (initially designed) - used with UDP together as a transport layer sometimes
  - URI vs. URL:
    - URI - uniform resource identifiers: identifier, identifying anything

- URL - uniform resource locator: an example of URI ⇒ location on internet
- URLs - Schemes
  - Various schemes
    - server name in DNS, IP addresses, HTTP protocol, resource on the host, query to server
- Dynamic & Static Contents
  - Procedure



purple: communication & data between server and client

red & green: interpret & execute command on server side

blue: get input / command and execute program on the local - security concerns

1. parse URL & resolve DNS
2. connect to the host
3. make HTTP request
4. receiver contents
5. close TCP/UDP connection & render the page

- HTTP Request
  - Basic Command
    1. GET: get resource
    2. HEAD: get the headers about the resource (meta-data)

       enable backward compatibility

       enable re-direction - for performance reasons (e.g. redirect to closer server)
    3. POST: append my contribution to the host
  - Feedback
    1. 1xx - not used currently;
    2. 2xx - OK (successful); 3xx - redirection (no longer in this address)
    3. 4xx - hosts' problem, e.g. 403 bad request; 5xx - server has problem
- States in HTTP
  - Default
    1. Stateless - server should not hold state (too much - each session needs a key)
    2. Stand along query
  - Session Key in URL
    1. Key: encoded in URL
    2. Passing the Key: as part of query ⇒ server interprets it from the query
  - Session Key in Cookies

1. Key in cookies: set & offered by server; held by client
2. Passing the Key: include the cookies in the communication with server, if relevant
3. Type of Cookies

    session cookies - deleted when page closed

    persisten cookies - static cookies with expire time

- Efficiency in HTTP
    - Parallelism

        use idle bandwidth & potential distributed servers

    - Persistence

        use one TCP connect for all requests

        need only one connection set-up (3-way handshakes)

    - Pipeline

        send all requests and wait for all resource from server (full-duplex)

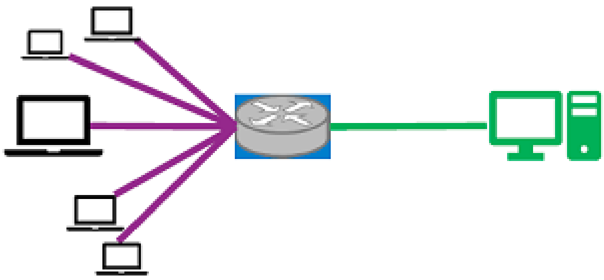
    - Caching
        1. Browser (local) level

            cache on demand (cache the most popular one)

        2. Proxy level - proxy cache ($\Leftrightarrow$ caching proxy)

            cache on routers / local LAN - closer to client

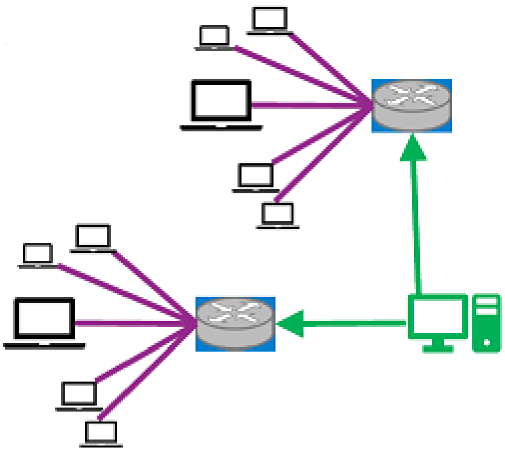
            share the cache in the LAN - enable security check & policies



        - Network server level - Content Distribution Network (CDN)
            1. distributed file system over internet - DNS redirects client to the best cache

                $\Rightarrow$ server offload; content closer to clients

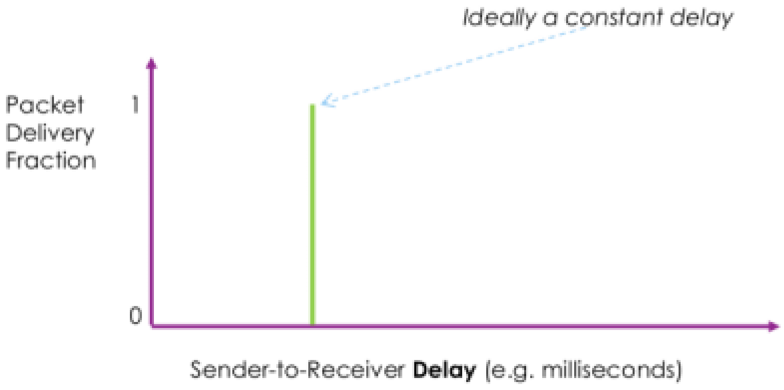
            2. cache before the request $\Rightarrow$ as a way of sharing distributed servers



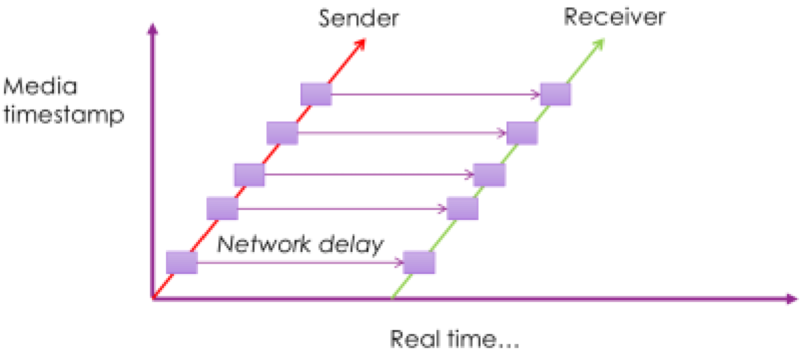
Real-Time Network

- Goal & Real-Time Definition
  - Task-specific Requirement
    - predictability, regarding both timing and result
    - logical correctness
  - Trade-off (in network environment)
    - timeliness (delay) vs. reliability (loss)
    - general option:
      1. TCP with CDN (content distribution network) $\Rightarrow$ more overhead (delay), more reliable
      2. UDP with interactive control $\Rightarrow$ low overhead (delay), less reliable
- Challenge:
  - Dynamic Network Environment
    - best-effort service (by default)
      1. variable delay
      2. loss

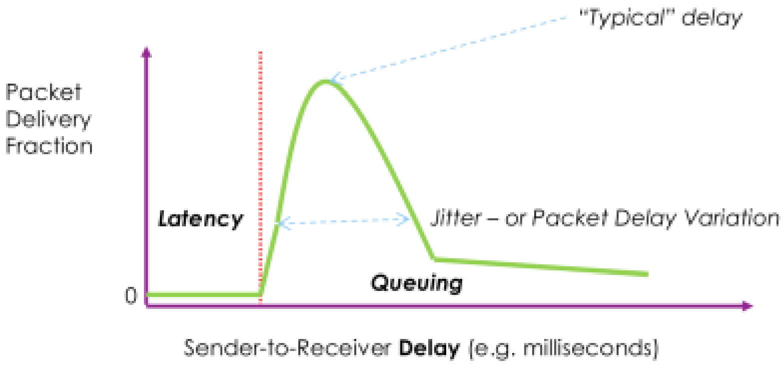
    $\Rightarrow$ NOT predictable, regarding both timeliness and quality
  - Limited Choice
    - Internet is not the best choice, but maybe the only choice
- Network Delay
  - Ideal Dealy
    - Distribution :



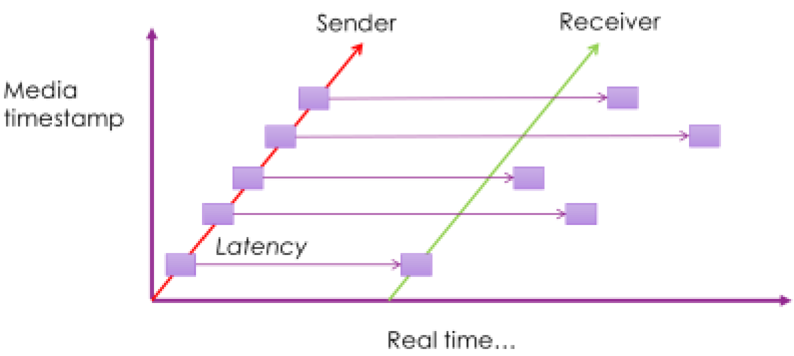
- Delay in real-time sequence:



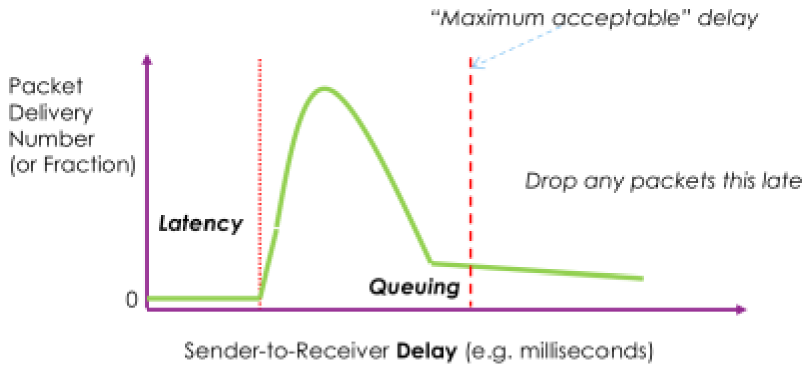
- Realistic Delay
  - Distribution:



