Computer networks I- Data Communications

D N Ranasinghe

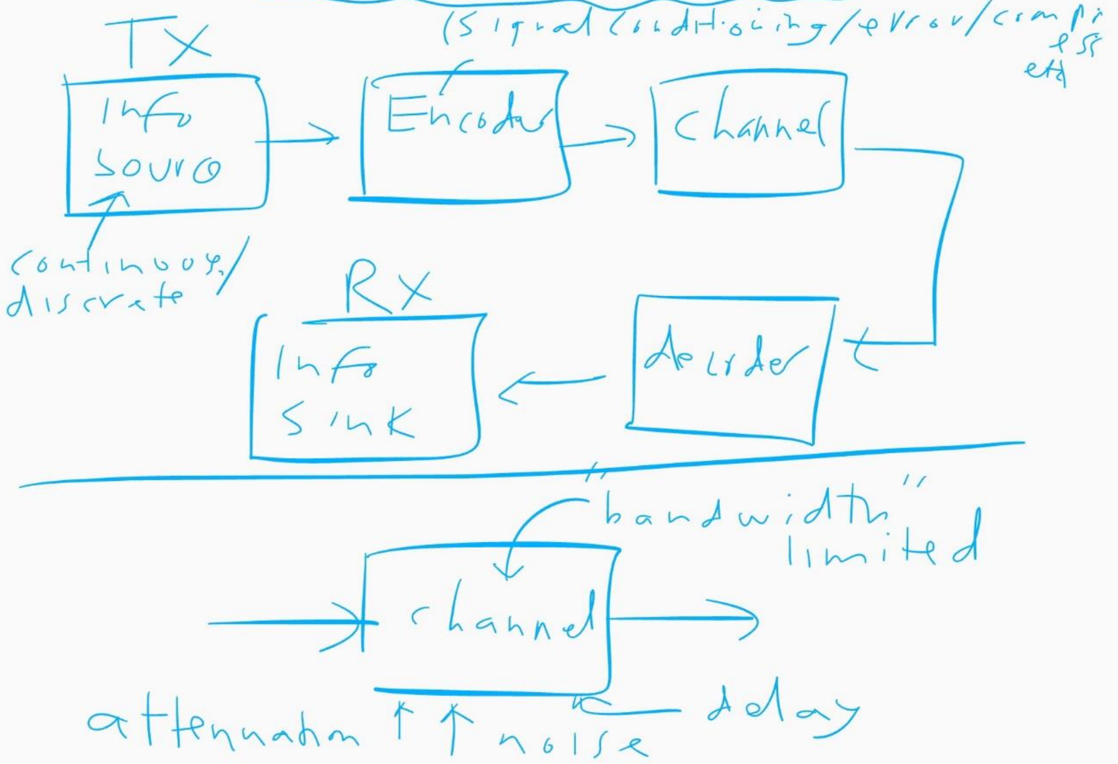
preliminaries

- Text books and assignments – see LMS for details
- Course will be augmented with material from CISCO
- Why study computer networks?
  - ICT – ‘communication’
  - Working/studying/business from home these days depends on communication
  - Cloud services provision depends on it
  - New developments – software defined networks
  - Etc., etc.,

fundamentals

- Infrastructure and protocols for Information sharing
  - infrastructure - network channels (wired, wireless, fixed, mobile, point to point and broadcast etc,)
  - channel properties (bandwidth, noise, delay distortion, attenuation)
  - protocols - bipartisan agreement for information exchange (start, exchange while correcting errors, terminate)
- types of information
  - man made, machine made, continuous, discrete, 1-D, 2-D etc.
- what is information: quantified by Claude Shannon; e.g., a single tone does not carry any ‘useful’ information

Data comm channel model



- physics of communication
    - time domain and frequency domain (spectrum) representations
    - Fourier series representation of any time domain periodic signal as the sum of a dc component and a mix of sine/cosine harmonics
    - e.g., spectrum of a pulse train as a Sinc function with infinite number of frequency components
    - the need to limit the spread of frequency components to fit into the available bandwidth

$$g(t) = \frac{1}{2}\cdot c + \sum_{n=1}^{\infty} a_n \cdot \sin(n\omega t) + \sum_{n=1}^{\infty} b_n \cdot \cos(n\omega t).$$

When $\omega = 2\pi f$; $f = \frac{1}{T}$ cycles/sec.

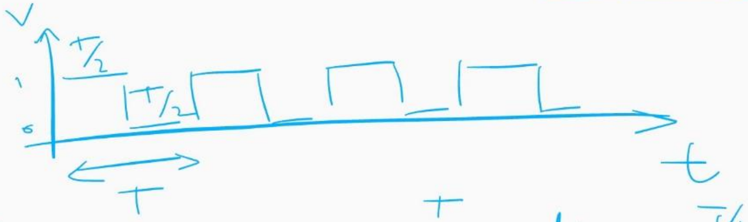
and: $c = \frac{2}{T}\int_0^T g(t)\cdot dt$

$$a_n = \frac{2}{T}\int_0^T g(t)\cdot \sin(n\omega t)\, dt$$

$$b_n = \frac{2}{T}\int_0^T g(t)\cdot \cos(n\omega t)\, dt$$

Note that $\frac{1}{2}\cdot c$ is the d.c. term and the rest, sine & cosine frequencies

Consider a typical discrete signal.



For this, $c = \frac{2}{T}\int_0^T g(t)\, dt = \frac{2}{T}\int_0^{T/2} 1\cdot dt$ $\boxed{= 1.}$

$$a_n = \frac{2}{T}\int_0^T g(t) \sin(n\omega t)\, dt$$

$$= \frac{2}{T}\int_0^{T/2} 1 \cdot \sin(n\omega t)\, dt = \frac{2}{T}\left[-\frac{\cos n\omega t}{n\omega}\right]_0^{T/2}$$

$$= \frac{1}{n\pi}\left[1 - \cos n\pi\right] \;//$$

obtained by substituting $\omega = 2\pi f = \frac{2\pi}{T}$.

