- Toh problem kya aayi, agar frame ka frame header hi corrupt hogaya toh aapke pass ek hi option bachata hai ki aap data ko phir se transmit karo.

- If frame length is corrupted, then you have to restart entire communication.

- Kyoki jis frame ka header corrupt hua hai, us frame se lekar aagey saari frame ko sahi tarike se reconize nahi kar payega.

- Pahale hi practical technique fail hogi, ki hum frame length batade, frame k starting mein aur wahi corrupt hogaya, toh uske baad ek bhi frame sahi tarike se pata nahi chalegi ki kaha se start horahai hai aur kaha par khatam horahi hai.

- Now Next toh technique introduced ki gayi thi woh the Character stuffing.

Character Stuffing :->

- Character Stuffing Technique kya kahate hai, Character Stuffing technique assume a particular character to represent frame end.

- Iska kya matlab, hamne assume kiya ki frame length ko assume karta hai 'y'.

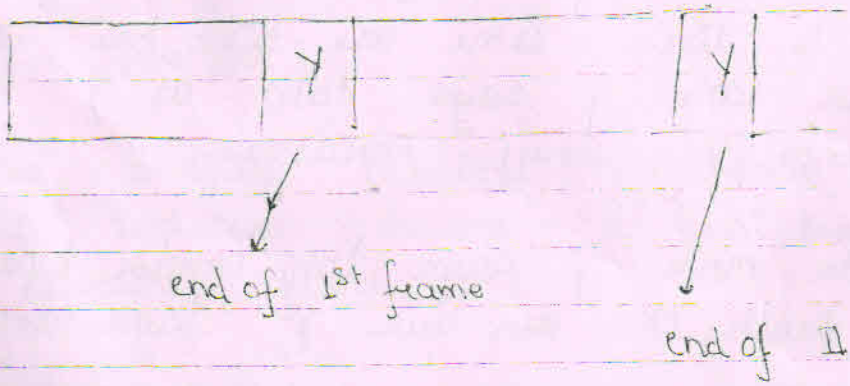
- Toh Sender kya karega jaha frame ka end hua

waha kya append kardega Y, receiver kya karega jab tak y nahi aaya, jab tak woh frame ko read karta rahaiga.

- Aur jaisehi Y aaya, kya assume kar lega receiver ki pahali frame end/terminate ho gayi hai.
- Now Y k baad dusri frame start hogayi hai.
- Toh in Character Stuffing we assume that a particular Character represent frame end.
- Toh hamne example k liye woh particular Character kya le liya 'Y' le liya hai.
- Now aab problem kya hai 'Y' kya data mein nahi aasakta hai kya.
- Aasakta hai aab data mein 'Y' aagaya., toh receiver kya assume karlega, frame end hogi.
- Toh isliye iske liye hum Character Stuffing, Stuffing is liye isme use kiya hai agar data mein agar woh Y aaya, toh us 'Y' k saath ek additional 'Y' stuff karderga.
- Agar data mein Y aaya toh sender kya karega is Y k saath ek ~~aur~~ aur Y laga dega.
- Toh data mein kabhi bhi Single Y kabhi bhi nahi aayega. Single Y kaha aayega frame k end

mein aayega.

- Toh aab Receiver kya karega usko do Y milegy do ka galat nahi milega kya data, bheja toh hamne application layer se ek hi Y tha.
- Reciever kya karega jab bhi do Y dikhe, toh kya usko frame end consider karega, toh nahi. ussi
- Frame end toh tab consider karega single Y pe.
- Agar receiver ko do Y mile toh ek Y ko stuf 'Y' assume karke drop kar dega.
- Aur jaha usko single Y milega wahi uske assume karn karega ki frame end terminate hogayi.
- Toh yeh technique kya kahlayi hamari Character Stuffing.



Character Stuffing

- Character Stuffing mein hum ek particular character ko assume karte hai ki kya, represent karta hai frame ka end.

- Aur frame end par hi woh character dikhega. Agar woh data character data mein aaya toh, aaisa ka aaisa hi woh character append hojayega.

- Toh data mein kabhi bhi single ɣ nahi aayega. Agar aaya bhi toh double ɣ aayega.

- Reciever kya karega double ɣ dekha ek ɣ ko drop kardega kyoki usko pata hai dusra stuffed 'ɣ' hai ki sender ne intensionally diya hoga.

- Aur jaha reciever ko single ɣ mila, waha woh kya assume kar lega ki frame end hogi hai.

- Toh character - Stuffing ka advantage clear hai, agar corruption bhi hua, ɣ ki place par z hogaya.

- Toh aap pahali frame ko sahi tarike se recongize nahi kar paogey, likin aap dusri frame ki end ko recongnize kar paogey.

- Agar ek frame corrupt hua toh uski frame boundary ko reconize nahi karpaogey, par dusri frame ki boundary ko reconize karne mein koi problem nahi hogi.

- Toh pahali wali technique mein kya horaha tha, agar ek frame corrupt hua toh aap saari frame

Recognize nahi kar paraha ho.

- Par yaha ek ka corrupt hua, jiska corrupt hua woh aur uske baad wali, yeh do aap recognize nahi kar paoogy.

- Baad wali ka 'Y' sahi aaraha hai, toh uske baad frames mein aapko koi problem nahi aayegi.