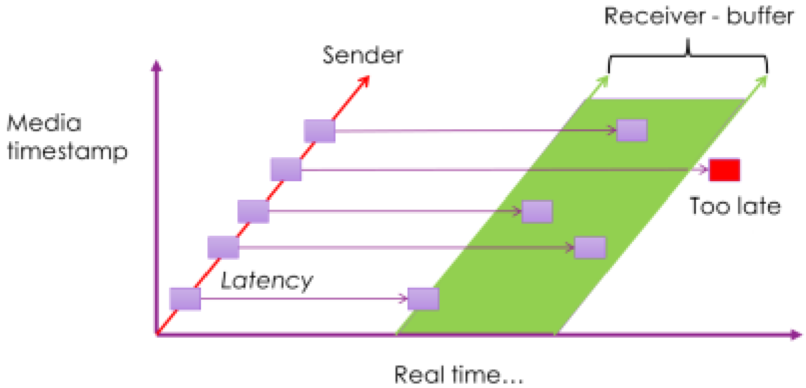
  - Delay in real-time sequence:



- Manage Delay
  - Buffer: Timimg requirement with hard-deadline
    1. in distribution



2. in real-time sequence



3. buffer size trade-off

   big buffer $\Rightarrow$ more tolerant to jitter, greater delay

   small buffer $\Rightarrow$ less tolerant to jitter, less delay

- Retransmission

  1. overhead:

     transmission of request or time-out ACK round-trip time

     packet transmission time

     queuing delay

     more buffer

  2. caching: router / other clients re-transmit

     e.g. in multicast, listener may re-transmit, insdeat of the sender

- Elastic buffer: adapting buffer

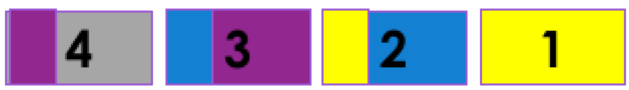
  1. stretch if delay is heacy
  2. shrink in good network condition

- Error correction

  1. enable interpolation between packet

  2. adaptive forward error correction: sending a compressed back-up of previous packet

     e.g.



     , where colour denotes content in packet

- Parallel-transmission:

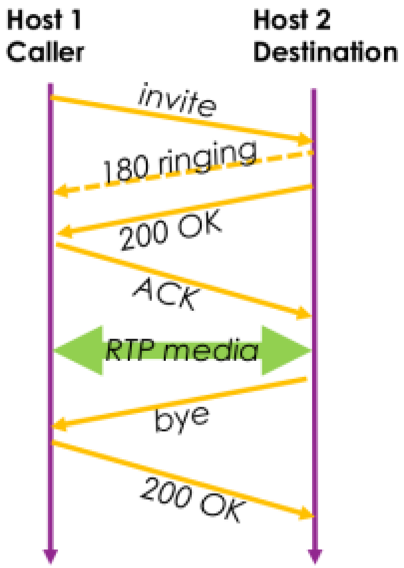
  several copies in different quality (normal, low, lower, ...) ⇒ redundancy

  (better than nothing)

- Real-time Transport Protocol (RTP) - Application Layer
  - Message

<table>
<tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td></tr>
<tr><td colspan="2">Ver.</td><td>P</td><td>X</td><td colspan="4">CSRC. Count</td><td>M</td><td colspan="7">Payload Type</td><td colspan="16">Sequence Number</td></tr>
<tr><td colspan="32">Timestamp</td></tr>
<tr><td colspan="32">Synchronisation Source Identifier</td></tr>
<tr><td colspan="32">Contributing Source Identifier (*N)</td></tr>
<tr><td colspan="32">Profile Headers, Extension Headers, Payload (...)</td></tr>
</table>

    - usually runs over UDP (sometimes TCP)
    - reliability: sequency number
    - timing: timestamp
    - multiple sources sync/async: contributing source Id, sync source Id
- RTP Control Protocol (RTCP)
  - Bi-directional Communication
    - out-of-band (on different port than RTP)
    - negotiation between senders & receivers (report their status)
  - Monitoring & Assisting
    - connection status: heartbeat, connection/disconnection
    - others: measure distance with rount-trip-time, can be used for re-transmission request
- Real-time Session - Video Conference
  - Network Structure
    - many to many ⇒ multiple sites, multiple media streams
  - Procedure
    - establish calls: Session Initiation Protocol (SIP)
    - negotiate details: Session Description Protocol
    - delivery media: RTP + RTCP
    - playout content: bufferes & applications
- Session Initiation Protocol (SIP)
  - Ability
    - establish and tear down calls - not noly for video
  - Signalling - over TCP and UDP



- Real Time Streaming Protocol
  - Ability
    - manage one-way media transmission ⇒ more tolerant to delay (content already generated)
    - establishes a streaming session and negotiates media transport
- Other Alternative Protocols:
  - HTTP
    - reason: get through firewall, many extensions
    - HTML5 with video player built-in

Internet of Thing (IoT)

- Goal & Definition
  - Connect Independent Devices
    - ranging from small (sensors, controllers) to large (house, vehicles)
- Challenge & Requirement & Principle
  - Scale:
    - large number

      ⇒ no limit on devices (addresses) and relationships (connections), but on messages

      (to avoid swamping networks)
  - Power
    - small device with limited power

      ⇒ do smart thing elsewhere; focus on minimal power needs (e.g. RF, solar etc.)
  - Networking
    - low power with remote location and widely distributed

      ⇒ more efficient transmission (limit its need), assisted by neighbour (mesh networks)
- Timeliness
  - real-time appllication

⇒ short message, distributed quickly

- Robusteness - Reliability
  - reliable function with limited aid

    ⇒ add on demand, as lightweight as possible
- Design
  - Publication - Subscription
    - separate messages publishing and consuming

      ⇒ allow for any type/number of sources&consumers

      ⇒ interface between them
  - Broker (server)
    - lightweight, flexible, open
    - a buffer for intercahnge
    - interface for consumers & sources
- Message Queueing Telemetry Transport (MQTT)
  - Publication-Subscription Design
    - temporal "database" for key-value pair (which deletes data as fast as it can)
    - usually over TCP
  - Topics
    - denoting the clients of a message

      ⇒ broker multicasts message to clients subsribing the topic
    - notation of topic
      1. / for topic level separator (subtopic, ...)
      2. wildcards: # for all subsequent topics; + for any possible topic for one level

         E.g. Sensors/# ⇒ all sensors, anywhere
    - special topic: $SYS/# ⇒ system information of the broker
  - Sources
    - publish messages under your topics

      ⇒ build your own database (denoted by topics and their sub-topics), with your own data-type
  - Consumers
    - subsrcibe to particular topics

      ⇒ topic may not currently exist (has no message for it)

      (apparently, need to agree on the message type with publisher beforehand)
  - Message

<table>
<tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>6</td><td>7</td></tr>
<tr><td colspan="4">Message Type</td><td>Dup</td><td colspan="2">QoS</td><td>Rtn</td><td colspan="8">Remaining Length</td><td colspan="16">Variable (optional) header(s)</td></tr>
<tr><td colspan="32">Payload</td></tr>
</table>

    - Message type: 16 in total, CONNECT, DISCONNECT, ACK, ...
    - QoS: quality of servuce, of (0,1,2)
    - Dup: denotes if the message is duplicate
  - MQTT Service over QoS Level
    - level 0 (default) - at most once
      1. sender pushes message, and then discards the message
      2. receiver either receives or not (messages dropped on the way)
    - level 1 - at least once
      1. sender stores application message and assigns an ID to it
      2. sender sends message and waits for PUBACK (Publish Ack)
      3. receiver receives and responds with PUBACK (with the ID)
      4. sender discards message

      Note: sender resends the message after a timeout, till it receives PUBACK
    - level 2 - exactly once
      1. sender stores application message and assigns an ID to it
      2. sender sends message and waits for PUBREC (Publish Received)
      3. receiver receives and responds with PUBREC, waiting for PUBREL (Publish Release)
      4. sender responds with PUBREL, waiting for PUBCOMP (Publish Complete)
      5. receiver responds with PUBCOMP
      6. sender discards message

      Note: the protocol (either sender or receiver) re-send current message after a timeout

      PUBREC: receiver ack that it receives application message

      PUBREL: sender ack that it knows the receival, stop sending application message

      PUBCOMP: receiver ack that it knows the stop(ensure PUBREL received)

      PUBREL, PUBCOMP: wrap around QoS level 1

      $\Rightarrow$ ensure application message presented only once (after the protocol ends)
  - States in MQTT
    - retained message:
      1. report the default/latest status when receiver subsribe on the topic
      2. only one retained message allowed per topic
    - last will message
      1. provided to broker at connection / when alive (a normal message with flag)
      2. pushed to topic when ungracefully disconntect detected (no new message after a timeout)
    - clean session
      1. flagged on connection

2. not clean $\Rightarrow$ broker keeps all messages (QoS 1, 2) for you, when you dropped off

   \Rightarrow huge burden on broker

3. clean $\Rightarrow$ treated as brand new client

- Security
  - username-password, client ID, etc...
  - certificates, signature, encrypted connection, etc...

