From Packet Switching -> Delay/Congestion
Packets queue in router buffers (when pkt arrival rate to link (temporarily) > output link capacity.

1. Processing Delay (duration < ms, bounded by packet size & router's HW capabilities)
Needs time to examine packet header for: 1) check for bit errors (checksum); 2) determine output link by dest IP addr (bounded) because packet size is fixed. (smaller the packet, better it is because if 1 bit dropped, delete only a smol pkt)

2. Queuing Delay (duration dep on traffic, unbounded)
Packet needs to wait to get to the front of the queue to reach output link due to congestion (input rate>output rate)
- Traffic Intensity = La/R (L=pkt length(bits), a= avg pkt arr rate, R=link bandwidth (bits/s))
Avg Link Utilisation - [0,1], ~0: delay smol, -> 1: delay large, >1: delay infinite

3. Transmission Delay (dep on link bandwidth, bounded)
- delay = L/R
- need time to push whole packet bits from router end to the link (cable)

4. Propagation Delay (dep on link length, bounded)
- delay = d/S (d=link length (m), S=propagation speed of bits on wire (app 2x10^8m/s))
- Need time for bits to propagate from the output end of Router 1 to input end of Router 2

Total Delay (nodal, router to router): Add up all 4 delay timings

Packet Loss:
- Transport layer protocol ensures delivery between end to end host system. Hence, a packet loss may be retransmitted by the sender, depending on the protocol agreed.

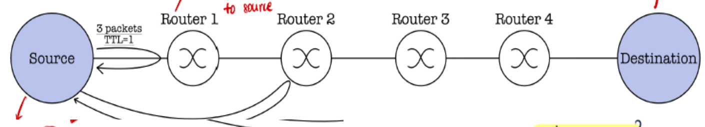
Computing Throughput:
- Packets are transferred between hosts through a series of router(s).
- Throughput: rate (bits/time unit) at which bits can be transferred between sender/receiver
- Effective throughput between 2 endpoints is limited by the **SLOWEST** link
- Instantaneous: rate at given point in time | average: rate over longer period of time

ICMP: Internet control message Protocol
to send error messages and operational information (standard messages)

TRACEROUTE

Finding internet packet delay and packet routes in real life

TTL decides #hops
EG. TTL=3, RTT signifies 6hops



Each router reduces TTL by 1. if TTL =0 but router ≠ destination, then it sends back ICMP TTL exceeded to source

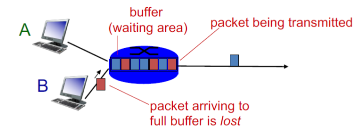
destination host finds out that the packet is intended for a port that is unusable, sends ICMP: port unreachable to source

(1) sends 3 packets w increasing TTL each time it receives ICMP TTL exceeded, and record the RTT (round trip time)
to a likely unusable port in the destination
(2) stops sending once it receives ICMP: port unreachable
(or when too many *** timeout has occurred)

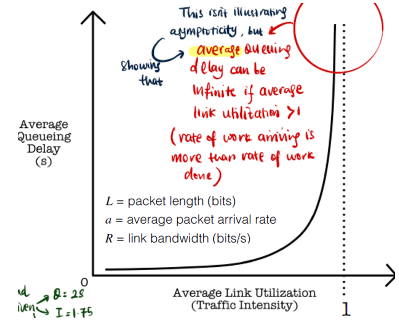
What is port #?
16 bits logical representation of a service or app in the host system.
upon receiving a packet, the OS will try sending it to the port # identified in the packet or send ICMP port unreachable message if no app has that port #

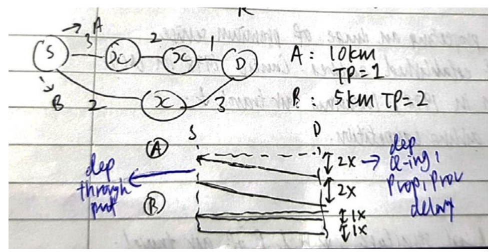
Packet loss

- queue (aka buffer) preceding link in buffer has finite capacity
- packet arriving to full queue dropped (aka lost)
- lost packet may be retransmitted by previous node, by source end system, or not at all



- Loss is *random* – each link has average *loss rate/probability*
- How do per-link loss rates compose into end-to-end loss rate?
- Impact of loss on applications: Email? Banking? Images? Video?





