Détails du code Hamming :

Dans le code de Hamming, la relation 2 ^ m> = n + 1 est établie que :

n = nombre de bits dans un bloc

m = nombre de bits de contrôle par bloc (m = n-k)

k = nombre de bits d'information dans le bloc

Codage:

1- Dans un premier temps, on obtient le nombre de bits de contrôle en fonction de ladite longueur en fonction de la longueur du bloc et du nombre de bits d'information. Les bits du bloc sont numérotés à partir de la droite d'un nombre.

2- Si le nombre de bits (total de contrôle et d'information) est devenu individuel. Ajouter un zéro à droite. Il est à noter que ce bit ne participe pas à la numérotation.

3- On remplit les bits qui sont des puissances de deux (... et 1,2,4,4,8) avec des bits de contrôle et le reste des bits (... et 3,5,6,6,7) avec des bits d'information. Chaque bit de contrôle est rempli de manière à ce que la parité du bit de contrôle et d'un ensemble de bits soit paire. Pour vérifier le solde, nous devons procéder comme suit :

A - On vérifie le solde des bits dont le reste est divisé par 2.

B- On vérifie le solde des bits dont le reste est divisé par 4, deux ou trois.

C- On vérifie le solde des bits dont le reste est divisé par 8, quatre, cinq, six ou sept.

D- Nous vérifions le solde des bits dont le reste est divisé par 16, huit, neuf, dix, onze, douze, treize, quatorze ou quinze. Etc.

Décodage:

1- Calculer le solde des bits

A- On calcule le solde des bits dont le reste est divisé par 2.

B- On calcule le solde des bits dont le reste est divisé par 4, deux ou trois.

C- On calcule le solde des bits dont le reste est divisé par 8, quatre, cinq, six ou sept.

D- On calcule le solde des bits dont le reste est divisé par 16, huit, neuf, dix, onze, douze, treize, quatorze ou quinze. Etc.

2- Ensuite, nous écrivons les équilibres obtenus sous forme de nombre binaire, respectivement. La valeur du solde obtenu dans la première étape est de 20, la deuxième étape est de 21, la troisième étape est de 22 et ainsi de suite.

3- Si le nombre obtenu à l'étape précédente est zéro, cela signifie qu'il n'y a pas d'erreur, mais s'il n'est pas zéro, le nombre résultant indique le numéro de bit qui a une erreur en binaire. Le bit souhaité doit être inversé.

4- Dans la dernière étape, nous supprimons les bits de contrôle, et si dans la deuxième étape de codage est ajouté un bit, nous le supprimons également. Enfin, les données originales sont obtenues.

Détails du code( fenêtre):

L'un des inconvénients du protocole d'arrêt et d'attente était que, comme l'expéditeur attendait de recevoir le message de confirmation de trame le plus récent avant d'envoyer la trame suivante, il était inactif tout le temps. Mais dans le protocole de fenêtre coulissante, le temps d'attente est également utilisé pour envoyer des données. Dans le protocole de fenêtre coulissante, "fenêtre" fait référence à la mémoire tampon utilisée pour stocker les images. Dans ce protocole, deux fenêtres ou deux mémoires tampons sont définies : l'une est la fenêtre émettrice et l'autre est la fenêtre réceptrice. Le nombre de trames que l'expéditeur doit envoyer est stocké dans la fenêtre de l'expéditeur. Si la taille de la fenêtre de l'expéditeur est supposée être n, les trames sont numérotées de 0 à n-1, et l'expéditeur envoie n trames à chaque fois au lieu d'une. La fenêtre du récepteur contient également le nombre de trames à recevoir.

Contrairement au protocole stop-and-wait, dans lequel la méthode de communication était semi-duplex, dans le protocole de fenêtre coulissante, la communication expéditeur-récepteur est complètement duplex, ce qui signifie qu'ils peuvent s'envoyer des données en même temps. Il s'agit d'une caractéristique importante du protocole de fenêtre coulissante.

Dans le protocole à fenêtre coulissante, l'émetteur stocke un certain nombre de trames en fonction de la taille de sa mémoire tampon ou fenêtre dite. Les trames de la mémoire tampon sont envoyées consécutivement. C'est-à-dire que l'expéditeur envoie la première trame, mais n'attend pas le message de confirmation du récepteur, mais envoie également les trames suivantes. Lorsque le récepteur reçoit le premier message de confirmation de trame du récepteur, l'expéditeur, qui est sûr qu'il a atteint sa destination, efface cette trame de sa mémoire tampon (fenêtre). Par conséquent, un espace est ouvert pour enregistrer une nouvelle trame et l'émetteur enregistre une nouvelle trame dans sa mémoire tampon. Dans ce cas, la fenêtre dite de l'émetteur (mémoire tampon de l'émetteur) glisse vers l'avant pour ajuster le nouveau cadre dans son cadre. De même, lorsque le deuxième message de confirmation de trame atteint l'expéditeur, l'expéditeur efface également la deuxième trame de sa mémoire, et la fenêtre glisse d'un pas de plus pour enregistrer la nouvelle trame. L'expéditeur envoie également d'autres trames au destinataire. Ce processus se poursuit jusqu'à ce que toutes les trames soient envoyées à la destination.

Ainsi, dans le protocole de fenêtre coulissante, l'expéditeur et le destinataire ne passent aucun temps d'attente et échangent des données tout le temps.

Dans le code défini, les informations sont transférées à l'aide de l'algorithme de fenêtre et l'ordinateur A dans la couche supérieure transfère les informations à l'ordinateur B dans la couche inférieure et sont stockées et récupérées dans la mémoire tampon à l'aide de la pile d'informations et dans un chemin séparé ce transfert Fait .