- MQTT Use Case - Smart Home

- Sensors
  - each publish to a state topic, regularly

- Controllable Devices
  - each subsribes to one or more state / command topic(s)

- Controllers
  - each publish to one or more command topic(s)

$\Rightarrow$ devices not directly controlled by controllers

- $\Rightarrow$ enable more Flexibility:
  - rule machine: given X (is published), do Y
  - state machine: combine rules, store states, note changes

Routing in Real World

- Overview
  - Routing
    - unicast routing $\Rightarrow$ multicast and etc. are based on this
    - distinction in forwarding vs. routing
    - global scale (beyond ARP)
  - Timescales of Activities (regarding routing)

What you're doing	How quickly	Because
Forwarding/ "load-sensitive" routing	Seconds	Bursts, Congestion
Routing	Minutes	Changes, failures
Traffic Engineering	Hours	Long-term load
Provisioning	Months	Customer demand

  - Expectations

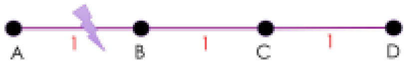
Expect	Because
Correctness	It has to get packets from A to B
Efficiency	Use available bandwidth well
Fairness	Don't ignore capable network elements
Convergence	Recover quickly from any disturbances
Scalability	Copes with increasingly large and complex networks

- Ideal Routing
  - de-centralized ⇒ distributed system
  - alike nodes ⇒ use same protocol, run same codes at same time
  - learning ability ⇒ learn through traffic & messages exchanges with neighbours
  - robust ⇒ deal with router, link, message failures

- Best Route
  - Various Measures for "Distance"
    - latency (delay)
    - bandwidth (speed)
    - cost (money)
    - hops count (forwarding)
  - Assumption
    - ignores link congestion
    - ignores router load
- Sink Tree (⇔ Source Tree)
  - Union of all shortest paths between one node and all other nodes
- Shortest-path Routing
  - Distance Vector
    - calculate source tree in a distributed fashion
    - forwarding table
      1. distance to neighbour
      2. distance of neighbour to every destination (their source trees)
    - ⇒ Distributed Bellman-Ford
      1. nodes only know costs to their neighbours, initially
      2. nodes only communicate with their neighbours

      ⇒ node broadcasts to neighbour of its current table till no more update (calculates on the fly)
    - potential issue: count-to-infinity problem (when update under node failure)

      e.g.



: B sees a path to A through C, which will send back B...

$\Rightarrow$ do NOT advertize route back to its source (where you learn)

- Link State
  - calculate source tree & maintain topology
  - fowarding table
    1. network topology
    2. distance to neoghbour & source tree for each neighbour (calculated from topology)
  - $\Rightarrow$ Dijkstra (BFS with first-ordered queue)
    1. complete topology required to be known at each node before the start
    2. for each node: $\mathcal{O}(E + V \log V)$, where E is number of edges, V of vertices (based on heap)
    3. nodes only know costs to their neighbours, initially
    4. nodes only communicate with their neighbours

    $\Rightarrow$ node passes the topology info (till no more update); and then calculate locally

    (though can calculate on fly as well)

    $\Rightarrow$ change in the view: broadcast and re-calculate
  - potential issue: replicated computation (on each node), yet effective
- Comparison

Expectation	Distance Vector	Link State
Correctness	Distributed Bellman-Ford	Replicated Dijkstra
Efficiency	Reasonable – shortest path	Reasonable – shortest path
Fairness	Reasonable – shortest path	Reasonable – shortest path
Convergence	**Slow… many exchanges**	**Fast – flood and compute**
Scalability	**Excellent – storage/compute**	**Ok – storage/compute**

Note: Distance Vector will flood the network at the start as well...

scalability: enterprises level still (not scale regarding global scale)

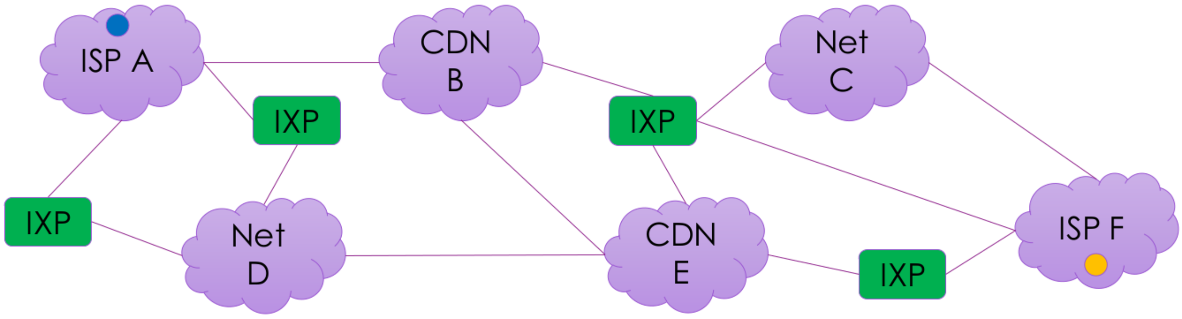
- Equal Cost Multipath routing (ECMP)
  - Overview
    - an extension for flexibility, instead of protocol/algorithm
    - allow for multiple paths between A and B $\Rightarrow$ better redundancy, performance
  - Shortest Route
    - a set of paths, instead of a path
    - sink/source acyclic graph, instead of tree
  - ECMP Forwarding: Choosing Path
    - random $\Rightarrow$ balancing load, afraid of jitter
    - based on relation $\Rightarrow$ rules enabled based on packet info (destination, source, etc.)
    - based on flow $\Rightarrow$ flow identifier in IPv6, less balanced, more predictable (same for 2.)
- Hierarchical Routing

- Goal
    - attempt scaling problem: routing tables, routing computing, fowarding tables, etc.
- Aggregation
    - LAN prefixes aggregated already ⇒ whole LAN as one host (node)
    - aggregate group of subnets into a larger subnet
    - geographical aggregation ⇒ desginated gateways

  ⇒ internal complexity hidden, thus shorter tables

  ⇒ cons: no longer the shortest path

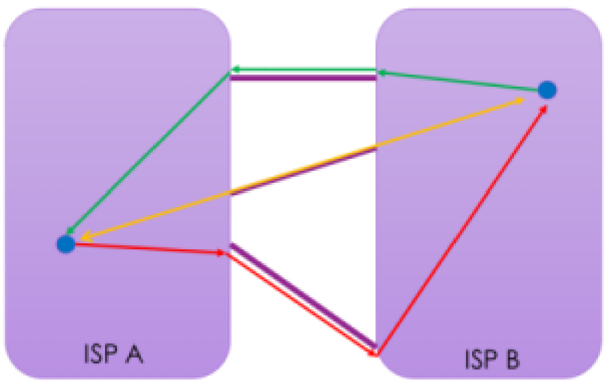
- Policy-Based Routing (Routing Policies)
    - Internet Overview

        - multiple ISPs inter-connected via Internet Exchange Point (IXP)
        - business-oriented
    - Layer 8+
        - money, politics, security, religion, ...

      ⇒ policies upon all layers

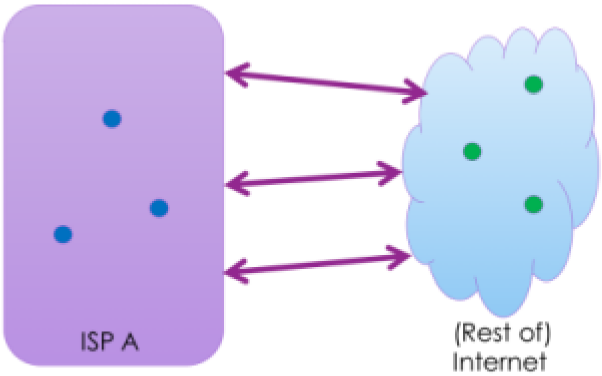
    - Shortest Path in Reality

        - shortest path is only local priority
        - global priority: get out of my business as quickly as possible ⇒ use others bandwidth

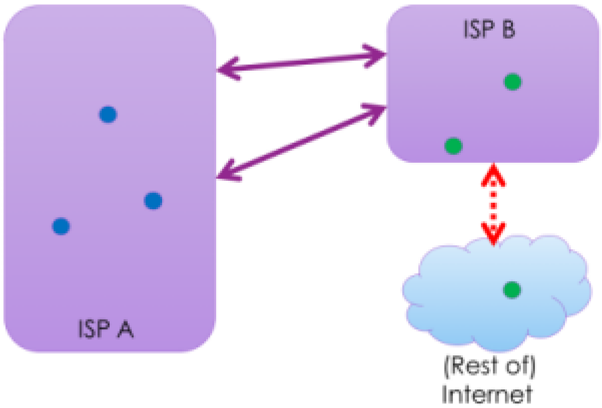
          ⇒ sub-optimal path & asymmetric path

    - Common Policies
        - Transiting



$\Rightarrow$ customer pays for traffic between ISP's network and outer internet

- Peering

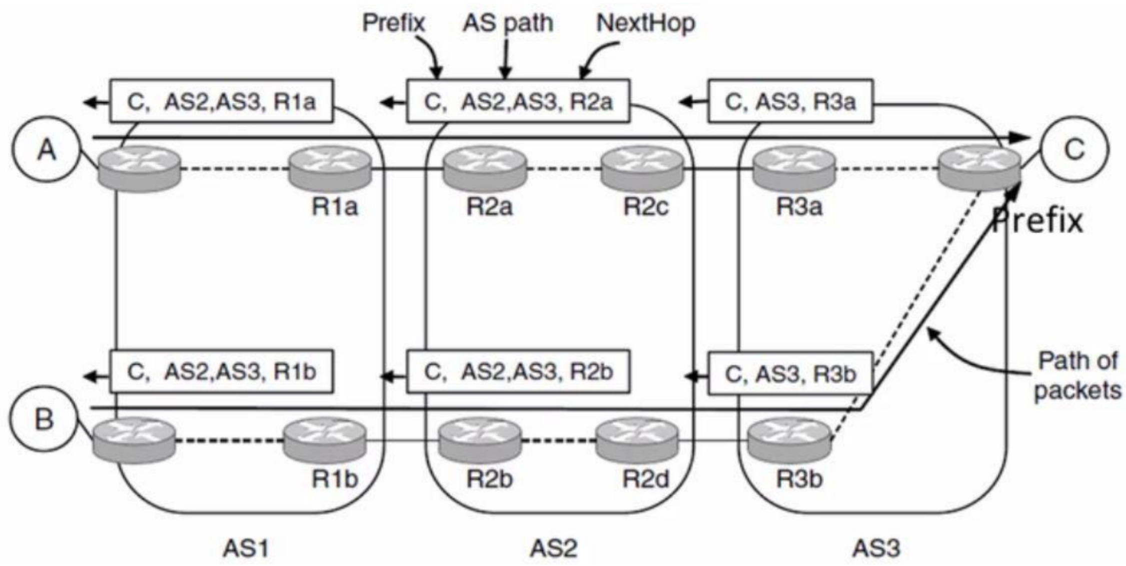


$\Rightarrow$ free traffic between ISP's and its peers' network $\Rightarrow$ mutual benefit