EG. given that $\bar{Q} = 0.01s$ when $I = 0.7$
$P_{active} = 0.3$
20 users, each sends 10pkt/s
1kb/pkt when active

(1) what is R to keep $\bar{Q} = 0.01s$ on avg?
avg #users

$I = \frac{La}{R}$

$0.7 = \frac{1000(10) \times 20 \times 0.3}{R}$ (n, P_a)

solve for R.

(2) what is R to keep Q under 0.01s at ALL times? (ie max $\bar{Q} = 0.01s$)

$0.7 = \frac{1000(10) \times 20}{R}$ ← worst case: all active

(3) And given $Q = 2s$, $I = 1.75$, $R = 512b/s$ what is the probability that the above happens?

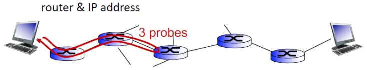
$1.75 = \frac{1000(10)n}{512}$ → find n.

$P_{a=n\,users} = {}^{20}C_n \times 0.3^n \, 0.7^{20-n}$

Traceroute and ICMP

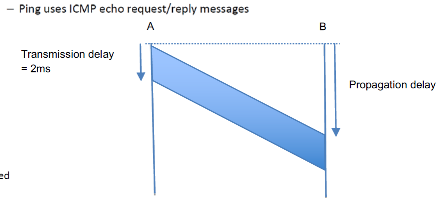
- source sends series of UDP segments to destination
  - first set (of 3 packets) has TTL =1
  - second set has TTL=2, etc.
  - unlikely port number
- when *n*th set of packets arrives to *n*th router:
  - router discards datagrams
  - and sends source ICMP messages (type 11, code 0)
  - ICMP messages includes name of router & IP address
- when ICMP messages arrives, source records RTTs

stopping criteria:
- UDP segment eventually arrives at destination host
- destination returns ICMP "port unreachable" message (type 3, code 3)
- source stops



NB: TTL is *time-to-live* field in IP packet header; it defines maximum number of times the packet can be forwarded before the packet is dropped (considered not deliverable)
Each router decrements TTL – if decremented TTL still +ve, packet is forwarded; otherwise packet is dropped

- Runs at both end hosts and routers
  - Traceroute uses (i) ICMP TTL Exceeded, (ii) ICMP port unreachable messages [port unreachable means no app is interested in the (destination port of the) packet]
  - Ping uses ICMP echo request/reply messages



1. In RSA, suppose we have a public key (n=55, e=3). Find the private key (n,d) corresponding to the public key such that d is the smallest possible.
(n=55, d=27) ***Note: p=5; q=11; z=40. Need 3d mod 40 to give a remainder of 1; smallest 3d is 81, so d is 27.***
Non-repudiation is the assurance that someone cannot deny the validity of something.
3. Assume that Alice and Bob had previously established a secret symmetric key S for encryption of communication between them, and Alice obtained a message M from Bob encrypted by S. For each of the following statements, say if it is true or false and explain why.
a) Alice knows that the message must indeed have come from Bob. **True. Only Alice and Bob are able to generate the message, and Alice knows that she didn't do it.** ***NB: The intended meaning of "come from Bob" is that Bob authored the message. If student answers false and explains it by saying that someone else may have sent Bob's***
message to Alice, count the answer as correct.
b) Alice can take the encrypted message to court with non-repudiation that the message indeed came from Bob. **False. Since besides Bob, Alice could also have generated the message, she can't prove it to the court that the message came from Bob and not her.**

4. 1000 bytes of data is sent as quickly as possible from A to B over a link of propagation delay 6ms and bandwidth 4 Mbits/s. Ignore nodal processing and queueing delays. (a) Draw a spacetime diagram of the whole transfer, from when the first bit is sent by A to when the last bit is received by B. (b) Calculate the transmission delay and indicate it in the diagram. (c) Indicate the propagation delay in the diagram.