Similarly we can obtain

$$b_n = \frac{2}{T}\int_0^{T/2} 1\cdot \cos n\omega t \, dt = \frac{1}{n\pi}\cdot \sin n\pi$$

Therefore,

$$g(t) = \frac{1}{2}c + \sum_{n=1}^{\infty} a_n \cdot \sin n\omega t + \sum_{n=1}^{\infty} b_n \cdot \cos n\omega t$$

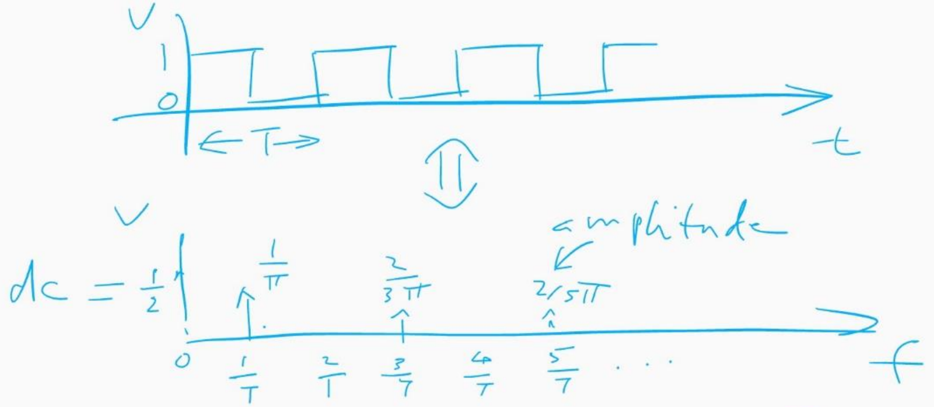
$$= \frac{1}{2}\cdot 1 + \sum_{n=1}^{\infty} \frac{1}{n\pi}\cdot[1-\cos n\pi]\cdot \sin(n\omega t) + \sum_{n=1}^{\infty} \frac{1}{n\pi}\cdot \sin n\pi \cdot \cos(n\omega t)$$

is the Fourier series representation for the binary (discrete) signal above.

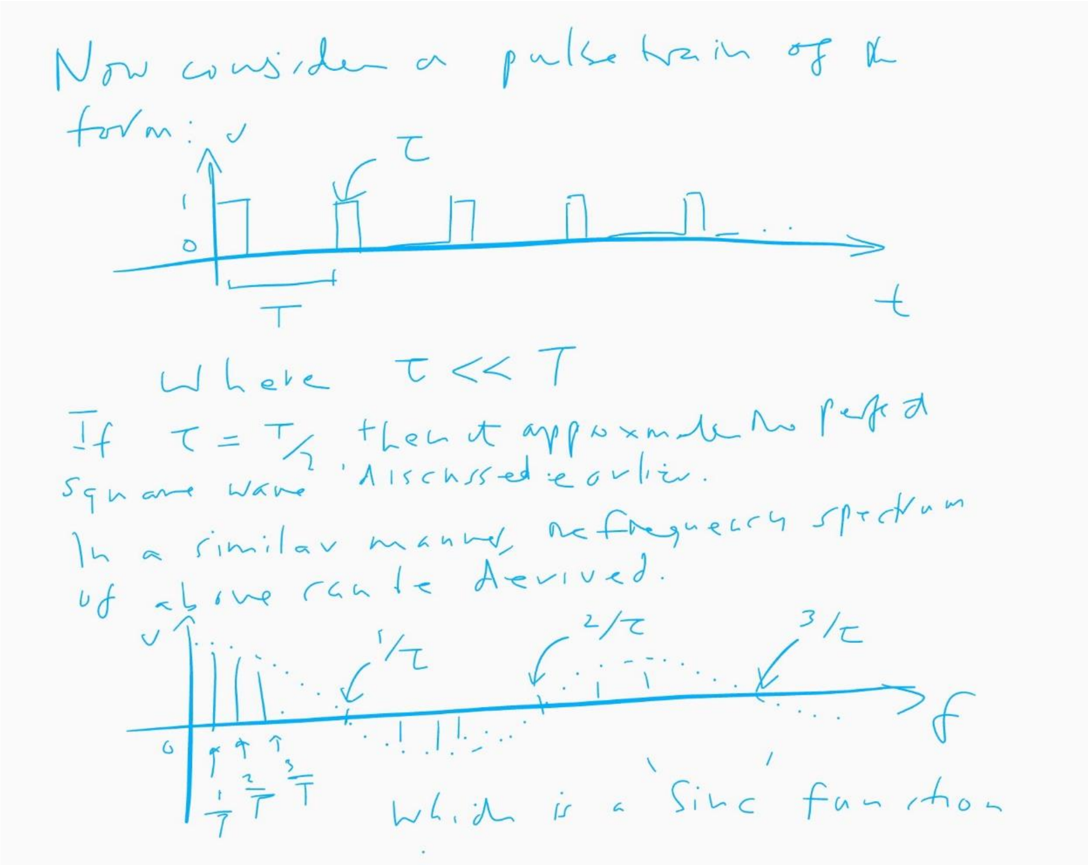
All $\cos(n\omega t)$ terms are zero for $n = 1 .. \infty$

Thus: $g(t) = \frac{1}{2} + \frac{1}{\pi}\left[\sin\omega t + \frac{2}{3}\cdot \sin 3\omega t + \frac{2}{5}\sin 5\omega t \; . . . . \right]$

Freq spectrum $\triangleq$ dc + odd harmonics.



↑ freq components appear at odd harmonics of the fundamental frequency $f_0 = \frac{1}{T}$.



- Exercises:
  - For the pulse train given, obtain the Fourier series representation
  - What happens to the reconstructed signal if higher harmonics are truncated?

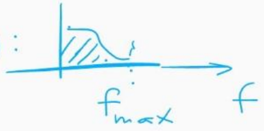
- continuous time to discrete time and vice-versa conversion
  - Continuous time -> discrete time is typically by sampling followed by quantizing each sample
  - Discrete time -> recovery of original continuous time signal is by low pass filtering (subject to Nyquist sampling condition)
- multiply continuous time signal by a sampling signal, quantize each sample (assign a bit length representation for each signal amplitude)
- at the receiver, reverse the process using DAC and Low pass filter; resultant replicated frequency spectrum

Discretising an analogue signal

↓ sampling; ie. multiply $f(t)$ by a pulse train $s(t)$ whose $\tau \simeq 0$

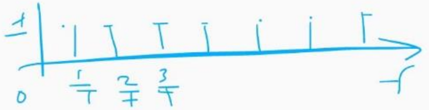
Suppose the frequency spectrum of

$f(t) \longrightarrow F(\omega)$:



and since

$s(t) \longrightarrow S(\omega)$:



then the spectrum of $s(t) \times f(t)$ would look like

which is a replicated version of $F(\omega)$ over $S(\omega)$.

Here, sampling rate $S = \frac{1}{T}$ samples per second.

Nyquist’s sampling theorem

- Digitisation of analogue signals
  - First, band limit the analogue signal to fmax
  - Sample the analogue signal at rate S samples per second
  - Quantize each analogue sample as n-bits per sample (i.e., 2^n discrete levels per sample)
  - frequency spectrum of the sampled analogue signal looks like a repeated pattern of original analogue spectrum at each S frequency component
  - To avoid ‘aliasing’ (overlapping of spectra), it can be seen that the sampling rate should be S > 2fmax.