- Border Gateway Protocol (BGP)
  - Goal
    - separation of interior routing protocols and exterior routing protocols

      $\Rightarrow$ intra-domain vs. inter-domain

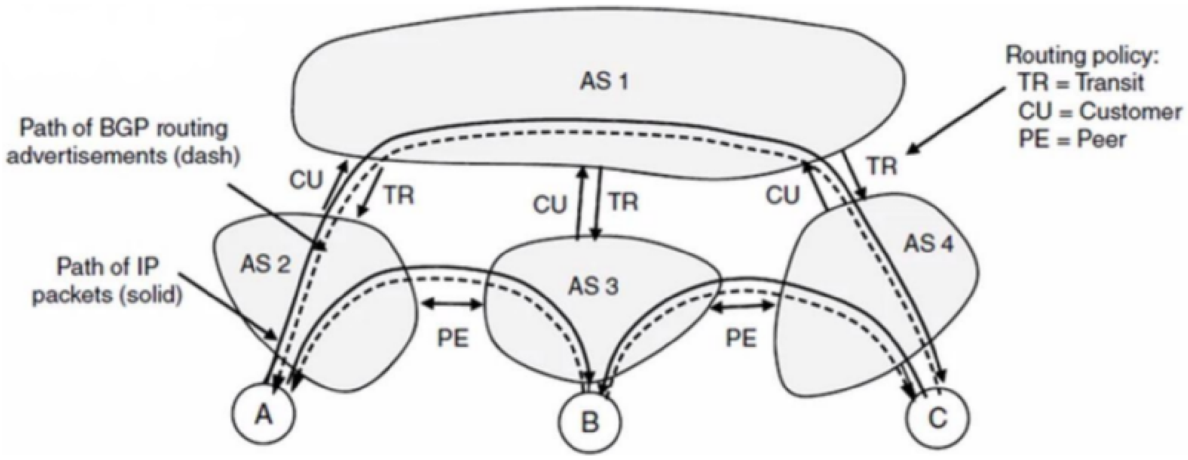
      $\Rightarrow$ maintain a globally distributed system (Internet)
    - destignation of border router (gateway)
    - aggregation all nodes within an 'autonomous system' (AS) - domain/region
  - Advertisement through Path Vector
    - path vector
      1. IP prefix
      2. next hop (highly aggregated node, usually an ISP)
      3. path: list of AS's to transit through (no distance indications)
    - Example: advertising host C



1. extend the path (list of AS's) while advertising, so that each gateway knows
2. BGP only advertises to border gateway (inside path decided locally)
3. BGP advertises to available paths, based on policy

   e.g. offer peering to some, while transiting to others
4. after finished: multiple paths available - can be chosen based on pure human reasons

- Business Example in BGP



  - transiting
    1. AS 1 sells to AS 2,3,4
    2. AS 2 sells to customer A, when it communicating with C
  - peering: AS 2 $\leftrightarrow$ AS 3; AS 3 $\leftrightarrow$ AS 4

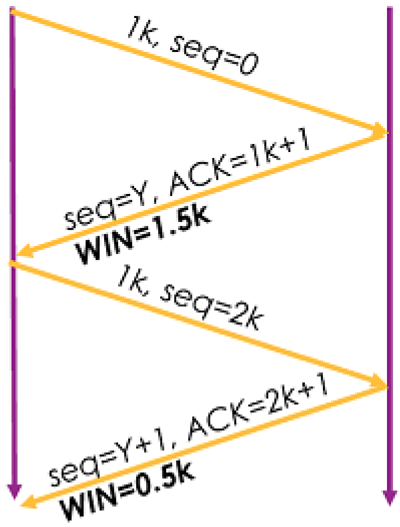
Flow Control and Congestion

- Sliding Window (sender) - Selective Repeat (receiver)
  - Sliding Window (sender)
    - allow upto w outstanding segments not ACKed
    - a timer per unACKed segment $\Rightarrow$ resend separately
  - Selective Repeat (receiver)
    - buffer many segments (store whatever received)

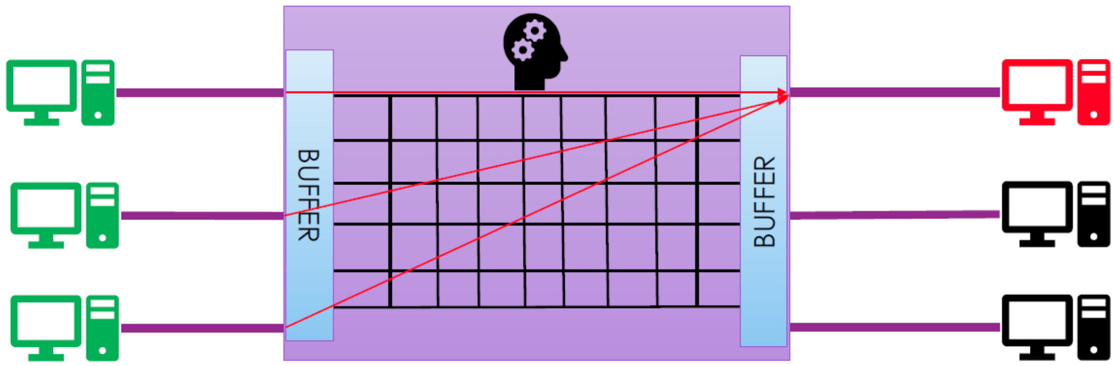
- ACK received segments; require missing segments at the same time

$\Rightarrow$ network oriented: optimized to keep network full

- Receiver Sliding Window ($\Leftrightarrow$ Flow Control Window)
    - Transport Layer
        - receives segments from network
        - appends it to application buffer till its full & wait for application
        - calculate the available slot left in buffer ($\Rightarrow$ windoe size win for flow control window)
        - report its window size win to sender
    - Application Layer
        - call recv() to read from buffer (release space in buffer)
    - Communication with Two Windows



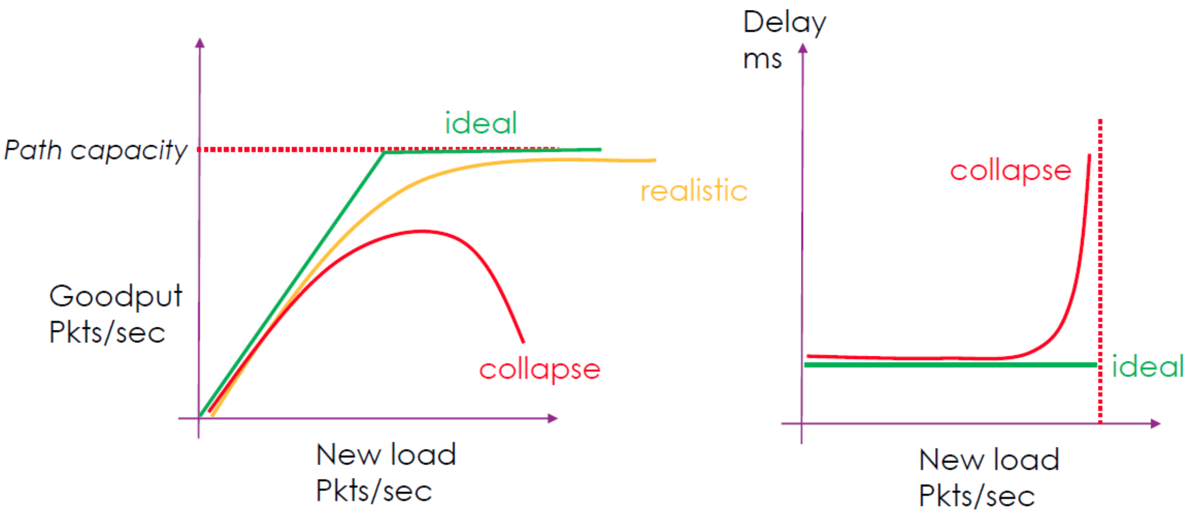
- Congestion
    - Potential Causes
        - too many input to same destination (on routers/switches)



1. outgoing link at a fixed clock rate
2. limited buffer $\Rightarrow$ loss when overflow

    - Goodput
        - throughput at application level

        $\Rightarrow$ number of useful information delivered by network to a destination per unit of time
    - Collapse