Important Security Properties:

1. Confidentiality: only sender & intended receiver should 'understand' the messages content (sender encrypts msg, receiver decrypts msg)
2. Authentication: sender & receiver should confirm identify of each other
3. Message Integrity: sender & receiver wants to ensure message not altered (in transit, or afterwards) without detection
4. Access: communication link b/w sender & receiver should be accessible
5. Availability: comm link b/w S&R should be available

Possible Security Attack:

1. Eavesdrop: Intercept messages
2. Alter Messages: Actively insert messages into connection
3. Impersonate: Fake (spoof) source address in pkt or any part of pkt
4. Denial of Service: Prevent service from being used by others
5. Hijack: Take over ongoing connection by removing sender/receiver unknowingly to either party

Cryptography

- Protects confidentiality, hence preventing attkers from eavesdropping
- Naive: subsitution cipher (mapping from a set of 26 letters to another set of 26 letters) Con: frequencies of letters not masked at all
- Two good methods to encrypt & decrypt plaintext:

1. Symmetric Key (faster than A key)

- encryption & decryp use same key (hard to agree)
- both below are fixed length inputs (if less than that, pad with 0)

Data Encryption Standard (DES):

> 56-bit symmetric key, 64-bit plain text input

> brute force in less than a day

Advanced Encryption Standard (AES): (slower but more secure)

> 128, 192 or 256-bit sym key, 128bit plain text inp

> brute force in 1 bil bil year (if 128bit key)

2. Asymmetric Key

Example: Public Key Crypto

- Sender & receiver do not share secret key
- Public encryption key known to all
- Private decrypt key known only to receiver
- Input/output size: depends on key size
- 1 round of encryption

- 2 important requirements:

1) Given public key, impossible to compute private key
2) Need a public-private key pair st public(m) = encrypted msg (encryp); private(encrypted msg) = m (decryp)

Data Encryption Standard (DES):

- An algorithm to perform symmetric key cryptography
- 56-bit symmetric key
- 64-bit plaintext input (padding may be needed)
- 16 rounds of encryption to produce 64-bit encrypted output
- It was US encryption standard in 1993
- Can be decrypted using brute force in less than a day
- 3-DES to make it *more secure*

Advanced Encryption Standard (AES):

- A better algorithm to perform symmetric key cryptography
- 128, 192, or 256-bit symmetric key
- 128-bit plaintext input (padding may be needed)
- 10,12,14 rounds of encryption to produce 128-bit encrypted output
- Replaced DES in 2001,
- If 128-bit key is used, brute force decryption will take 1 billion *billion* years (yes, double billions right there), with a supercomputer

Rivest-Shamir-Adleman (RSA) algorithm (1977):

- Sender, receiver, do **not** share secret key
- **Public** encryption key **known to all**
- **Private** decryption key known only to receiver
- 1024 - 4096-bit asymmetric (public-private) key pair
- 1 round of encryption
- Input/output size: depends on key size

Two **important** requirements:

- Given **public** key, it is **impossible** to compute the private key
- Need a public-private key pair such that,
  - public(m) = encrypted message ------ **encryption**
  - private(encrypted message) = m ------ **decryption**

$$[(a \mod n) \pm (b \mod n)] \mod n = (a \pm b) \mod n$$
$$[(a \mod n) * (b \mod n)] \mod n = (a * b) \mod n$$
$$[(a \mod n)^d] \mod n = a^d \mod n$$

1. Choose 2 large **prime** number : p & q (1024 bits each)
2. Compute:

$$n = pq,\ z = (p-1)(q-1)$$

relatively prime: dont share common divisor except 1
eg 9 and 16
but 11 and 22 are not