The replicated version of $F(\omega)$ over $S(\omega)$ should not exhibit spectrum overlaps. That is,

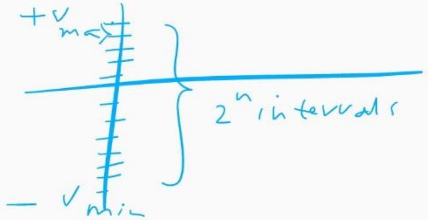
$$\left(\frac{2}{T} - \frac{1}{T}\right) \geq 2f_{max}\text{; interval of separation.}$$

or,

$$S \geq 2f_{max}.$$

Quantisation

Each analogue sample can be quantised to n-bits ($= 2^n$ levels).



- Exercises
  - Find out a mistake in one of the diagrams here
  - What happens if S < 2fmax?

- bits/second (bit rate) = sampling rate or baud rate (samples/sec) x quantisation (bits/sample)
- Suppose S = 2fmax, then the raw bit rate of the digitized analogue signal is n.S (bits per second)
- Simple example of PSTN PCM coding
  - at 4kHz voice, the sampling rate is 8000 samples /sec
  - For typical voice, 8 bits/sample suffices
  - So, the raw PCM bit rate is 64kbps. For example in ADSL to the home, the bandwidth restriction is removed and a higher data rates in the order of a few Mbps is achieved

- Now consider data signals.
- when encoded as a voltage, signal looks like 1 bit per sample, and 1 sample in each T seconds (= 1/T samples per second)
- Since bit rate (bits per second) = bits per sample x samples per second, for any given fixed sample rate (= Baud rate), we can increase the bit rate by increasing the number of bits/sample encoding
- For example, we can group 2 bits per sample, leading to 4 distinct quantisation levels (voltage, phase as in QPSK, wavelength as in WDM etc).
- Bit rate will increase but the down side is the increased quantized levels will interfere with the channel noise

Typical 'data' signal:

1100101100110111110010 . . . → t

Here, we could take one bit per sample; 2 bits per sample, 3 bits per sample etc in 'bit groups'.

That is

1, 1, 0, 0, 1, 0, 1, 1, 0, 0 . . . .

or 11, 00, 10, 11, 00, 11, 01, ...

or 110, 010, 110, 011, 011, ... etc

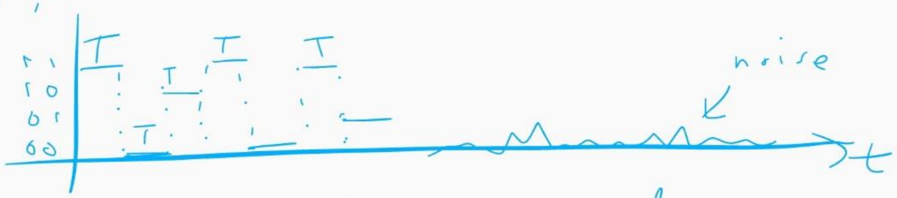
and encode each bit group as one voltage level

Let each 'bitgroup' be a sample of T (sec) duration.

So,

1 – bit per sample.

or,



2 – bits per sample.

With 2-bits per sample, we have 4 levels of quantisation (00, 01, 10, 11).
All the time T is fixed, whether 1 bit or multiple bits. Noise can interfere, when multiple quantisations exist.

Shannon’s noisy channel capacity theorem

- Says that the maximum bit rate that can be supported by a channel of bandwidth W with a signal to noise of SNR is W.log2 (1 + SNR)
- Signal quality of a channel is measured by the Signal to Noise Ratio (SNR) given in deci Bels (dB) a logarithmic value.
- For example, if SNR is 30dB, then 30 (dB) = 10 log (base 10) SNR, where SNR is 10^3 or 1000 in linear terms
- Therefore for a 4 kHz channel with 30dB SNR, the max. bit rate is around 40kbps according to Shannon’s theorem

- Aligning with the Nyquists sampling theorem, with a minimum sampling rate of 2x4k samples per second, to reach a bit rate of 40kbps, the maximum quantisation allowed is (40/8 = 5) bits per sample encoding
- this gives the allowed number of quantisation levels as 2^5
- Therefore for a 4KHz channel, in the presence of a noise level of 30dB, we cannot quantize at 8 bits/sample, but only at 5 bits/sample, just to avoid the discrete signal being corrupted by noise, and to recover the bits correctly at the receive end

Shannon's noisy channel theorem:

$$\text{max bit rate} = W \cdot \log_2(1+SNR).$$

$$\cancel{30}\,3 \text{ (decibels)} = \cancel{10}\log_{10} SNR$$

$$SNR \text{ (linear)} = 10^3 \simeq 1023$$

$$\therefore \text{max bit rate} \cong 4\times10^3 \log_2(1024)$$

for $W = 4$ KHz.

$$\therefore \text{max bitrate} = 40 \text{ Kbps} \;//$$

How many bits/sample groupings would this represent?

Using bps = samples/sec × bits/sample

$$40\times10^3 = \underbrace{(2\times4000)}_{\text{Nyquist}} \times \text{bits/sample}$$

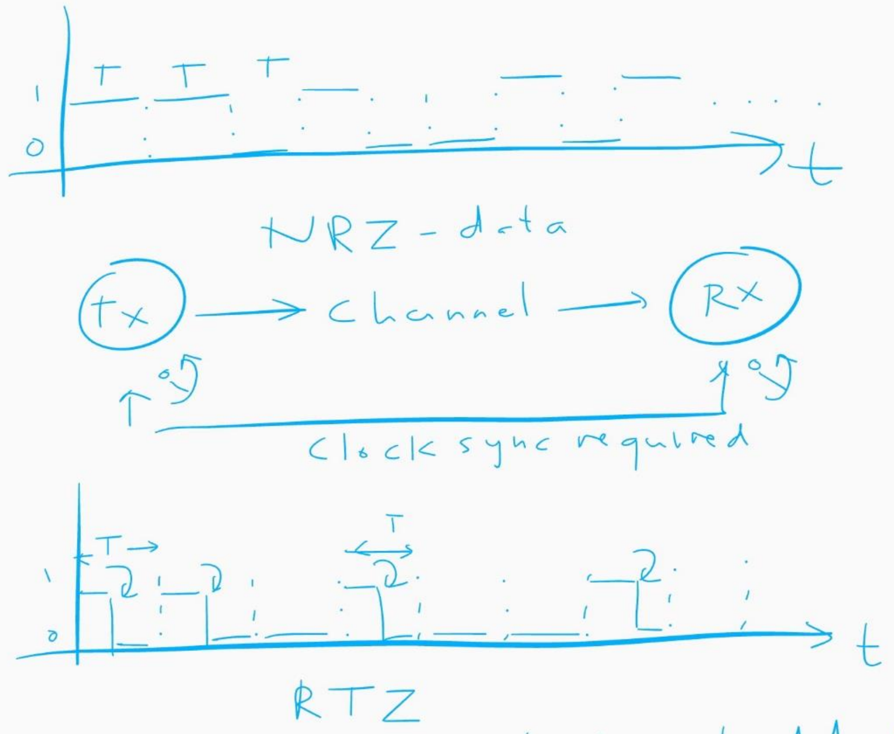
$$\therefore \text{(max) bits/sample} = 5. \;//$$

