01 – Physical Layer

$C = 2B\log_2(M)$ $(C = 2B$ [2 níveis]$)$

$Baudrate = 2B\left(\frac{symbols}{s}\ ou\ baud\right)$

$Bitrate = 2B\log_2(M) = C$

C – channel capacity
B – bandwidth/ **2B** – baudrate
M – levels used to encode information

$$SNR = \frac{P_r}{N_0 B_c}$$

SNR – Signal to Noise Ratio

$$C = B_c \log_2(1 + SNR)$$

$$N = N_0 B_c$$

Space Loss:

$$\frac{P_t}{P_r} = \frac{(4\pi d)^2}{\lambda^2} = \frac{(4\pi f d)^2}{c^2} \qquad \lambda f = c$$

B_c – bandwidth of the channel (Hz)
P_r – signal power at receiving antenna (W)
P_t – signal power at transmitting antenna
N_0 – White noise (W/Hz)
d – propagation distance between antennas
c – speed of light = 3×10^8 m/s

$P_r = P_t \cdot Ganho\ (em\ W)$
$P_r = P_t + Ganho\ (em\ dB)$ $\qquad P_{dBW} = 10\log_{10} P$
$Ganho = \frac{1}{Atenuação}(em\ W)$ $\qquad P_{dBm} = 10\log_{10}\left(\frac{P}{1mW}\right)$
$Ganho = -Atenuação(em\ dB)$

02 – Delay Models

Queue Models

- Customers (packet to be transmitted through a link) arrive at random times to obtain service (transmit a packet).

$$T_{pac(frame)} = \frac{L}{C} = \frac{1}{\mu}$$

- **M/M/1 Queue** – Poisson Arrival, Exponential service time and Time Division Multiplexing

$$N = \frac{\rho}{1-\rho} = \frac{\lambda}{\mu-\lambda} \qquad T = \frac{1}{\mu-\lambda}$$

$$T_W = T - T_S = \frac{1}{\mu-\lambda} - \frac{1}{\mu} = \frac{\rho}{\mu(1-\rho)}$$

$$N_W = T_w\lambda = \frac{\lambda}{\mu-\lambda} - \frac{\lambda}{\mu} = N - \rho$$

- **M/M/1/B Queue** – this queue has limited capacity (B buffers). Packets can be lost, and the probability of packet being lost = P(B) -> Queue is full

$$\sum_{i=0}^{B} P(i) = 1 \qquad P(n) = \rho^n P(0)$$

$$P(0) = \frac{1-\rho}{1-\rho^{B+1}} \qquad P(B) = \frac{(1-\rho)\rho^{B}}{1-\rho^{B+1}}$$

Probabilidade de perder dados: $P(B) = \frac{(1-\rho)\rho^B}{1-\rho^{B+1}}$

$$se\ \rho = 1 \rightarrow P(B) = \frac{1}{B+1}$$

$$se\ \rho \gg 1 \rightarrow P(B) = \frac{\lambda-\mu}{\lambda}$$

- **M/D/1 Queue** – Used when packets have constant size.

$$E[X] = \frac{1}{\mu}; E[X^2] = \frac{1}{\mu^2}$$

$$T_w = \frac{\lambda}{2\mu^2(1-\rho)} = \frac{\rho}{2\mu(1-\rho)}$$

- **M/G/1 Queue**

Chegadas->Poisson; Attend -> Arbitrário

Tempo de espera médio: $T_w = \frac{\lambda E[X^2]}{2(1-\rho)}$

$$N = \lambda T = \lambda\left(T_w + \frac{1}{\mu}\right) = N_w + \rho$$

- **D/D/1 Queue** – chegadas e atendimentos seguem distribuição determinista

A fila M/M/1 tem um nr médio de pacotes N superior ao da fila D/D/1.

Kendall notation ➔ **A/S/s/K**

» A – arrival statistical process
» S – service statistical process
» s – number of servers
» K – capacity of the system in buffers (assumed ∞ if omitted)

02 – Data Link Layer

Main functions: provide service interface to the network layer by transferring data from the source's network layer to the network layer on the destination's machine, eliminate/reduce transmission errors, regulate data flow – slow receivers not swamped by fast senders.

