**Redes IP**

**Time to live or TTL ->** quantidade maxima de vezes que o IP datagram can transit antes de atingir o destination host.

**Echo request e echo reply ->** (1) Testa o software IP e ICMP do emissor, (2) a conectividade incluindo os routers no percurso entre o recetor e o emissor e vice-versa, (3) o IP e ICMP do recetor

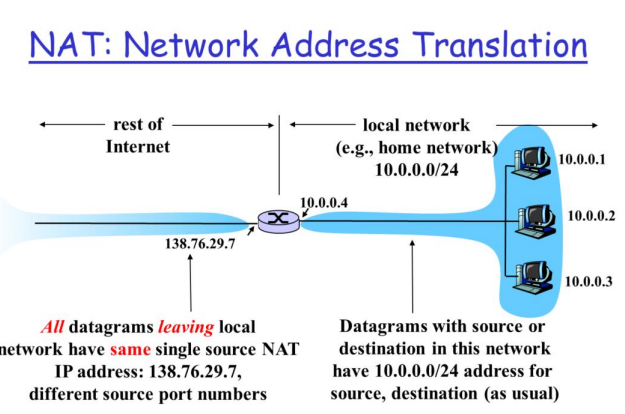
**ARP ->** Todos recebem um ARP request mas apenas o que é suposto comunicar dá reply

**Mac address Table of a switch ->** cada switch tem uma tabela de encaminhamento, em que cada entrada é da forma: (1) Mac Address (2) interface (3) Time to live

**Auto-aprendizagem:** o switch aprende que estações podem ser atingidas por cada uma das interfaces

**Tabela vazia:** **flooding,** após conhecer onde está o destino faz **fowarding**

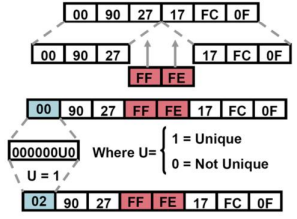
**Nat-DHCP-IPv6**

**Nat ->** todos os datagrams ao sair da local network passam a ter um ip, determinado manualmente

**DHCP ->** Permite atribuir de forma dinâmica IP addresses de uma network a um novo host, após a sua entrada

**IPv6 ->** auto-configuration pode ser stateless ou stateful

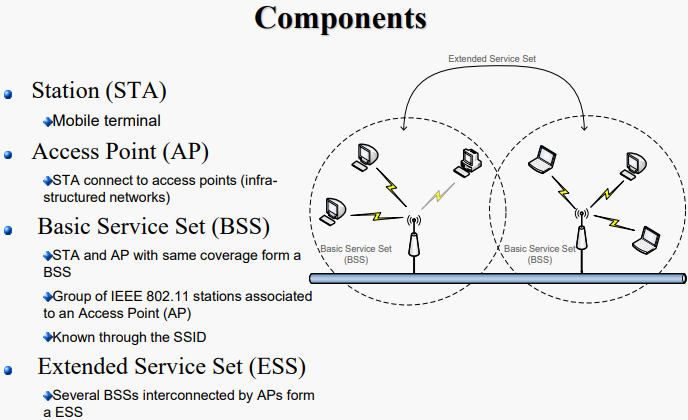
Criaçao do ipv6 atraves do ipv4 exemplo:

****

**Stateless auto-configuration ->**combina local info(ex MAC address) com informação do router (prefixo define a local sub-network)

**Stateful auto-configuration ->** configurado através do DHCPv6

**Ethernet and wireless**

**Components: STA,AP, BSS, ESS**

**Wired vs wireless differences :**

**Ethernet ->** can detect collisions

**Wireless ->** rádios cannot detect collisions:

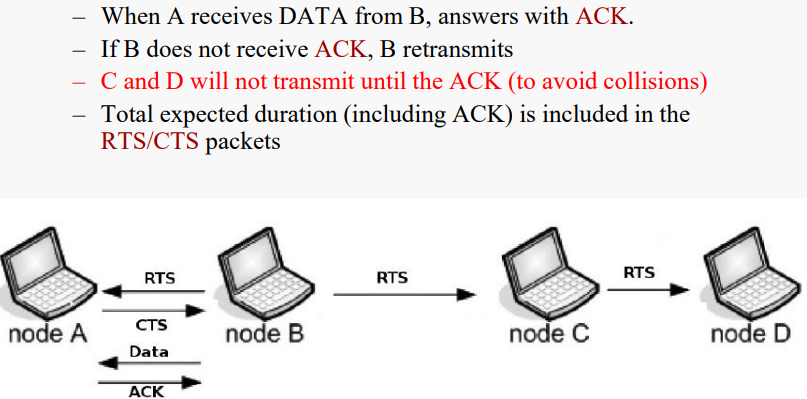
Full-duplex **->** ambos conseguem enviar e receber informação simultaneamente