3. Choose e, where e<n, and e,z are relatively prime to one another
4. Choose d, where (ed-1) mod z = 0
5. **Public key, is (n,e):**

$$c = m^e \mod n \longrightarrow$$

m = message (unencrypted), where **m must be < n**
c = encrypted message

6. **Private key, is (n,d):**

$$m = c^d \mod n$$

7. An important property:

$$\text{if } k = m^d \mod n,$$
$$k^e \mod n = m = c^d \mod n$$
$$x^y \mod n = x^{(y \mod z)} \mod n \text{ for any } x, y$$

note this is an example, so we choose small numbers for ease of calculation.

1. Choose ~~2 large~~ **prime** number : p=5 & q=7
2. Compute:

$$n = 5*7 = 35,\ z = (5-1)(7-1) = 24$$

3. Choose e = 5, where e<n, and e,z are relatively prime to one another
4. Choose d = 29, where (ed-1) mod z = 0
5. **Public key, is (n,e):**

$$c = 12^5 \mod 35 = 17$$

m = 00001100 (this is 12)
c = 00010001 (this is 17)

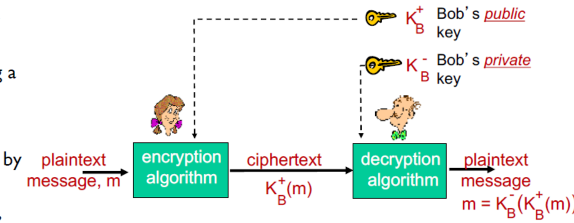
6. **Private key, is (n,d):**

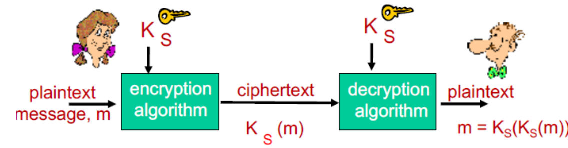
$$m = 17^{29} \mod 35 = 12$$

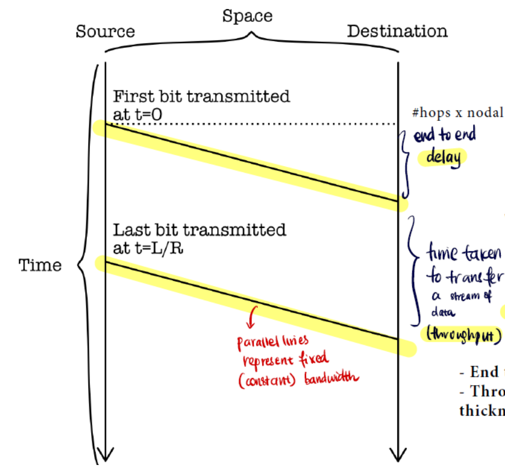
- message: just a bit pattern
- bit pattern can be uniquely represented by an integer number
- thus, encrypting a message is equivalent to encrypting a number.

example:

- m= 10010001 . This message is uniquely represented by the decimal number 145.
- to encrypt m, we encrypt the corresponding number, which gives a new number (the ciphertext).







NETWORK PERFORMANCE VISUALIZATION

The Space-Time diagram

Network performance is both **delay** and **throughput** limited

- End to end delay: how flat
- Throughput/bandwidth speed: thickness

requirements:

1. need $K^+_B(\cdot)$ and $K^-_B(\cdot)$ such that
   $K^-_B(K^+_B(m)) = m$
2. given public key K^+_B, it should be impossible to compute private key K^-_B

The following property will be *very* useful later:

$$K^-_B(K^+_B(m)) = m = K^+_B(K^-_B(m))$$

use public key first, followed by private key

use private key first, followed by public key

result is the same!

follows directly from modular arithmetic:

$(m^e \bmod n)^d \bmod n = m^{ed} \bmod n$
$= m^{de} \bmod n$
$= (m^d \bmod n)^e \bmod n$