Lab 13:

1. One method of reducing bandwidth use is to compress the data being transmitted. Let A = {a/20, b/15, c/5, d/15, e/45} be the alphabet and its frequency distribution. Compute the optimal coding for each character. What is the average number of bits/symbol of the codes?

Use Huffman Coding. Huffman Coding assigns shorter codes to more frequently occurring characters, reducing the average number of bits per symbol. Combines two nodes with the lowest frequency together and repeats the process until one node is left.

E=45, A = 20, B = 15 , D = 15, C = 5

E=45, A = 20, B = 15 , P1 = 20 (D = 15, C = 5)

E=45, A = 20, P2 = 35 (B = 15 + P1( D = 15, C = 5)

E=45, P3 = 55 (A = 20 + P2)

P4 = 100 (E=45 + P3)

I chose to group D before B but they are interchangeable as they have the same frequency. So my graph and codes are based on D being grouped before B.

Therefore codes are :

A paper with writing on it

Description automatically generated with low confidenceE = 0

A = 10

B = 110

D = 1111

C = 1110

The average number of bits is the length of each code by its own frequency, summed, divided by total frequency.

So E Length is 1 \* frequency 45 = 45

A Length is 2 \* frequency 20 = 40

B Length is 3 \* frequency 15 = 45

D length is 4 \* frequency 15 = 60

C length is 4 \* frequency 5 = 20

Sum is 45 + 30 + 45 + 60 + 20 = 200

200 / total frequency 100 = 2 bits per symbol.

Therefore, the average number of bits per symbol for the given coding scheme is 2 bits/symbol.

1. Please describe the information exchanges and the actions taken for both server and client according to the diagram for delta compression.

Chart

Description automatically generated with medium confidence

The client acknowledges which state it received.

The server keeps track of the last state the client received. Delta compresses the current state based on this.

1. Server Sends base state for the object.
2. Server sends changes since the base state with the ID of 1.
3. Client Acknowledges receiving state changes with ID 1.
4. Server Sends changes since base State with ID 2.
5. Client decompresses the received state using the base state to get to State 2.
6. Server receives acknowledgement of State 1
7. Client Acknowledges receiving state changes with an ID of 2
8. Server sends changes since state 1 with the ID of 3
9. Server receives acknowledgement of state 2 being received by the client.
10. Client acknowledges it has received state changes of ID 2 but acknowledgement 3 is lost.
11. Server sends changes since state 3 with the ID of 4 but Packet is lost.
12. Server sends changes since state 3 with the ID of 5 but Packet is lost.
13. Client acknowledges receiving state changes with the ID of 5.
14. Server receives acknowledgement of state 5 being received by the client and sends changes since state 5 with the ID of 6.
15. One method of reducing bandwidth use is to compress the data being transmitted. Use the LZW algorithm to compress the string: BABAABAAA. Note that Uppercase A has ASCII value 65 in decimal. Draw diagrams to aid your explanation if appropriate.

Basic Dictionary A = 65 , B = 66

String = BABAABAAA

|  |  |  |  |
| --- | --- | --- | --- |
| Current | Next | Output | Add To Dictionary |
| B | A | B : 66 | BA : 256 |
| A | B | A : 65 | AB : 257 |
| B | A | In dictionary check next |  |
| BA | A | BA : 256 | BAA : 258 |
| A | B | AB in dictionary so check next |  |
| AB | A | AB : 257 | ABA : 259 |
| A | A | A : 65 | AA : 260 |
| A | A | AA is in dictionary, output AA : 260 as end of string |  |

Output is B | A | BA | AB | A | AA

So compression is : 66 | 65 | 256 | 257| 65 | 260

Original uncompressed is : 66 | 65 | 66 | 65 | 65 | 66 | 65 | 65 | 65

Compressed from 9 to 6 groups