Framing consists of breaking up the bit stream into discrete frames, computing a short token called a checksum for each frame, and including the checksum in the frame when it is transmitted. When a frame arrives at the destination, the checksum is recomputed. If it is different from the one contained in the frame, the data link layer knows that an error occurred. To find errors there's **Byte Count**, **Byte Stuffing** and **Flag Bits with Bit Stuffing**.

No Error Probability: $P = (1-p)^n$
Error Probability: $P = 1-(1-p)^n$
i Error Probability: $P = \binom{n}{i}p^i(1-p)^{n-i}$

p = bit error probability
n = frame length

Técnicas de deteção de erros:
Parity Check – 1 bit de paridade adicionado a k bits de informação (d=2)
Bi-dimensional Parity – 1 bit por linha e 1 por coluna (d=4)
Internet Checksum – (d=2)
Cyclic Redundancy Check (CRC) – nr fixo de check bits (d>3)

$$FER = 1-(1-BER)^n$$

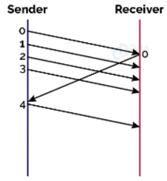
$$T_p = d * \tau_a = \frac{d}{V} \qquad a = \frac{T_p}{T_f} \qquad T_f = \frac{L}{R}$$

Débito máximo: $R_{MAX} = S * (R\ [ou\ C])$

T_f – tempo de transmissão (ms) **L** – tamanho da trama (bits) **T_p** – tempo de propagação (ms) **d** – distância (km)
τa – atraso de propagação (μs/km) **S** – eficiência **R (=C)** – data rate (bits / s) **d** – nr min de erros nec. para erro ñ ser detetad**o**
N_r – no médio tent. p.transmitir trama c. sucesso **M** – representação em mód. de no de seq. **P_e (=FER)** – prob. de transmissão de trama com erros **ρ (=BER)** – bit error ratio, prob. de um bit ter erro **k** – nr bits necessários para codificar W tramas

Sliding Window

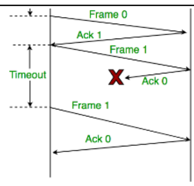
$Se\ W \geq 1 + 2a$:
$U = 1$
$Se\ W < 1 + 2a$:
$$U = \frac{W}{1+2a}$$



Stop & Wait ARQ

$$N_r = \frac{1}{1-P_e}$$

$$S = \frac{1-P_e}{1+2a}$$



Go-Back-N ARQ

(Pode enviar várias frames antes de receber

$Se\ W \geq 1 + 2a$:
$S = \frac{1-P_e}{1+2a*P_e}$ $\qquad W = 2^k - 1$

Se W < 1+2a :

$$S = \frac{W(1-P_e)}{(1+2a)\cdot(1-P_e+W\cdot P_e)}$$

$$W_{max} = M - 1 = 2^k - 1$$

K is number of bits used to code sequence numbers

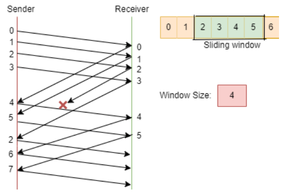
Selective Reject/Repeat ARQ (janelas grandes)

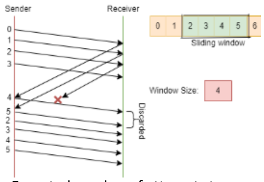
Se $W \geq 1 + 2a$: tamanho máximo:

$S = 1 - P_e$ $\qquad W = 2^{k-1}$

Se $W < 1 + 2a$:

$$S = \frac{W\cdot(1-P_e)}{1+2a} \qquad W_{max} = \frac{M}{2} = 2^{k-1}$$





Stop and Wait: Efficiency (S):

$$S_{max} = \frac{T_f}{T_f + 2\times T_{prop}} = \frac{1}{1+2a}$$

- Efficiency with Errors

$$S = \frac{T_f}{E[A](T_f + 2\times T_{prop})} = \frac{1-p_e}{1+2a}$$

- Probability of k attempts required to transmit a frame with success (pe = FER)