Half-duplex **->** enviar e receber alternadamente, enquanto um envia o outro só recebe (avoid self-interference)

**MACA:Multiple Access with Collision Avoindance**

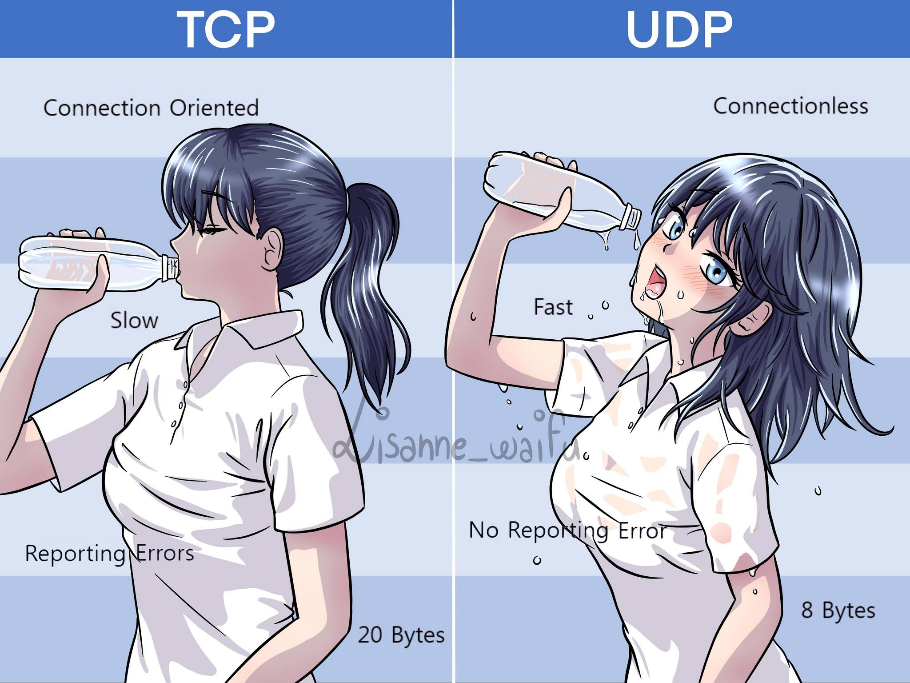
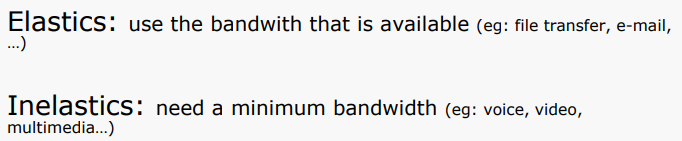
**MACA ->** evita colisões através dos signalling packets RTS (request to send) e CTS (clear to send)

**ACK (**acknowledgements) -> para evitar erros:



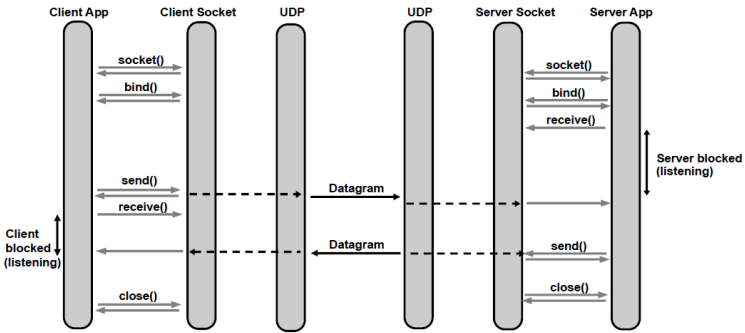
**Application Requirements and Networks**