- Exercises
  - In some text books Shannon's theorem for the noisy channel and Nyquist's theorem for the noiseless channel is presented separately. Here we do not do so. Try to reason why.
  - With the sampling rate fixed, why cannot the number of quantisation levels be increased indefinitely to achieve a higher bit rate?

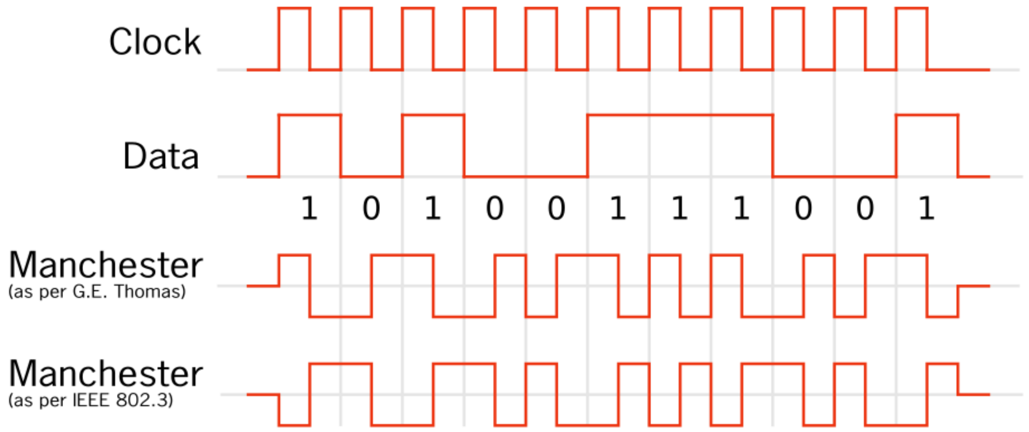
Signal Encoding

- How to condition information to be sent over a channel?
- a simple data stream can be encoded in to a voltage signal (to be transmitted over wire) in the form of a NRZ (non-return-to-zero) format
- main problem with this encoding is that it has lost clock information; when a series of 11…1 are transmitted there is no (1->0) transition that can be detected by receiver, in order to lock on to the signal
- For the correct sampling of the data sent by the transmitter, the receiver requires clock synchronization
- self-clocking information can be embedded if NRZ is converted to RTZ (return-to-zero)

- Manchester code is an example of a self clocking signal encoding; also its spectrum has no dc component as it is bipolar (+/-v)
- if self-clocking is embedded in data, it is called ‘asynchronous’, else if data is followed by special clock pattern, then it is called ‘synchronous’
- No free lunch – when clocking is embedded, the encoded signal has twice the sampling rate due to halved sampling width (each sample width is now T/2, and not T), so it requires twice the bandwidth as the original



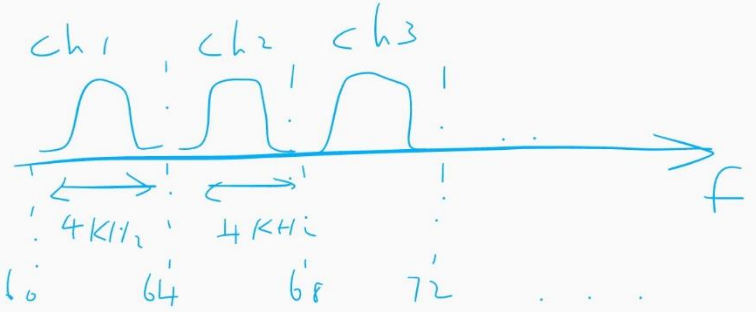
· Imperfect, but clock embedded
· occupies double the bandwidth



Signal multiplexing

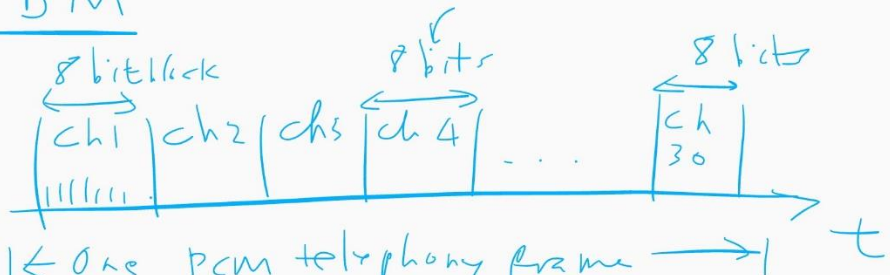
- combining signals to make full use of the available wide bandwidth of ‘trunk’ infrastructure
- FDM, TDM and statistical TDM
- FDM – analogue signal channels are spaced over in ‘raised blocks of frequency’
- TDM – digital signal channels are spaced over in time with a distinct slot per channel of 125 microsecond duration containing 8 bits of that channel
- e.g., 30 channels of (equivalent 4kHz analogue voice) each 64kbps data stream can be combined to obtain a 1.92 Mbps trunk
- since clocking is not embedded, 2 additional channels with clocking/signalling is added, so this becomes a ‘synchronous’ transmission
- called a 30B+2D PCM trunk with a data rate (64 x 32 =)2.048 Mbps

FDM



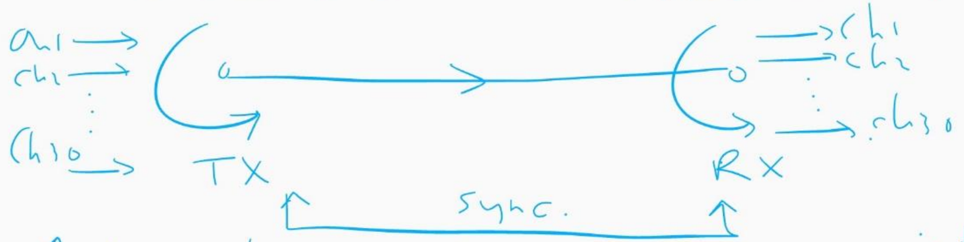
Typical 4KHz voice channels combined on a 'moved up' frequency axis. Broad bandwidth can carry many multiplexed channels.