$$P[A=k] = p_e^{\,k-1}(1-p_e)$$

- Expected number of attempts to transmit a frame with success

$$E[A] = \frac{1}{1-p_e}$$

$$RTT = 2*T_{prop} + T_f$$

$$T_a = T_w + T_s = \frac{1}{\mu-\lambda} = \frac{1}{\mu(1-\rho)} = \frac{N}{\lambda} \qquad T_w = \frac{N}{\mu} \qquad T_s = \frac{1}{\mu} \qquad \rho = \frac{\lambda}{\mu} = \frac{R}{C} \qquad \mu = \frac{C}{L}$$

$$M = \frac{\log\left(\frac{P_b}{1-\rho}\right)}{\log(\rho)}, se\ \rho \neq 1$$

Teorema Little
$N_s = \lambda T_s = \rho$
$N_w = \lambda T_w = \rho N$

$$N = N_w + N_s = \lambda \cdot T_a = \frac{\rho}{1-\rho}$$

$$P_b = \frac{(1-\rho)\rho^M}{1-\rho^{M+1}}, se\ \rho \neq 1 \qquad P_b = \frac{1}{M+1}, se\ \rho = 1 \qquad P_n(T) = \frac{(\lambda T)^n e^{-\lambda T}}{n!}$$

Statistical Multiplexing
$T_p = \frac{L}{C}$

Freq. Div. Multi.
$C_c = \frac{C}{m} \quad T_p = L\cdot\frac{m}{C}$

Time. Div. Multi.
$C_c = \frac{C}{m} \quad T_p = L\cdot\frac{m}{C}$

$C \rightarrow capacidade\ do\ canal\ (kbits/s)$
$L \rightarrow tamanho\ do\ pacote\ (bits)$
$R \rightarrow tráfego\ médio\ (kbit/seg)$
$\lambda \rightarrow taxa\ de\ chegadas\ (pacotes/seg)$
$\mu \rightarrow taxa\ de\ envios\ (pacotes/seg)$
$\rho \rightarrow intensidade\ média\ de\ tráfego(taxa\ de\ utilização$
$T_a \rightarrow tempo\ médio\ de\ atraso\ dos\ pacotes\ (ms)$
$T_W \rightarrow tempo\ médio\ de\ espera\ na\ fila$
$T_s \rightarrow tempo\ médio\ de\ serviço$
$N \rightarrow nº\ de\ clientes\ no\ sistema$
$N_s \rightarrow nº\ de\ clientes\ a\ serem\ servidos$
$N_w \rightarrow ocupação\ média\ da\ fila\ de\ espera$
$V_s \rightarrow pacotes\ em\ processamento$
$V_S \rightarrow pacotes\ em\ espera$
$M \rightarrow nº\ de\ buffers$
$P_n \rightarrow nº\ de\ chegadas\ no\ intervalo\ T$
$P_b \rightarrow nºprob\ de\ bloqueio(perda\ de\ pacotes)$
$T_p \rightarrow$ Tempo nec. para trasm. um pacote
$C_c \rightarrow$ Capacidade canal
m → uma divisão da ligação

Bitrate (Byte/s) = W/Rtt Rtt = tempo de ida + volta

O valor da janela de congestionamento de uma ligação TCP é calculado pelo emissor e pode variar durante uma ligação TCP.

Noise high ➔ low M

high Signal to Noise Ratio (SNR) ➔ high M

L2 usa MACs, L3 usa IPs, L4 usa ports

1-Physical Layer

Converter P(potencia) de mW para dBm(decibel miliwatt) e dBW(decibel Watt)

$P_{dBW}=10\log_{10}P$ $P_{dBm}=10\log_{10}(P/1mW)$

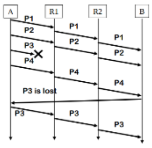
2-Data Link Layer

Main functions:
- » Eliminate/reduce transmission errors;
- » Regulate data flow;
- » Provide service to the network layer;

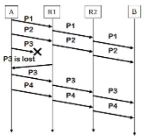
Services provided:
- » Unacknowledged connectionless service;
- » Acknowledged connectionless service;
- » Acknowledged connection-oriented service;

Split this bit stream into frames (sets of bits)?
- » Character (byte) count;
- » Flag bytes, with byte stuffing;
- » Start and end flags, with bit stuffing;

Error detection:
- » Simple Parity check;
- » Bi-dimensional Parity;
- » Cyclic Redundancy Check (CRC);