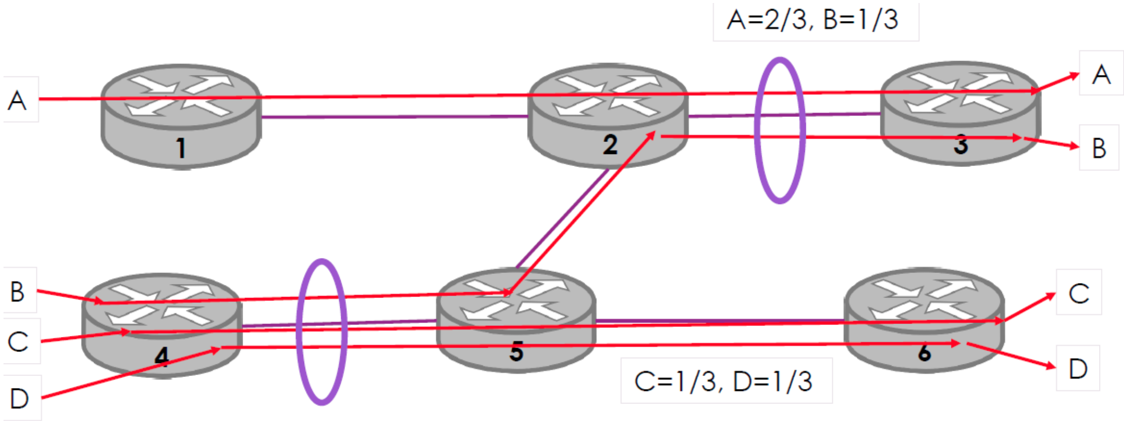
- Causes:
    1. loss appears when approaching the capacity
    2. TCP requests to re-transmit
    3. new messages held back at senders - busy resending old messages & causing more traffic

- Sensing the Congestion

  ⇒ signal from network:

Signal	Pros/Cons?
Packet loss	• Really obvious • Don't detect congestion till it happens
Packet delays	• Detect congestion earlier • Detection is more inferred than actual
Router signal *Explicit Congestion Notification (ECN)*	• Detect congestion earlier • Needs the affected router and hosts to support it

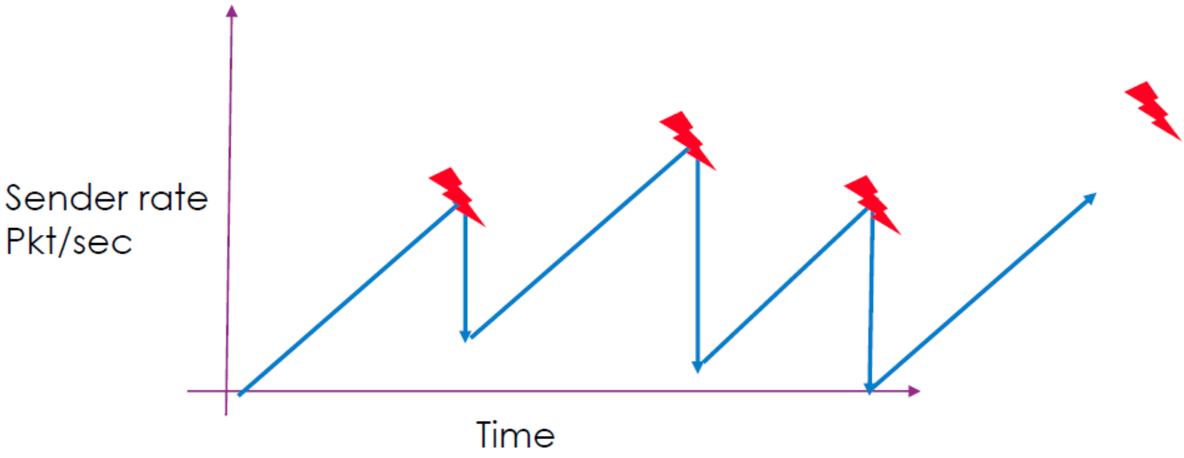
- Manage Capacity
    - Focus
        - efficent use of total capacity
        - fair allocation of total capacity
    - Multiplexing
        - works in Link layer ⇒ due to back-off scheme
        - no high-level control ⇒ sender (application on TCP protocol) just flood the network...
    - Max-Min Fairness
        - goal: maximize the minimum & share the rest ⇒ pareto optimality
        - adapt over time: detect starts of a sender and adapt to the share
        - example:



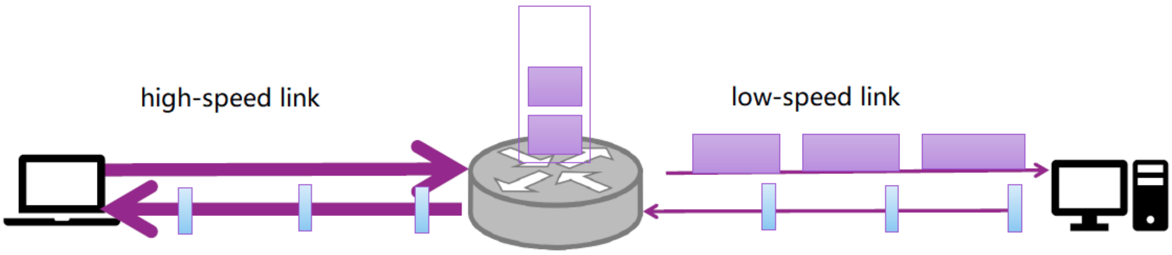
1. bottle-neck: B,C,D using $\frac{1}{3}$ of R4-R5 link
2. $\Rightarrow$ A can have $\frac{2}{3}$ of R2-R3 link

- Adapting Over Time
  - Types
    - open/closed loop
      1. open: reserve circuit in advance
      2. closed: adjust based on feedback (loss, etc.)
    - host/network driven
      1. host: host adapt
      2. network: network policying, strong control yet inflexible
    - rate/window based
      1. rate: set the rate of sending
      2. window: watch for the (flow control window) window size
- Additive Increase Multiple Decrease (AIMD) in TCP
  - Type:
    - closed-loop, host-driven, window-based
  - Collabration across Layers
    - network layer: provide feedback
    - transport layer: adapt the sending behaviour
  - Behaviour over Time - Sawtooth
    - slowly increase (baby step) to probe the network
    - quickly decrease (halves) to avoid congestion (when detecting packet loss)

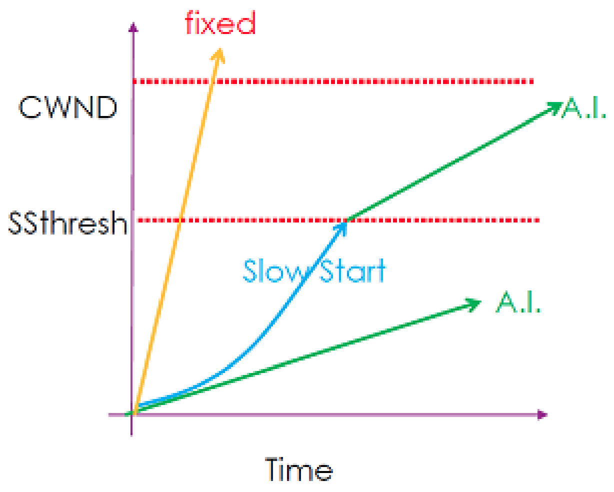
    $\Rightarrow$ essentially mimicing the approaching towards parato optimality



- Features
  - converges to a fair and efficient allocation (when all hosts run it)
  - effective, simple (no burden on network)
- Implementing AIMD
  - ACK clocking process:
    - procedure
      1. burst in the beginning, due to the high speed link $\Rightarrow$ router's buffer copes temporarily
      2. buffer drain after sender matches the ACK rate (slows down)
    - gain
      1. ACK rate reflects the bottle neck in the route $\Rightarrow$ low speed link gets exposed
      2. traffic becomes steady stream



  - Congestion Window (CWND)
    - similar to sender sliding window, denotes the number of outstanding packets (not ACKed)
    - sensitive to congestion $\Rightarrow$ adapt according to congestion
  - Slow Start with Additive Increase
    - procedure
      1. start slow, e.g. CWDN size 1
      2. increasing fast - double the size $\Rightarrow$ exponential increase
      3. additive increase after overshoot $\Rightarrow$ additive increase from $\frac{1}{2}$ capacity
    - gain
      1. probe the capacity faster
         1. stay within the comfort zoon longer (slow-start threshold - maximal capacity of CWND size)



- Fast Retransmit
  - procedure
    1. sender detect (the 3rd) duplicate in its received ACK of a segment

       $\Rightarrow$ something get through (due to CWDN), yet lose the next one
    2. re-send that next segment (assume only 1 segment lost)
  - gain
    1. maintain the ACK clocking

       (if wait for time-out $\Rightarrow$ CWDN may full $\Rightarrow$ no ACK $\Rightarrow$ restart all the way from ACK clocking)
- Fast Recovery
  - procedure
    1. detect congestion (loss), multiple decrease (halves the CWDN size)
    2. assume everything is okay except for that single segment

       (no need for assuming given selective repeat)
    3. continue with the latest sent segment (smooth the window up)
  - gain
    1. recover and continue on with the latest segment directly
    2. avoid re-starting all the way from ACK clocking
- General Implementation
  - ACK clocking
  - slow start and then addictive increase
  - AMID afterwards (with fast retransmit & fast recovery)

- Beyond AMID in TCP
  - Explicit Congestion Notification
    - router notifies receiver its buffer getting full
    - receiver ACK with a flag
    - sender adjust
  - Quality of Service

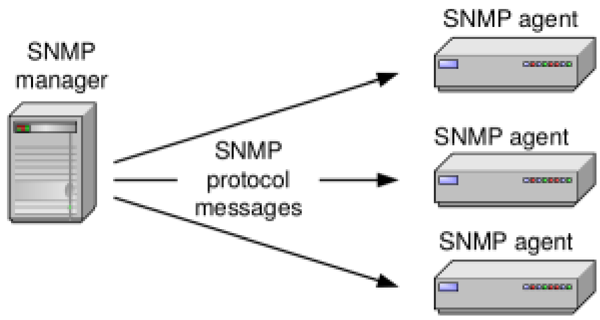
        - tag on packet based on packet type
    - Software Defined Network
        - application require & set up its own network environment
    - etc...