TDM



|← One PCM telephony frame →|
with combined data rate = 30 x 64kbps
= 1.92 Mbps

8 bits including sync. bits per channel are extracted from each 64 Kbps channel.



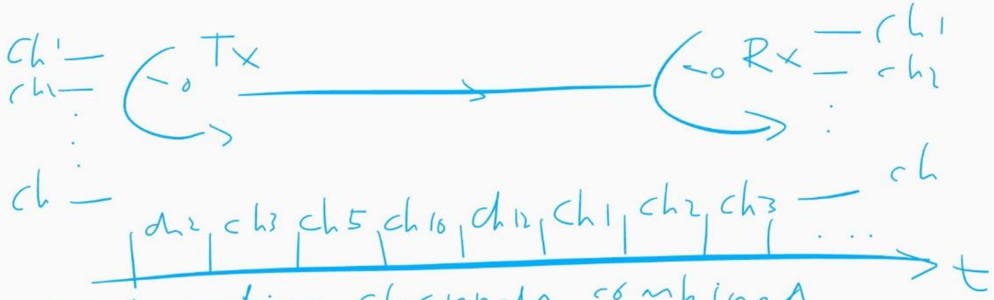
A hierarchy of such mux levels can exist.

- statistical TD multiplexing – consider two broad traffic types ‘digitized voice/video’ and database query/web browsing
- browsing/query is intermittent whereas the other is continuous in time
- if data flows continuously, can allocate channels incrementally (i.e., TDM) and if data is intermittent, in order to avoid waste of empty slots, channels should be randomly allocated based on the activity level (i.e., statistical multiplexing)
- in TDM, both Tx and Rx know that channels rotate in sequence
- in statistical multiplexing only way to inform Rx of the identity of the data slot, is by attaching a ‘label’ to the data ‘packet’; this label is the ‘address’ and forms the basis of packet switching

Statistical TDM

Wasteful resource to allocate channel slots, if corresponding channels are inactive in TDM.

Traffic types
- Streaming data → voice/video
- Intermittent data → dB query/browsing

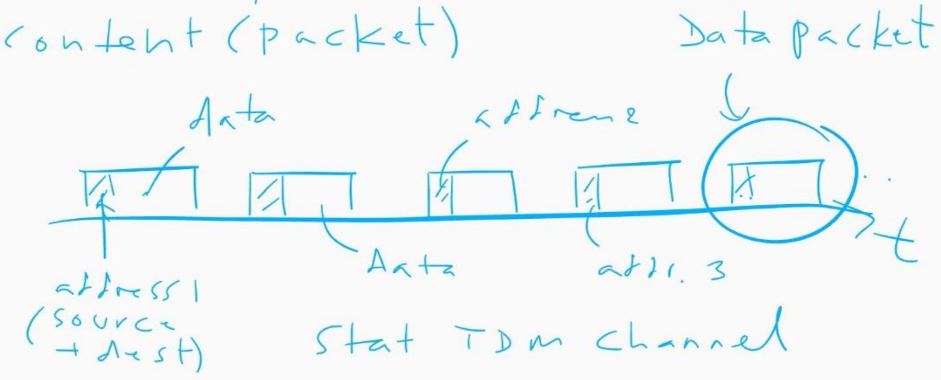


· only active channels combined
· statistically combined TDM

Stat. TDM gave rise to packet switching.

Why?

Without synchronisation possible as in TDM, only way for Rx to identify channel content was by a label, or "address" of the content (packet)



Stat TDM channel

Error detection and correction

- 3 modes of communication – simplex (one way only) , half duplex (one way at a time), full duplex (both ways simultaneously)
- error correction applies for simplex: receiver must self-correct as it cannot request for repeat transmits
- military and space communications as simplex scenarios
- a typical data packet can contain a linear, cyclic or polynomial codes based code block to automatically correct data errors
- in duplex, error detection and request for retransmission can be used; simple parity check as a 1-bit error detection code
- More the redundancy built into data, easier to detect/correct errors

Tx $\longrightarrow$ Rx

one way transmission (simplex)

Tx $\rightleftarrows$ Rx

two way transmission (duplex)

Error recovery

- Error detection by Rx (& ask Tx to resend)
  - duplex OK
- Error correction by Rx
  - if Simplex only.

Error Detection

· simple parity bit

data	triplicate	add odd parity
0	000	000:1
1	111	111:0

send (Tx) $\longrightarrow$ ch $\longrightarrow$ Rx receive 0101 $\longrightarrow$ odd parity violated (Errors detected)

noise corrupts bits

Parity bit can detect only 1-bit errors, not multiple bit errors.

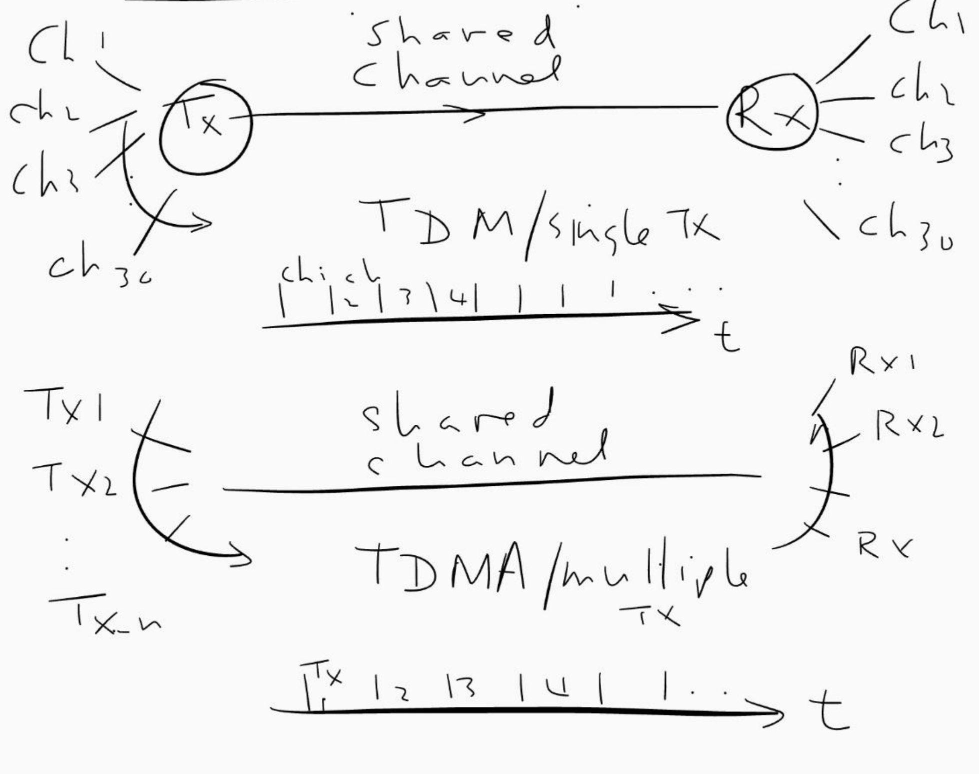
- Exercises
  - Compare Manchester encoding of a data signal with that of HDB3 encoding. What are the differences?
  - What is the significance of a signal encoding having zero dc component?
  - In the 30 channel TDM/PCM scheme, how do we get a channel width of 125 microseconds?

