`

Collision detection at Physical layer
==

A very common problem that is obvious at Physical layer (L1) of OSI model based network communication is data collision between two nodes connected to each other via some sort of physical cable or rarely in wireless mode as well.

let's discuss the collision problem in a typical TCP/IP based communication first...

TCP/IP based communication :

Assume a case where two routers are connected simply via a LAN cable,
LAN cables follow two protocols : Half-duplex and full-duplex,
collision are not possible in full-duplex since their are different wires in cable for Rx and Tx.
In case of half-duplex, collisions are possible.

look at the following scenario :

Network Setup:

> Router A is connected to Router B via a half-duplex LAN cable.
> Router A manages its own LAN with multiple devices (e.g. Device 1, Device 2, Device 3).
> Devices within Router A's LAN send packets to the global internet through Router A, then to Router B, and finally to the global web.

Packet Transmission Sequence:

> Device 1 sends a packet to Router A, which then forwards it to Router B. Router B sends it to the global internet.
> Device 2 then sends a packet to Router A, which is queued to be sent to Router B.
> Meanwhile, a response to Device 1's packet arrives back at Router B, which needs to send this response to Router A.

Potential Collision:

> Simultaneous Transmission: Suppose Router A begins transmitting Device 2's packet to Router B at the same time that Router B is trying to send the response to Device 1's packet back to Router A.
> Since the connection between Router A and Router B is half-duplex, only one device can transmit at a time on the shared medium. If both Router A and Router B attempt to transmit at the same time, a collision will occur.

Collision Handling:

> Collision Detection: Both routers will detect the collision because the signal on the wire will differ from what each router is transmitting. In half-duplex mode, Ethernet uses Carrier Sense Multiple Access with Collision Detection (CSMA/CD) to handle such situations.
> Back-off and Retransmission: After detecting the collision, both Router A and

Router B will stop transmitting. Each will wait for a random back-off period before attempting to retransmit its data. The random back-off period helps to minimize the chance of another collision when they retry.

Here the actual collision detection is done by the router's themselves, they monitors the signal on the Tx wires during transmission, in case of collision, the signals on Tx wires differs from the original signals which the router has sent, thus it may trigger an interrupt and execute some algorithm to tackle the collision and make the system operable again.
At the end, the communication mechanism where collisions are possible must have a dedicated hardware present at the nodes.

CAN bus communication :

CAN communication at its low uses CAN bus which has two wires for both transmission and reception, thus is similar to half-duplex method,
CAN protocol uses single shared bus for communication between all CAN nodes.
CAN protocol follows "differential signaling" mechanism to transmit data.

$$ Differential signaling in CAN protocol :

CAN protocols also follows differential signaling but with different voltage levels for D+ and D-. Instead of D+ and D- , CAN_H and CA_L are used respectively.
Differential signaling is not only for noise cancellation, low power consumption but used for avoiding collisions as well via the mechanism of 'Bus Arbitration.'

Voltage levels for bit transmissions :

~ logical 1 : CAN_H and CAN_L are both kept at same voltage level of nearly 2.5V and thus differential voltage of 0V. this is also called 'recessive bit'.
~ logical 0 : CAN_H is maintained at higher voltage than CAN_L , typically CAN_H at 3.5V and CAN_L at 1.5V. this is also called 'dominant bit'.

In normal state when there is no data transmission and bus is in idle state, then the wires are in recessive state i.e. both wires are at same voltage.
Since CAN protocol uses differential signaling, thus its reliable to Electro-Magnetic Interferences and additional noises.

Now, what if multiple node tries to send the data simultaneously ??? , i.e. how collision will be handled by CAN protocol ???

CAN protocol instead of handling collisions after they have happened, avoids collisions via 'Bus arbitration'.

Let's see how collisions are detected and instantly avoided...

Their are two cases during data transmission, first one is when one node is already transmitting, then all other nodes are in listening state thus can't transmit their own data, and the second case is when two or more node starts transmitting data simultaneously.

In second case, their would be collision, CAN handles collision by the design implementation of its own, each device before sending any data, sends out its Node ID. Now, no two or more device have same node ID, during simultaneous transmission, each of the node continues to check for the signal it has sent, if there is change in signal, this means some other node is also sending the data.
Node ID is of certain bits say 11 bits, so when node ID is sent by two or more nodes simultaneously, say two nodes A , B are sending their IDs as 00000011111 and 00000001111, the integral value of ID(B)>ID(A).
let transmission bit-rate be 1 bps, then at t=0, bit 0 will be send i.e. the bus will be in dominant state at t=0, and remains in dominant state up to t=6, now after t=6, 'A' will try to drive the bus to recessive state i.e. logical 1 , whereas the node 'B' will try to keep the bus in dominant state i.e. logical 0.
The CAN mechanism is designed such that the dominant bit overrides the recessive bit, i.e. even if there is recessive bit present on the bus, once the dominant bit enters, it makes the entire bus in dominant state.
So, when the 7th bit collides, the entire bus will be in dominant state i.e. in logical 0 state, thus 'A' will find this unpleasant since the sent signal and monitored signals are mismatching.
Thus it will stop sending the data further, but B will also detect signal anomaly thus is also supposed to stop data transmission, but this is not the case, actually, the anomaly is detected by both but then both nodes look at the bit they are currently transmitting, if '0' is being transmitted, it continues to transmit since the anomaly happened because some of the node has sent bit '1' and if '1' is being sent and anomaly is detected, the node will stop sending the data since anomaly happened because some other node is sending bit '0'.
Thus the dominant bit overrides the recessive bit and thus the node with lower ID will win the arbitration and continues to transfer the data,
So, A and B both will detect the anomaly and since A is sending recessive bit thus stops sending the data and B is sending dominant bit thus continues to send the data. thus, lower the CAN node ID, higher will be the priority of its CAN frames.

The mechanism via which CAN protocol handles the collision is called "Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA for short)".