    ⇒ managing packetsrandomly running through a network ⇒ non-trivial

Network Monitoring

- Overview
    - Goal
        - capacity & usage
        - congestion info (places, level, ...)
        - harware / software status
        - changes / update
        - quality of service (application satisfied?)
    - Perspectives
        - within my administrative domain ⇒ monitoring directly
        - Beyond my administrative domain ⇒ info shared by others
    - Feedback from Network
        - Explicit Congestion Notification (ECN)
        - Internet Control Management Protocols (traceroute, ping...)
        - ACK from TCP
        - Application monitoring
- Simple Network Monitoring Protocol (SNMP) - Application Layer
    - Design Principle
        - lightweight ⇒ no extra burden
        - portable ⇒ operate on various platform (sitches, routers, APs, servers, etc.)
        - helpful ⇒ able to fix things
        - scalable ⇒ global monitoring, extensible
        - interactive ⇒ queries/response on demand
    - Components
        - SNMP agents ⇒ software on the target maintaing the status and info

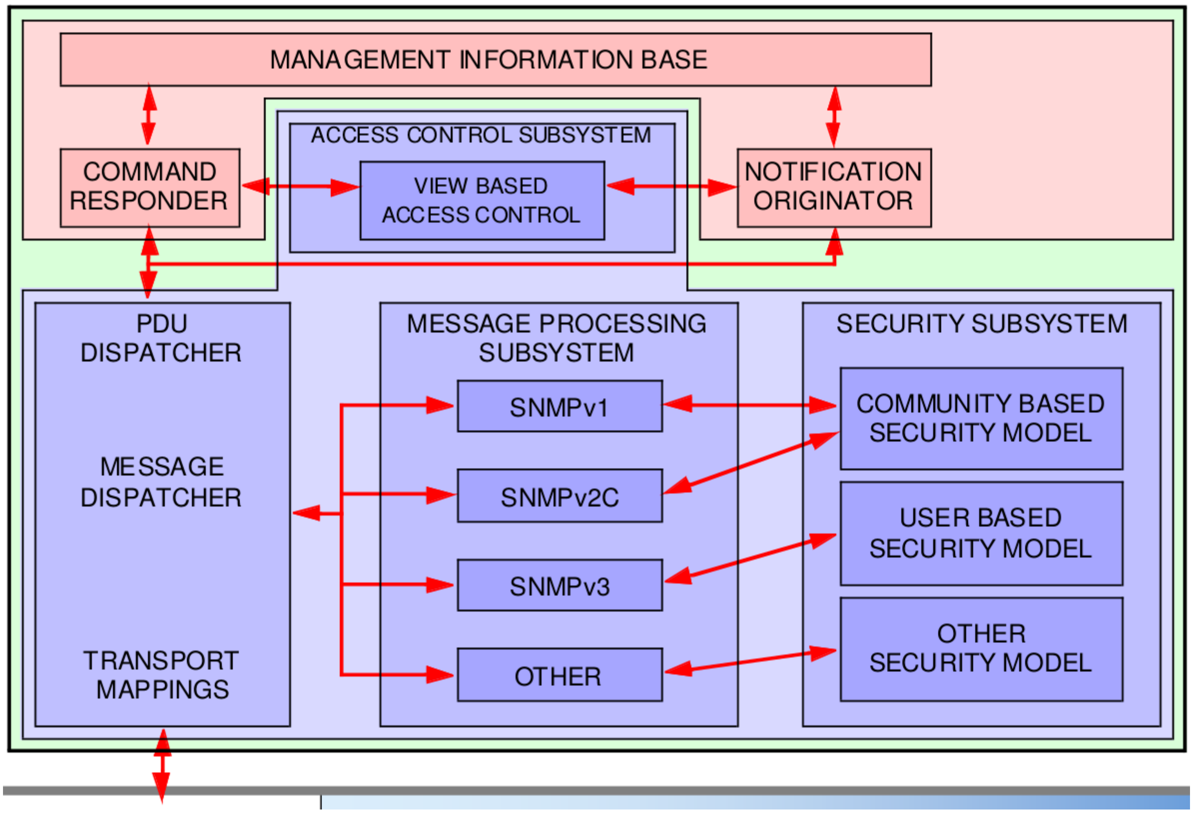
          (use proxies to talk to non-SNMP devices)
        - SNMP managers ⇒ application contacting agents for management / queries
        - Management Information Bases (MIBs) ⇒ describe the database
        - SNMP protocol



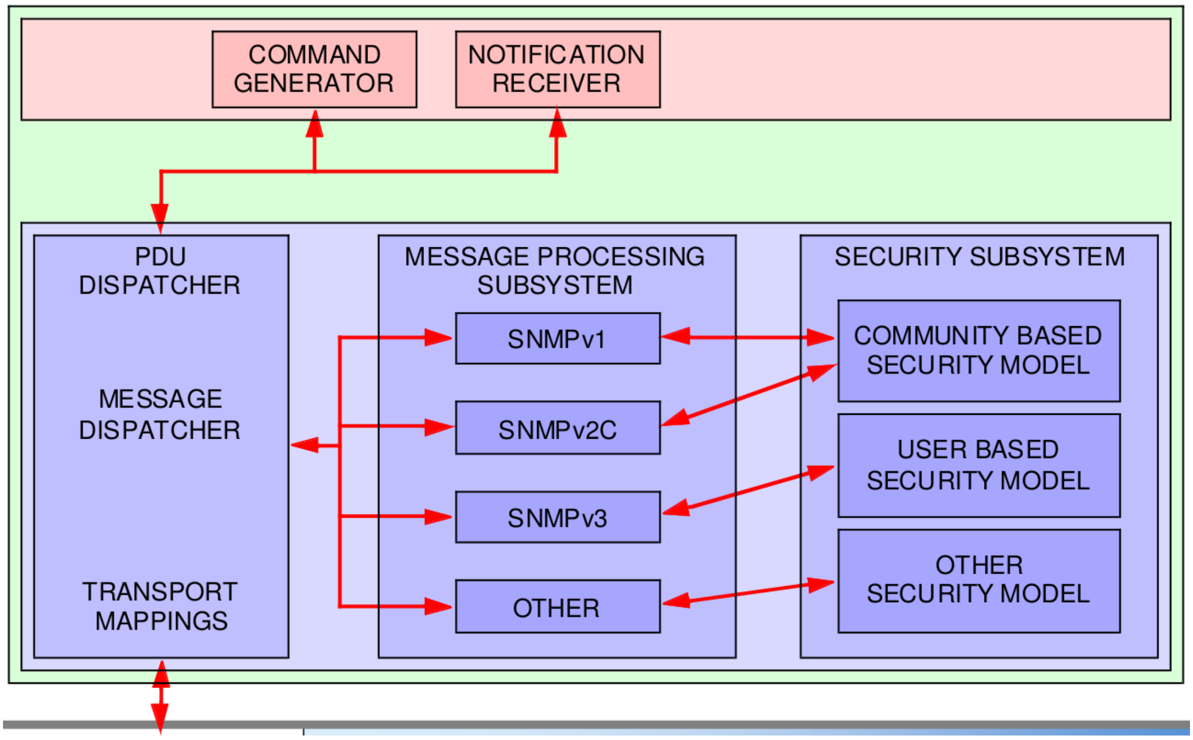
- SNMP Agent
  - counters, gauge, timer since start-up, strings
    1. counter e.g. packets on an interface
    2. gauge e.g. memory/disk space usage
  - light queries / command interface

⇒ no calculation / rate / history on the device (maintained at remote monitor)

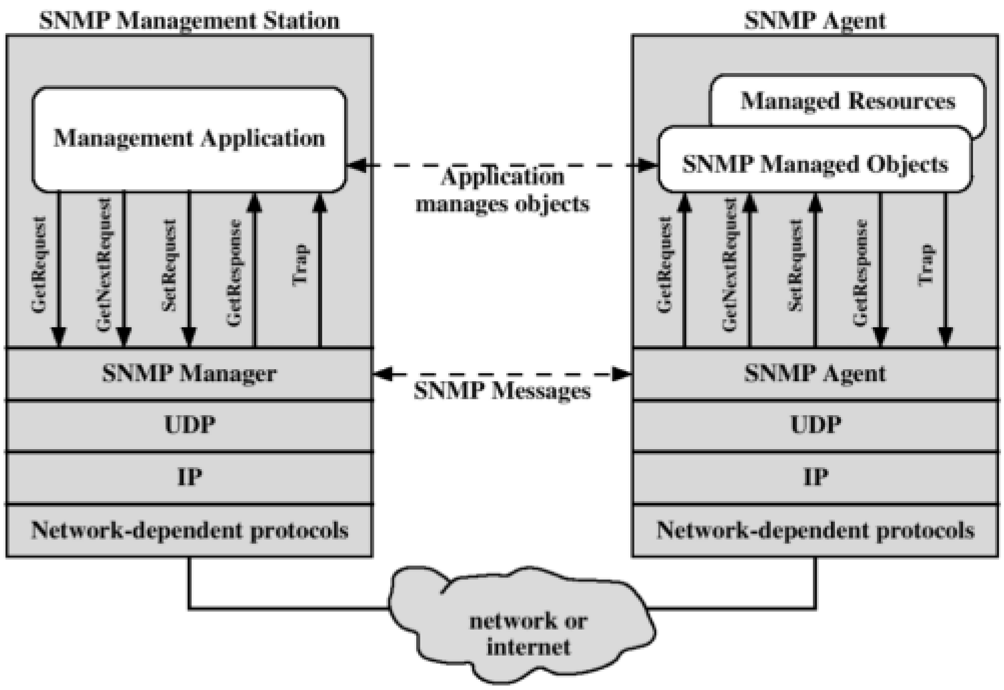
⇒ no pressure, basic information (yet manager may need to ask frequently)