- Exercises
  - Another way to classify traffic is real time traffic vs. store-and-forward traffic. Where would voice/video fall?
  - If we use TDM for store-and-forward traffic, the overall trunk utilization efficiency will fall. Discuss.
  - Show that the 3-bit code shown in the diagram can not only detect errors, but also can correct errors.

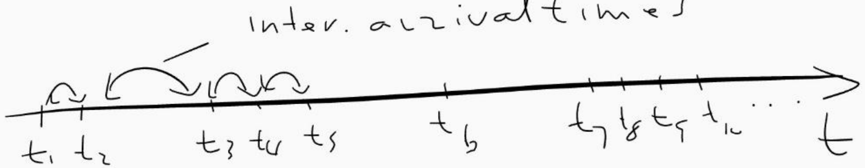
Multiple access schemes

- Allocating a common or shared channel resource (time, bandwidth) among many competitive senders and receivers
- Scheme depends on the traffic type: if traffic is streaming like (e.g. 64kbps -> 128 packets/sec of 64 byte packets) then it is fair to assign a permanent slot in time or frequency for that channel (e.g. TDM/single Tx or TDMA/multiple Tx, for slotted time)
- Granting a permanent slot (like the priority bus lane on a shared road) is a *static* resource allocation method

For Streaming data



Poisson traffic model



If inter-arrival times are exponentially distributed, i.e.,

$$pdf(\text{inter-arrivals}) = \lambda . e^{-\lambda t}$$

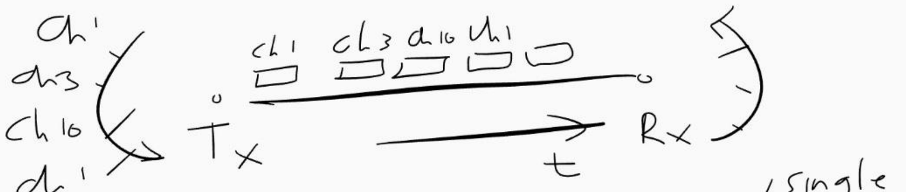
where $\frac{1}{\lambda}$ is mean inter-arrival duration,

then we say, the number of arrivals "k" within an interval $[0, T]$ is given by

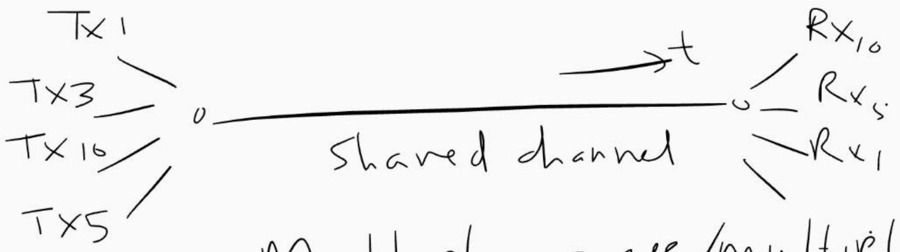
$$\frac{\lambda^k . e^{-\lambda}}{k!}$$; which is a Poisson pdf.

- alternative to static channel allocation is *dynamic* channel allocation, i.e., allocate on demand
- data traffic (like those ~30 years ago) had a Poisson arrival process (inter arrival times exponentially distributed leading to the number of arrivals in an interval (0,T) being Poisson distributed)
- such traffic is best assigned the resources, on demand
- Statistical TDM studied earlier is an 'on demand' allocation method for single Tx case
- For multiple Tx, dynamic channel allocation can be done *deterministically* (Token rotating in round robin) or *probabilistically* (random access, with a chance of collision)

For data with Poisson features



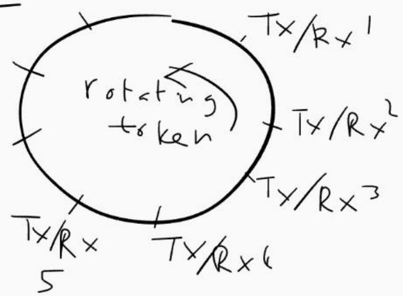
Statistical TDM/ single TX



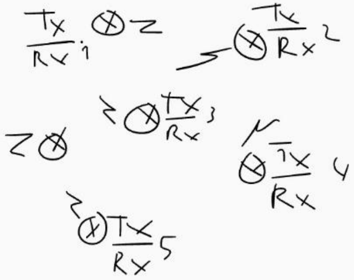
Multiple access/multiple TX
on-demand
channel allocation

On demand channel allocation

deterministic access
Eg: Token Ring



random access
Eg: Aloha & variants

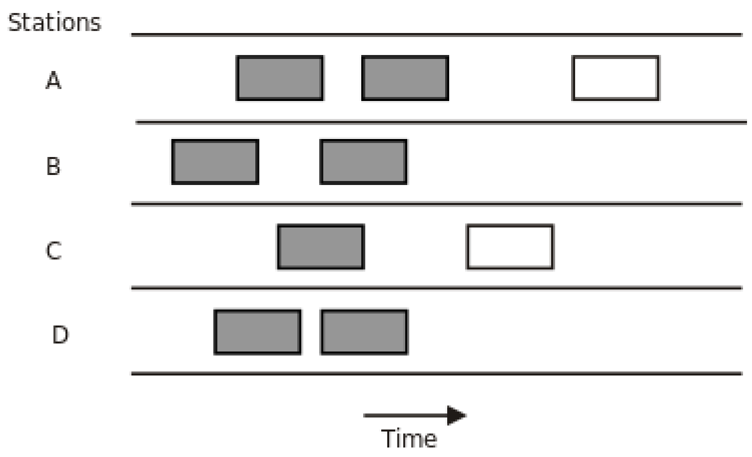


- probabilistic schemes: aloha, carrier sense multiple access (CSMA) and variants like CSMA/CD, CSMA/CA…
- fundamental principle in probabilistic schemes: listen before talk as it takes finite time for signal propagation and to avoid collisions

