CHƯƠNG 4
XỬ LÝ SỐ LIỆU TRUYỀN

Trang bị cho sinh viên: Các phương pháp mã hóa số liệu mức vật lý, phương pháp phát hiện sai và sửa sai, cách thức nén số liệu và mật mã hóa số liệu trong quá trình truyền số liệu.

4.1 MÃ HÓA SỐ LIỆU MỨC VẬT LÝ

Số liệu cung cấp từ máy tính hoặc các thiết bị đầu cuối số liệu thường ở dạng nhị phân đơn cực (unipolar) với các bit 0 và 1 được biểu diễn cùng mức điện áp âm hoặc dương. Tốc độ truyền dẫn của chúng được tính bằng số bit truyền trong một giây. Các số liệu này khi truyền đi sẽ được biến đổi sang dạng tín hiệu với các kỹ thuật mã hóa khác nhau. Các tín hiệu này được đặc trưng bằng sự thay đổi mức điện hoặc tốc độ truyền của chúng vì thế chúng được xác định bằng tốc độ của sự thay đổi này, còn được gọi là tốc độ điều chế và được tính bằng Baud.

4.1.1 Unipolar:

Là dạng đơn giản nhất và nguyên thủy nhất. Cho dù đây là dạng đã lạc hậu,nhưng tính chất đơn giản của nó luôn là tiền đề cho các ý niệm về phát triển các hệ thốngphức tạp hơn, đồng thời phương pháp này cũng giúp ta nhìn thấy nhiều vấn đề trong truyền số liệu phải giải quyết. Hệ thống truyền số liệu hoạt động trên cơ sở gởi các tín hiệu điện áp trong môi trường kết nối, thường là dây dẫn hay cáp. Trong nhiều dạng mã hóa, một mức điện áp biểu thị cho giá trị nhị phân 0 và một mức khác cho giá trị 1. Cực tính của xung tùy thuộc vào giá trị điện áp là dương hay âm. Mã hóa đơn cực (unipolar) là phương pháp chỉ dùng một dạng cực tính, thường thì cực tính này biểu diễn một giá trị nhị phân, thường là 1, còn giá trị điện áp không thường dùng cho giá trị bit 0.

4.1.2 Mã hóa NRZ (Non Return to Zero Level):

NRZ là một kỹ thuật mã hóa kênh giúp chuyển các giá trị logic bit thành các xung điện có thể truyền qua đường truyền hữu tuyến. Mã hóa NRZ là trong thời gian một bit tín hiệu không trở về mức 0, dùng trong trường hợp lưu dữ bằng vật liệu bằng từ tinh. Ít dùng trong truyền số liệu

4.1.3 Nonreturn - to - Zero - Level (NRZ – L)

0 = mức cao

1 = mức thấp

Đây là dạng mã đơn giản nhất, hai trị điên thế cùng dấu (đơn cực) biểu diễn hai trạng thái logic. Loại mã này thường được dùng trong việc ghi dữ liệu lên băng từ, đĩa từ,. . .

4.1.3 Nonreturn - to - Zero Inverted (NRZ – I)

0 = chuyển mức điện thế ở đầu bit

1 = không chuyển mức điện thế ở đầu bit

NRZI là một thí dụ của **mã vi phân**: sự mã hóa tùy vào sự thay đổi trạng thái của các bit liên tiếp chứ không tùy thuộc vào bản thân bit đó**.** Loại mã này có lợi điểm là khi giải mã máy thu chỉ cần dò sự thay đổi trạng thái của tín hiệu thì có thể phục hồi dữ liệu thay vì phải so sánh tín hiệu với một trị ngưỡng để xác định trạng thái logic của tín hiệu đó. Kết quả là các loại mã vi phân cho độ tin cậy cao hơn.

4.1.4 Mã RZ

Các mức nhị phân của dữ liệu được biểu diễn bằng các mức điện áp tương ứng trong nữa chu kỳ bít, sau đó trở về 0 trong nữa chu kỳ kế tiếp

4.1.5 Mã Manchester

0 = Chuyển từ cao xuống thấp ở giữa bit

1 = Chuyển từ thấp lên cao ở giữa bit

4.1.6 Mã Manchester vi sai (Differential Manchester)

Luôn có chuyển mức ở giữa bit

0 = chuyển mức ở đầu bit

1 = không chuyển mức ở đầu bit

4.1.7 Mã B8ZS

Là mã AMI có thêm tính chất: chuỗi 8 bit 0 liên tục được thay bởi một chuỗi 8 bit có cả bit 0 và 1 với 2 mã vi phạm luật đảo bit 1

Nếu trước chuỗi 8 bit 0 là xung dương, các bit 0 này được thay thế bởi **000 + - 0 - +**

Nếu trước chuỗi 8 bit 0 là xung âm, các bit 0 này được thay thế bởi **000 - + 0 + -**

Nhận xét bảng mã thay thế ta thấy có sự vi phạm luật đảo bit ở 2 vị trí thư 4 và thứ 7 của chuỗi 8 bit.