- SNMP Proxy
  - interface for monitoring device without SNMP software (re-present report)
- SNMP Manager



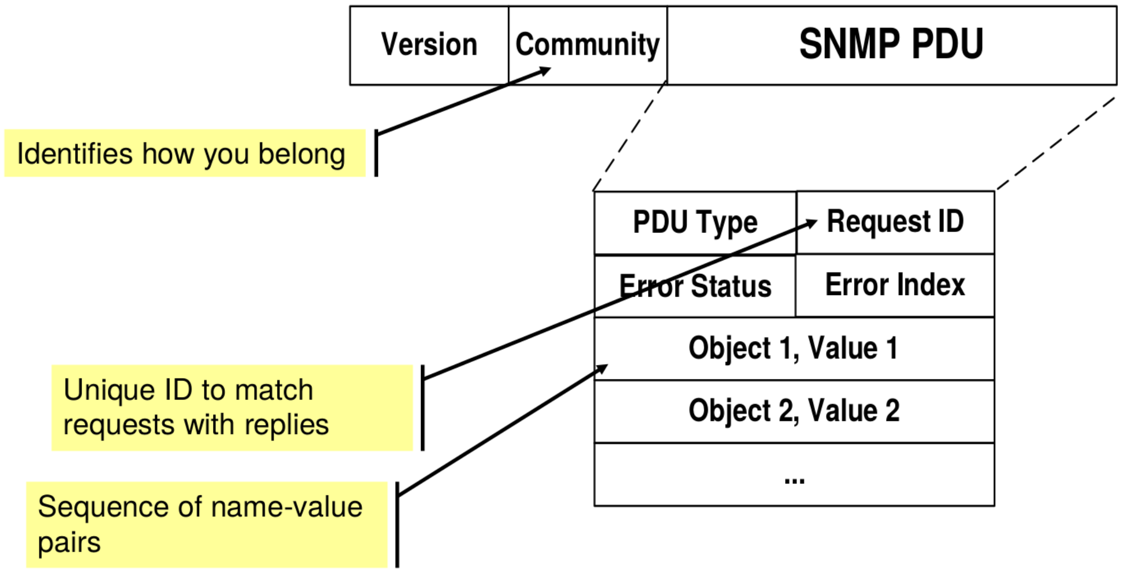
- SNMP Protocol
  - over UDP ⇒ lightweight
  - client-server model (where servers are agents)



- SNMP Messages
  - connectionless ⇒ request ID for session
  - requests
    1. get / set
    2. trap ⇒ a notification from agent to manager, triggered by events at the agent (including connect/drop, expected/unexpected restart, neighbour drop ...)

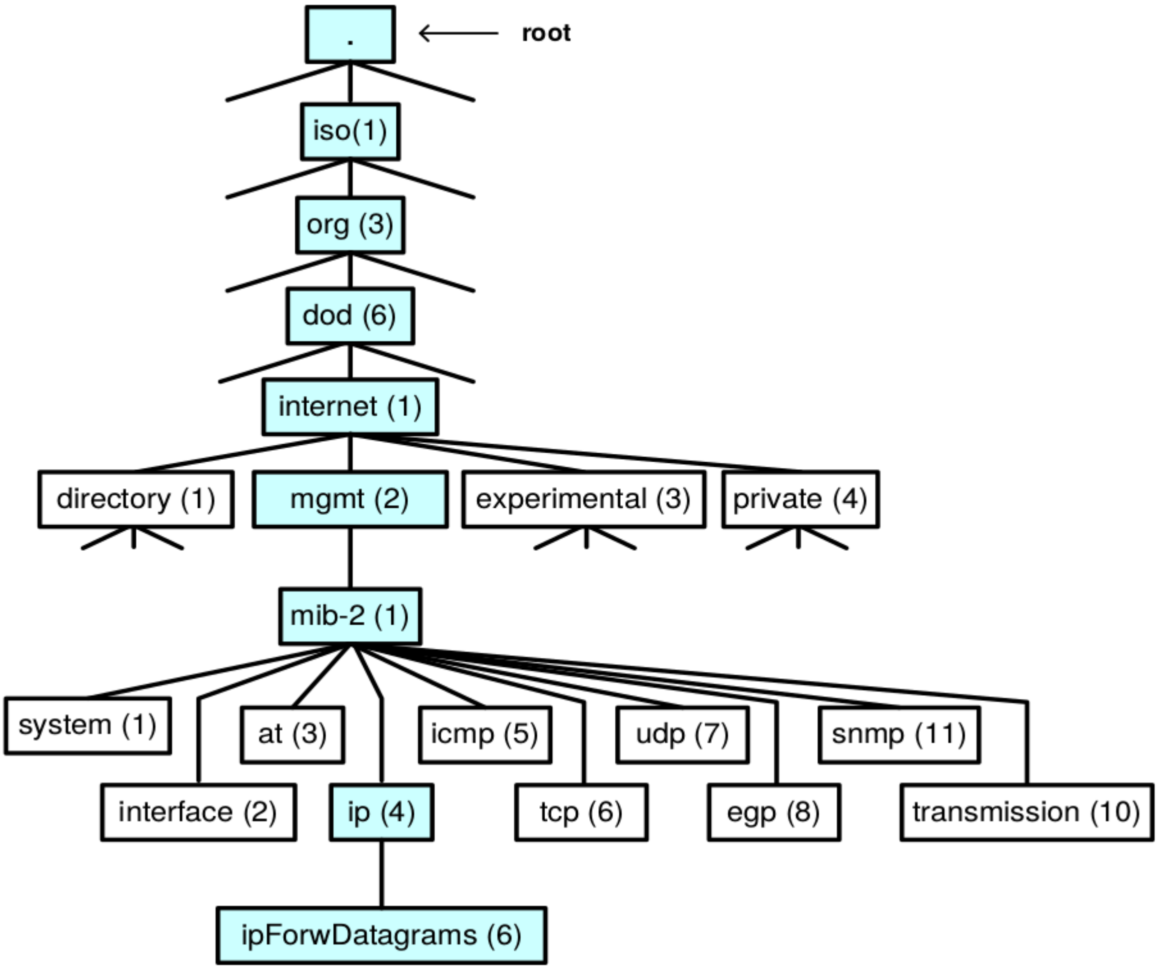
⇒ written in standardised language: ASN.1

- message



- SNMP Management Information Bases (MIB)
  - design structure - ASN.1(Abstract Syntax Notation One)
    1. DNS structure - tree hierarchy for levels of object identifier (OID)

       ⇒ pros: aggregation benefit & scalability
    2. OIDs for global uniqueness (a node in tree), associated with human-readable names
    3. MIB objects: identified by OID, essentially a variable (with its meta data)



- functino of MIB:
  1. map index of OID (leaves in the tree) $\rightarrow$ a variable with its meta data
  2. associate leaves with their description, access permission, type, etc.
  3. offer traversing function to offer a table

  example: MIB object (with an OID of 1.3.6.1.2.1.4.6)

```
ipForwDatagrams OBJECT-TYPE
    SYNTAX  Counter
    ACCESS  read-only
    STATUS  current
    DESCRIPTION
            "The number of input datagrams for which this
            entity was not their final IP destination, as a
            result of which an attempt was made to find a
            route to forward them to that final destination.
            In entities which do not act as IP Gateways, this
            counter will include only those packets which were
            Source-Routed via this entity, and the Source-
            Route option processing was successful."
    ::= { ip 6 }
```

- Security
  - message level
    1. offers: message integrity, authentication & privacy (encryption)

    2. provide different level of security (3 levels)
  - proxies level
    1. a limited view & queries - limited info
    2. beacons: multicast to get information from a set of devices on network (the information are shared - public info)

Network Security

- Overview
  - Layers
    - across all layers
      1. physical link: directly intrefered by cutting, tapping or snooping
      2. network / transmission: packets/segments are not trustable
      3. application: flaw in design/code etc.
  - Risk Management
    - risk & threat
      1. design & code flaws
      2. human flaws
    - consequence of risk vs. effort to remove risk
    - encryption - hide the content
    - integrity - confirm message not changed on its way
    - authautication - confirm message from expected source
- Encription
  - Symmetric - shared key
    - both ends knows the key & same algorithm for both ends
    - pros:
      1. fast & efficient
    - cons:
      1. key sharing / distribution is a weakness
      2. same key used too heavily
      3. attacker usually knows the algorithm
  - Asymmetric - public/private key
    - key pair with public key for encrypting, private key decrypting
    - pros:
      1. no explicit key exchange $\Rightarrow$ easier key sharing
    - cons:
      1. heavy & slow
      2. need to trust the public key - identification
  - General Approach

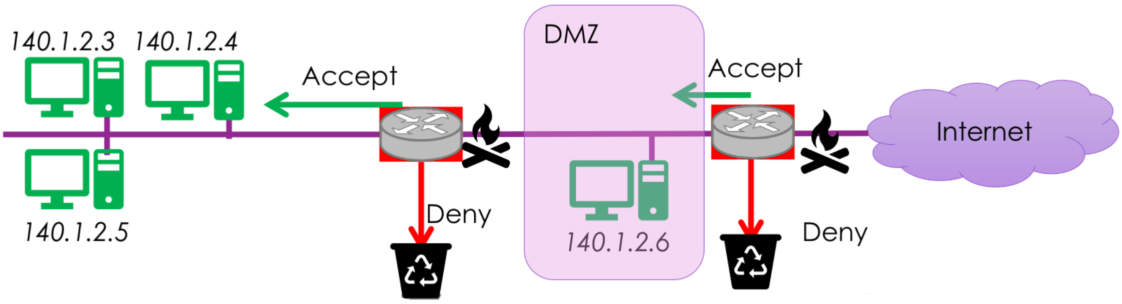
        - share / distribute the symmetric key with
            1. asymmetric encryption
            2. other pathway (phone message, email, etc.)
        - switch symmetric key regularly ⇒ session-key
- Integrity
    - Signature
        - a summary of message, using hash, message digit
        - encripted by session key (after calculation)