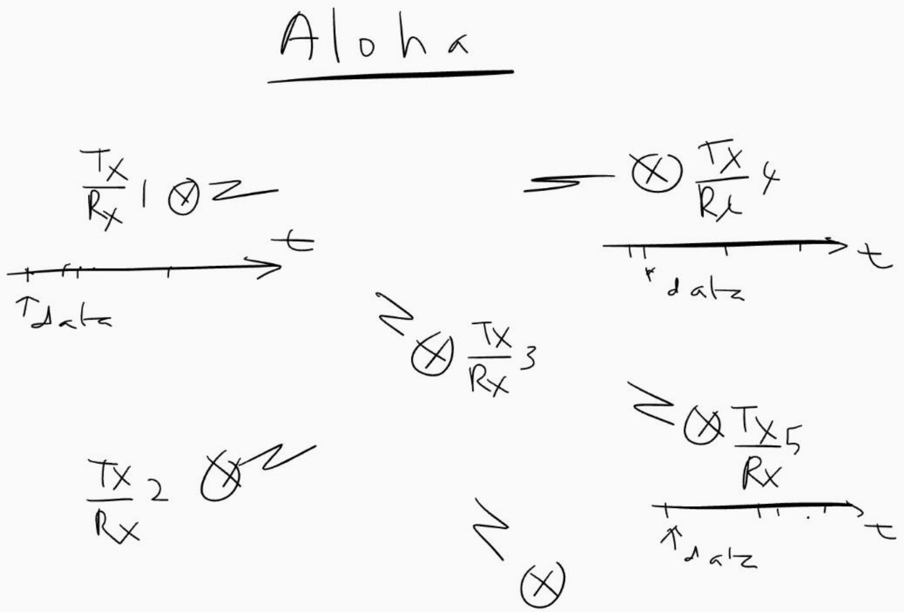
- Exercises
  - Dynamic or ‘on demand’ channel allocation is more suitable for Poisson like data traffic. Here too, we have two options: deterministic or probabilistic channel access. Under heavy traffic conditions deterministic access is preferred to probabilistic access. Why?

Aloha

- One of the earliest attempts in packet data communications (1970’s)
- fixed multiple radio Tx/Rx units within overlapping radio coverage areas broadcasting packets at arbitrary time instances
- Packets are generated at random intervals (inter packet gaps = exponentially distributed) at each Tx
- Protocol
  - If Tx has data to send, send data packet
  - If Tx, while transmitting data, also receives data from another Tx, then a ‘collision’ has occurred; re send data



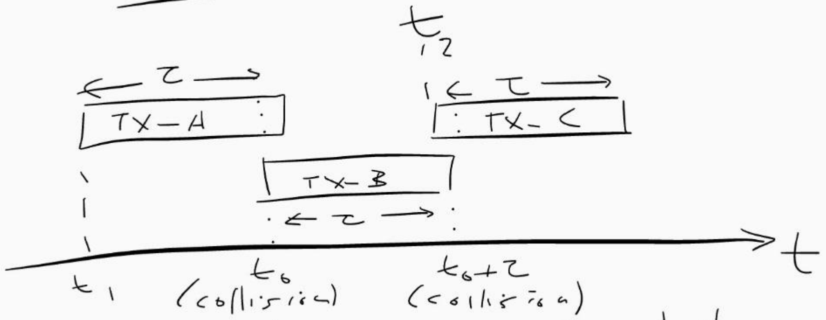
Stations
A
B
C
D
Time



Aloha
Tx
Rx
1
2
3
4
5
t
Tdata

- Suppose the packet transmit time is tau (a constant), then the *vulnerable period* is 2.tau (see explanation)
- Suppose the total traffic load on the shared channel is G (packets per tau interval), then G is distributed with a Poisson pdf
- Within a (0, 2tau) interval the total load is 2G and that the probability [k=0 colliding events in (0, 2tau)] = e^-2G.
- Therefore, the carried load (or the throughput) = S = G.e^-2G. Plot of Aloha shows that maximum efficiency is around 20%, and this is due to too many collisions caused by transmitting haphazardly

Vulnerable period (Aloha)



For TX-B packet transmitting at t_0:

(i) if TX-A starts at $t_1 > t_0 - \tau$

or (ii) if TX-C starts at $t_2 < t_0 + \tau$

TX-B packet will collide with either TX-A or TX-C.

We define vulnerable period of TX-B as $t_2 - t_1 \simeq 2\tau$.

Generally on any probabilistic shared access channel:

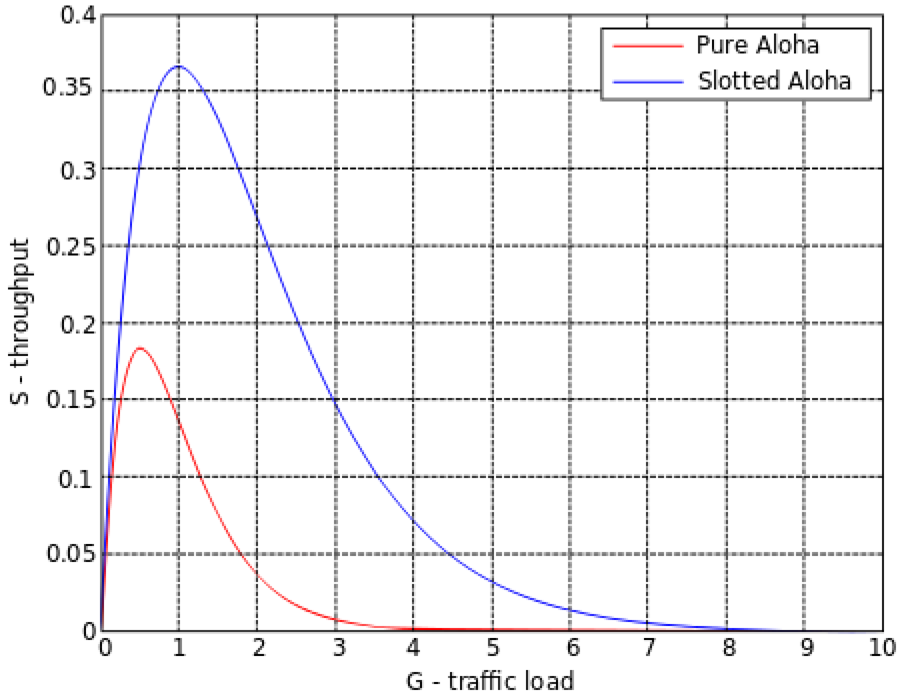
$\boxed{\text{total load} = \text{carried load (good throughput)} + \text{wasted load (retransmissions)}}$

(packets per unit time)

Suppose the total load 'seen' on Aloha channel $= G$ (packets per τ), which should be Poisson distributed.