**Applications ->** Elastics, inelastics

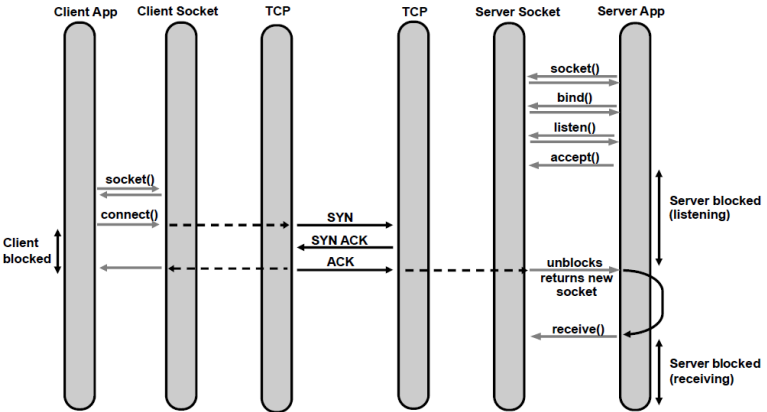


**UDP ->** useless/wet dumbass

**Connection-Less ->** passa direto



**TCP ->** é mais seguro

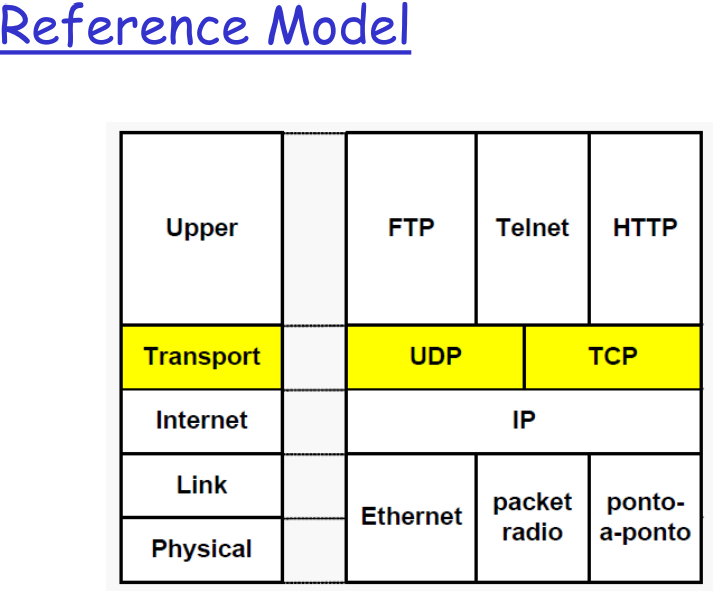
**Connection-Oriented->** precisa de ack 

**Ports and Transport**

**Transport protocols:**

**Sender ->** parte as mensagens em segmentos

**Receiver ->** reconstrói os segmentos em mensagens



**Multiplaxing ->** entregar segmentos à socket correta

**Demultiplexing ->** receber data de varias sockets e por em caixinhas com headers

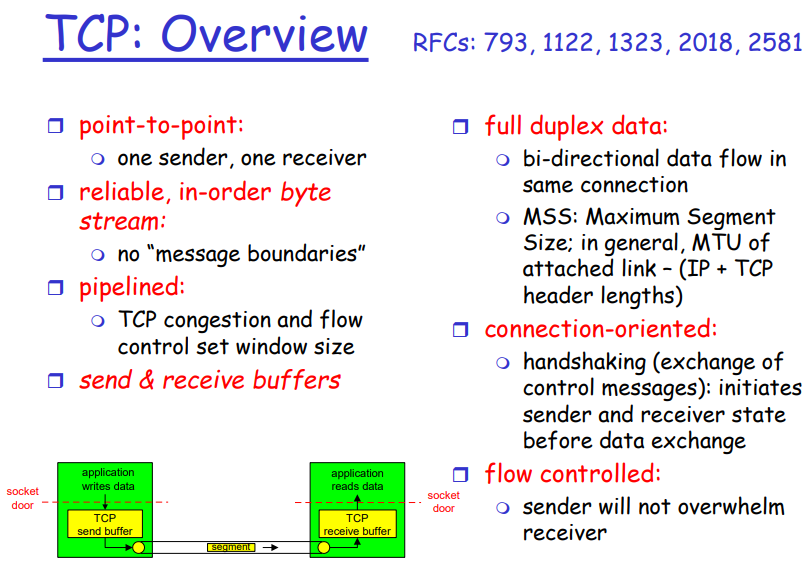
**UDP ->**

**->** performance parecida com ip;

**->** permite troca de dados pelo header e port identifier

**->** permite enviar data para múltiplos destinatários(multi-point communications)

**TCP ->** O protocolo TCP especifica três fases durante uma conexão: estabelecimento da ligação, transferência e término de ligação. O estabelecimento da ligação é feito em três passos, enquanto que o término é feito em quatro.

****

**Sequence numbers (SN) ->**