    ⇒ ensure (encrypted) message is not changed along the way
- Authentication
    - Freshness
        - time-stamp (time-related stamp) within the signature

          ⇒ avoid the application not accounting time (avoid replay attck e.g. keep transfering to other)
    - Certification / Validation
        - check the offered public key against the one registered on CA (Certificate Authorities)
- Security in Layers
    - Secure Socket Layer - between Application and Transport
        - gain
            1. verification of server
            2. message exchange, with confidentiality, integrity, authentication and freshness
        - procedure
            1. authentication - validate related servers ⇒ through the CA hierarchy (trust chain)
            2. session-key sharing / distribution, using public key
            3. start the encrypted communication
        - cons
            1. another layer ⇒ flaws in code
            2. not used by all application
    - Firewalls - across Layers
        - gain
            1. over the network layer ⇒ explicitly block messages
            2. operate at multiple places ⇒ local host, router, gateway (NAT), modem, access point, etc.
            3. operate at multiple layers ⇒ network, transport, application
        - types
            1. stateless (basic) ⇒ based on IP addr, port, protocol type and other policies
            2. stateful ⇒ tracking communication, rules chaning by events

               e.g. allow any connection initialisation, yet timeout after idle
            3. application firewall ⇒ deep packet inspection, understand context

- two stage firewall
    1. protect some more than others ⇒ demilitarised zone
    2. outside: light firewall (for effective access)
    3. inside: application firewall for heavy protection

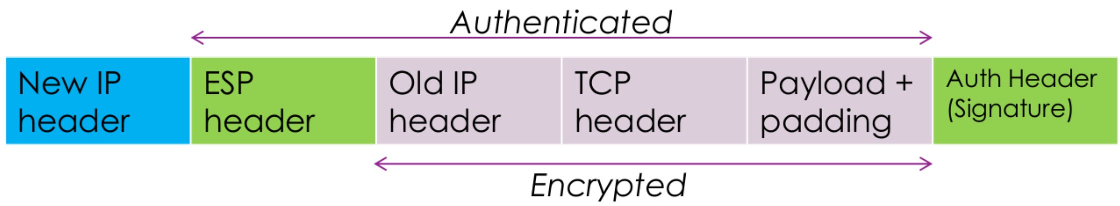


- End-to-End confidentiality - Network Layer
    - gain
        1. islands of trust ⇒ end-to-end / host-to-host / subnet-to-subnet confidentiality
        2. extra protection from intermediate leak of information (can be inferred)
    - deployment
        1. leased line ⇒ private path vs. expensive, hard to manage routing
        2. over public internet ⇒ virtual leased line ⇒ Virtual Private Network (VPN)
    - Virtual Private Network (VPN)
        1. tunnel (encapsulate) IP packets across the Internet ⇒ IP in IP

           (has actually no protection)

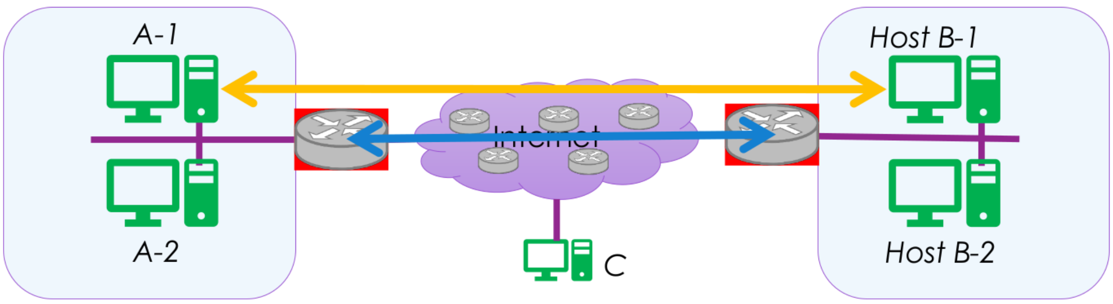
        2. fix three points (entry,), with internet routing in between (vs. circuit)

        3. use IP security over IP ⇒ secure connections between end points (router/host)



(ESP: encapsulating security payload)

- VPN type
    1. router-to-router (tunnel mode) ⇒ transparent, NAT-friendly, with forwarding
    2. host-to-host (transport mode) ⇒ NAT-challenged, no forwarding (with no inner IP header)



- address opaqueness (hidden)

1. the destination of outer IP: where the initial IP packet appears in the network
2. the original address is hidden

- discovering VPN

1. unusual access pattern
2. well-know VPN companies list
3. no other means...

- Hardware Security
  - pyhsical access to copper, fiber, wireless APs ...

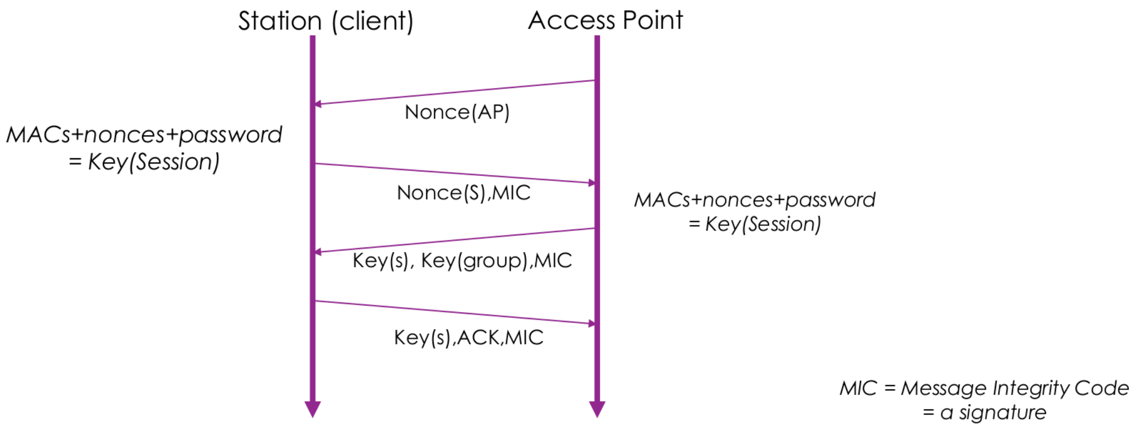
    1. tap the copper
    2. detect energy loss in fiber
    3. just monitoring around APs

    e.g. host 1 $\rightarrow$ frames in secure packets $\rightarrow$ AP $\rightarrow$ frames in raw packet $\rightarrow$ host 2
  - hardware/software interface provided at network devices

    1. reflect the traffic to another port, ...
    2. SNMP agents, operating systems...
- Security in Environments
  - Home Wireless Network
    - typical un-open network
    - client-to-AP $\Rightarrow$ preshared secret key (PSK)

      1. client authenticates to AP (that it know the secret), using password
      2. derives a session key from the password
    - AP-to-client $\Rightarrow$ group temporal key (GTK)

      1. key-on-key, keys over key-on-key $\Rightarrow$ involve multiple keys
      2. includes group temporal encryption key, (AP) Tx key, (AP) Rx key, etc...



    Note: nonce = random number
  - Enterprise Network
    - pass-through authentication

      1. each client has its own credential $\Rightarrow$ sending to remote server for authentication
      2. using both username and password

e.g. ANU wireless

- Denial of Service - beyond Encryption
  - Goal
    - prevent server from the target clients
  - Basic Means
    - resource starvation
  - Attacks
    - ping of death
      1. an IP packet with inappropriate fragmentation tag, size
      2. buffer overflow in system ⇒ crash the host

      fix: better OS protection
    - SYN flood
      1. flooding SYN to establish connection
      2. taking up resource on server

      fix: SYN cookies ⇒ no setup until cooresponding ACK (similar to mem allocate on linux)
    - attacking application-level flaws
    - spoofing
      1. draft queries for someone else

      fix: ingress filtering ⇒ router check
    - multiplers
      1. network with hacked devices ⇒ flooding
      2. big responses for small request, often with spoofing
  - Main Solution
    - content distributed network (CDN)
    - edge routers / attack detection
    - upstream provider support ⇒ filtering traffic at the source, provided by ISP
    - ingress filtering everywhere!

Real-World Network Example

- ArrNet
  - Overview
    - education use
    - high speed within cooperators' own their own fibers (part of the fiber)
    - rollout its own network & fault tolerance
- Infini Band for HPC
  - Overview
    - cost: heat of switch & cables
    - topologies: fat tree (predicted latency), 3-D tours, hypercube
    - RDMA: application starts connection itself (no kernel & TCPIP stack)

        ⇒ specialized hardware needed (DMA across network)
        - link layer handles subnet distribution
        - acceleration of data flow
        - centralized ⇒ need subnet manager & agent (in device to respond to manager)
        - element: channel adapter, switch, router, subnet manager
- TransACT
    - Overview
        - Open & full-service network, service providers (video, data, voice etc.) are not threatened

          ⇒ network focus only on access, providing users with more options
        - scattered users ⇒ last mile problem

          (fiber carries – like highway, but not distribute to end-user – like small roads)
        - HFC problem – coax as huge antenna (noise picked uip at joint point)
        - FTTN + xDSL solution, with constraint:
            1. XDSL performance vs. Distance curve
            2. bundle size allowed
            3. self-catered node (deal with its heat...)