End-to-End ARQ
(Transport or Application layers)



Link-by-Link ARQ
(Data Link layer)

3-Delay Models

Multiplexing strategies:
- »Statistical Multiplexing; $T_{frame} = L/C$
- »Frequency division Multiplexing; $T_{frame} = Lm/C$
- »Time Division Multiplexing;

Queue model:
- »Customers arrive at random times to obtain service;
- »Customer ➔ packet to be transmitted through a link;
- » Serve a packet = transmit a packet;

Merging Property:
A_i -> Poisson Processes with rate λ_i;
$A = \sum A_i \rightarrow \lambda = \sum \lambda_i$;

Splitting property: Packets arrive to a router according to a Poisson Process (A, λ). They are routed to two output lines with probability p and 1 - p. Packets leaving are still Poisson processes (A, pλ) e (A, (1 - p) λ).

Statistical Multiplexing vs TDM/FDM:
Statistical Multiplexing delay -> $T = \frac{1}{\mu-\lambda}$
Dividing the capacity in m equal portions using TDM or FDM delay becomes-> $T = \frac{m}{\mu-\lambda}$
Queues in Tandem
Case1: Q1= M/D/1 -> Arrival to Q2 is not Poisson; $\lambda_2 < \mu_2$ ➔ $1/\lambda_2 > 1/\mu_2$; no waiting at Q2
Case2: Q1=M/M/1 -> arrival to Q2 strongly related to packet length -> shorter packets will catch up long packets -> Q2 cannot be modeled as M/M/1

4-Mac Protocols

Channel Partitioning:
» Time Division Multiple Access | Frequency Division Multiple Access;

Random Access:

» Pure Aloha:
-No slot concern;
- Stations transmits when it has a frame to transmit;
- If more than one frames are transmitted, they collide and are lost;

» Slotted Aloha:
- Time divided into time slots;
- (Re)transmissions only the beginning of a slot;

» CSMA:
- » If channel sensed free ➔ transmit frame;
- » If channel sensed busy ➔ defer transmission;
- » Usa-se quando $T_{frame} >> T_{prop}$;
- » a=Collision probability
- » Vulnerability time= T_{prop};

$a = T_{prop} / T_{frame} << 1$

Persistent CSMA
- » Medium free ➔ station transmits
- » Medium busy ➔ station waits until medium becomes free, then transmits

Non-persistent CSMA
- » Medium free ➔ station transmits
- » Medium busy ➔ station waits a random time, then repeats algorithm

p-persistent CSMA
- » Slot time = round trip time = $2*T_{prop}$
- » Medium free ➔ station transmits with probability *p* or defers no next slot (1-*p*)
- » Medium busy ➔
  if transmission deferred from previous time slot ➔ same as collision
  else ➔ station waits until medium becomes free, then repeats algorithm

» CSMA with Collision Detection (CSMA/CD):
-Once a collision is detected, CSMA/CD immediately terminates the transmission thus shortening the time required before a retry can be attempted;
- $T_{slot} = 2*T_{prop}$;
- Doesn't use ACK;
- To detect a collision $T_{frame} > 2 * T_{prop}$;

» CSMA with Collision Avoidance (CSMA/CA)
- avoid collisions by beginning transmission only after the channel is sensed to be idle;
- It's an unreliable method;
- It uses ACK;

Taking turns:
- » Polling;
- » Token passing;

5- Network Layer

Datagram network – connectionless service; no circuit concept; packets forwarded using destination host address; packets between same source-destination pair may follow different paths.

Virtual Circuit network – connection-oriented service; circuit establishment ➔ data transfer ➔ circuit termination; path defined from source to destination; router maintains "state" for every established circuit.

Forwarding table entry:
<prefix, mask ➔ gateway, interface>

ARP: Address Resolution Protocol – used to obtain the MAC address associated to a given IP address

TCP usa uma variação do Go-Back-N para recuperar os pacotes perdidos.

5-Transport Layer

TCP congestion protocol:

- Objective
  - » If network congestion decreases ➔ `CongestionWindow` increases
  - » If network congestion increases ➔ `CongestionWindow` decreases
- `Bitrate (Bytes/s) ➔ CongestionWindow/RTT`