That is: $\text{pdf(load)} = \dfrac{G^k \cdot \exp(-G)}{k!}$

$= \Pr[k \text{ events in } (0, \tau)]$.



Since the load seen within a 2τ-vulnerable period is $2G$, we have:

$$\Pr[K=0 \text{ events in } (0, 2\tau)] = \left. \frac{(2G)^K \cdot e^{-2G}}{K!} \right|_{K=0}$$

$$\Pr[\text{no collisions}] = e^{-2G}.$$

$\therefore$ The fraction of total load G which suffers no collisions

$= G \cdot e^{-2G}$

$=$ good throughput

$=$ carried load

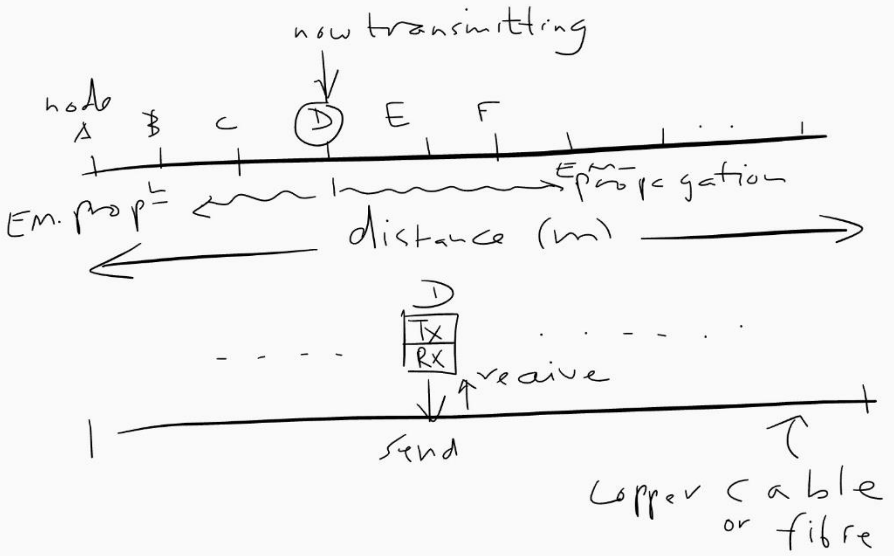
$= S.$ //

- Exercises
  - (pure) aloha is a fundamental model of probabilistic multi access schemes. Prove that its maximum throughput is 1/2e and that occurs at G=0.5, as confirmed by the graph.

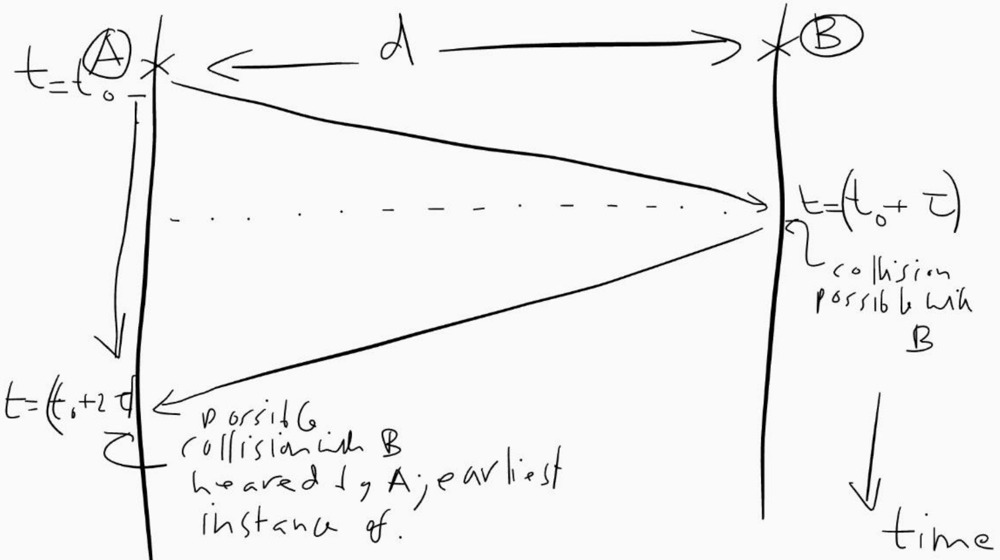
CSMA/CD principles

- Carrier Sense Multiple Access with Collision Detection
- Carrier sensing (listen before talk) can improve efficiency; stopping when a collision is detected can improve further
- CSMA/CD is used in Ethernet (IEEE standard 802.3)
- Protocol on a bus based shared medium:
  - If medium 'silent' then send
  - Listen while sending; if collision detected stop sending and retry later
- Suppose the distance between two nodes A and B is d (meters) then, time for A's first bit to reach B is (tau = d/c), where c=EM propagation velocity; if B, sensing the channel idle, has also been transmitting during this time, the first bit of B will collide with first bit of A at t=tau; this collision will be heard by A, only after a further tau time, that is at (2.d/c)

CSMA/CD on a bus based shared channel



Propagation diagram for CSMA/CD



$\therefore$ vulnerable period of A = 2τ

CSMA/CD

According to the propagation diagram 'A' should transmit for at least 2τ (where $\tau = \frac{d}{c}$) time to properly detect any collision with 'B'.

Now, $\tau(\text{sec}) = \dfrac{\text{distance between A \& B } (d)}{\text{E-m Velocity on media } (c)}$

If data transmit rate of A is R (bps) then,

packet length corresponding to 2τ

$= R.2\tau = R.\left(\frac{2d}{c}\right)$ bits. = minimum packet length.

- Exercises
  - CSMA/CD is well studied because of its Ethernet origins. For a copper cable of length 1km acting as the shared medium, with each node able to transmit data at 3Mbps, what is the permissible minimum packet length? Assume c=2x10E8 m/s.
  - What is the total time wasted in the case of a worse case collision?

- In other words, A's packet length should be at least R.(2d/c), in order that A detects a possible collision with B, where R is the source transmission rate
- On a collision, A and B both will stop transmitting and will only re-start to sense the channel after a random interval (exponentially lengthening after each successive collision)
- Ethernet technology is widely used starting from 3Mbps, 10Mbps etc but now, the shared access resolution protocol is superseded by *switched Ethernet* at 100Mbps, 1Gbps and 10Gbps over copper and fibre
- Under heavy traffic, shared Ethernet has a low efficiency of around 40% but better than aloha