~~Agar first frame corrupt hogayi,~~

~~Toh Character Stuffing~~

- Par Character Stuffing technique mein problem kya hai, agar puri frame mein aapko Y bhejna hai, toh aapki frame length kya hojayegi double, har Y k niche ek Stuff Y, aur agar aapko 10 Y send karne hai toh aapko kitane bhejane pareguy 20 'Y'.

- Joh unit of Stuffing is 8 bit aur bahut jada hai yeh.

- Isiliye hum Character Stuffing bhi use nahi karte hai.

- Joh phir konsi technique use ki jata hai, Bit Stuffing.

Bit Stuffing :→

In Bit Stuffing, a particular bit pattern represents frame ~~length~~. end.

- Jaise hamne decide kiya our bit pattern is 6 consecutive one's.
- Agar 6 one (111111) ek sath aagaye toh kya hoga, toh kya hoga frame end
- Six one ek saath kaise aayegy toh sender frame k end mein six one append kardega.
- Aur jaisehi six one ek saath aaye, reciever kya assume kar lega ki frame end hogayi hai.
- In bit stuffing a particular bit pattern is assumed to indicate frame end.
- Joh abhi filal bit-stuffing ne kya assume kiya six consective one's.
- Jaise reciever ko, six one's dikhe woh kya assume kar lega ki yeh frame end hogayi hai.
- Lekin phir problem hai six one kaha aasakte hai data mein bhi toh aasakte hai toh six one data mein aayegy toh woh usko kaise handle karega.
- Agar aapka bit pattern 6 bit ka hai, toh jaisehi five one aaye uske baad six bit kuch bhi kyo ha ho
- 5 bit k baad woh kya append kardega 0.
- So that kabhi data mein kabhi bhi six one

IT-2004 → 80 → (d) option is correct

aayegey hi nahi.

- Now aab reciever kya karega, jaha five dekhe uske baad 0 hoga toh us 0 ko stuff 0 assume karke drop kar dega.

- Aur jaha six one dikhe woh kya assume karega ki frame end hogayi.

- Agar bit pattern n- bit ka hai toh woh kitane bit ko check karta hai (n-1) bits ko.

- Aur jaha (n-1) bit match hogayi, bahale n^{th} bit kuch bhi kyo na ho, woh kya stuff karega 0 stuff karega.

ERROR CONTROL:→

Ques:→ Why to use Error detection techniques?

Ans→ Error occur in data due to attenuation, distortion, noise and interface interference.

TYPES OF ERROR:→

- Single bit Error
- Multibit Error
- Burst Error.

- Single bit Error:→
Only one bit get corrupted common in parallel transmission

VRC :→

- VRC stands for Vertical Reducdancy Check.
- VRC is like parity Sy Scheme.

Ques:- Yeh parity Scheme Kya hoti hai ??

Ans:- Agar aap odd parity maintain Karke chalte ho, toh kya hota hai -

- Including parity bit number of ones Should be odd.
- Agar hum even parity Scheme follow karte hai toh Kya hota hai, toh including even parity bit, number of 1's (ones) should be even.
- Suppose given collection of bit is our data - and hamne decide Kiya ki per 7 bit hum ek parity bit insert karegy data mein.
- Per 7 bit there will be parity bit.

1 0 1 1 0 0 1 [ ]

- If odd parity bit is maintain toh is parity bit Ki value kitani hogi -
- Odd parity humne kya kaha tha including parity bit number of one's Should be odd.
- Agar hum odd parity likar chal rahai hai, toh Kya hoga - yaha par numbers of ones kitane

hai 4 i.e even, toh yaha par parity bit ki value kitane hogi one (1).

1 0 1 1 0 0 1 [1]

└ odd parity.

- Agar odd parity bit maintain karnahai hai toh parity bit ki value kitane aanahai hai 1 (one).

- Agar hum even parity like chal rahai hai, toh value of parity bit will be zero.

1 0 1 1 0 0 1 [0]

└ even parity.

- Hamne parity bit add karke data send kar diya, toh reciever bhi kya check karega ko kaise pata chalega there is an error.

- Parity kitane aane chahiye odd, toh agar ~~number~~ number of ones odd hogey toh reciever assume karega ki hamara data mein koi error nahi hai but agar even aaye number of ones toh reciever assume karega ki data mein error hai.

- Suppose hamane send kiya tha A, par reciever ko kya mila B toh reciever assume karega ki error aayi hai kya nahi aayi hai

A = 10101011

B = 11101001

- Joh error check karne k liye sabse pahale reciever number of ones count karega, agar no. of ones even aaye toh error otherwise no error. Hum assume kar rahai hai ki reciever odd parity lekar chal raha hai.

- Joh B me number of one's kitane hai odd toh no error.

- But yaha par error hai.

A: 1 0 1 0 1 0 1 1
B: 1 1 1 0 1 0 0 1

→ yaha par 2 bit error hai.

* Joh parity bit can detect single bit error

* But yeh two bit ki error ko detect nahi kar payegi.

- Join parity bit can detect single bit error or odd length error.

P

1 0 1 1 0
0 1 0 0 1

Yeh error detect hojayegi.

because yaha par error wale data mein no. of ones kitane hai even ⇒ error (odd parity le kar