4.1.8 Mã HDB3

Là mã AMI có thêm tính chất: chuỗi 4 bit 0 liên tục được thay bởi một chuỗi 4 bit có cả bit 0 và 1 với 1 mã vi phạm luật đảo bit 1

Sự thay thế chuỗi 4 bít của mã HDB3 còn theo qui tắc sau: Cực tính của xung trước đó	Số bít 1 từ lần thay thế cuối cùng	
	Lẻ	chẵn
-	**000-**	**+00+**
+	**000+**	**-00-**

4.2 PHÁT HIỆN LỖI VÀ SỬA SAI

4.2.1 Tổng quan

Khi dữ liệu được truyền giữa 2 DTE, các tín hiệu điện đại diện luồng bit truyền rất dễ bị thay đổi sai số đó do nhiều nguyên nhân: đường dây truyền, lưu lượng truyền, loại mã đùng, loại điều chế, loại thiết bị phát, thiết bị thu. Đặc biệt là do sự thâm nhập điện từ cảm ứng lên các đường dây từ các thiết bị điện gần đó, Nếu các đường dây tồn tại trong một môi trường xuyên nhiễu thí dụ như mạng điện thoại công cộng. Điều này có nghĩa là các tín hiệu đại diện cho bit 1 bị đầu thu dịch ra như bit nhị phân 0 và ngược lại. Để xác suất thông tin thu được bởi DTE đích giống thông tin đã truyền đạt được giá trị cao, cần phải có một vài biện pháp để nơi thu có khả năng nhận biết thông tin thu được có chứa lỗi hay không, nếu có lỗi sẽ có một cơ cấu thích hợp để thu về bản copy chính xác của thông tin.

Để chống sai khi truyền số liệu thường có 2 cách :

Dùng bộ giải mã có khả năng tự sửa sai

Truyền lại một bộ phận của dữ liệu để thực hiện việc sửa sai, cách này gọi là ARQ(Automatic Repeat Request).

4.2.2 Phương pháp kiểm tra chẵn lẻ theo ký tự (Parity Bit)

Phương pháp thông dụng nhất được dùng để phát hiện lỗi của bit trong truyền không đồng bộ và truyền đồng bộ hướng ký tự là phương pháp parity bit. Với cách này máy phát sẽ thêm vào mỗi ký tự truyền một bit kiểm tra parity đã được tính toán trước khi truyền. Khi nhân được thông tin truyền, máy thu sẽ thực hiện các thao tác tính toán trên các ký tự thu được , và so sánh với bit parity thu được. Nếu chúng bằng nhau thì được giả sử là không có lỗi, ở đây ta dùng từ giả sử, bởi vì cách này có thể không phát hiện được lỗi trong khi lỗi vẫn tồn tại trong dữ liệu. Nhưng nếu chúng khác nhau thì chắc chắn một lỗi xảy ra .

Để tính toán parity bit cho một ký tự, số các bit trong mã ký tự được cộng module 2 với nhau và parity bit được chọn sao cho tổng số các bit 1 bao gồm cả parity bit là chẵn (even parity) hoặc là lẻ (odd parity)

Trong bộ mã ASCII mỗi ký tự có 7 bit và một bit kiểm tra

Với kiểm tra chẵn giá trị của bit kiểm tra là 1 nếu số lượng các bit có giá trị 1 trong 7 bit là lẻ và có giá trị 0 trong trường hợp ngược lại.

Với kiểm tra lẻ thì ngược lại. Thông thường người ta sử dụng kiểm tra chẵn và bit kiểm tra gọi là P. Giá trị kiểm tra đó cho phép ở đầu thu phát hiện những sai sót đơn giản

Ví dụ:	Kí tự	Mã ASCII	Từ mã phát đi	Bit kiểm tra P
	A	1000001	10000010	0
	E	1010001	10100011	1

Phương pháp parity bit chỉ phát hiện các lỗi đơn bit (số lượng bit lỗi là số lẻ)và không thể phát hiện các lỗi 2 bit (hay số bit lỗi là một số chẵn)

4.2.3 Phương pháp kiểm tra tổng khối

Khi truyền đi một khối thông tin, mỗi ký tự được truyền đi sẽ được kiểm tra tính chẵn lẻ theo chiều ngang, đồng thời cả khối thông tin này cũng được kiểm tra tính chẵn lẻ theo chiều dọc. Cứ sau một số byte nhất định thì một byte kiểm tra chẵn lẻ cũng được gửi đi, byte chẵn lẻ này được tạo ra bằng cách kiểm ta tính chẵn lẻ của khối ký tự theo cột. Dựa vào các bit kiểm tra ngang và dọc ta sẽ xác định được toạ độ của bit sai và sửa được bit sai này

Chúng ta có thể thấy rằng mặc dù các lỗi 2 bit trong một ký tự sẽ thoát khỏi kiểm tra parity theo hàng, nhưng chúng sẽ bị phát hiện bởi kiểm tra parity theo cột tương ứng. Dĩ nhiên điều này là đúng chỉ khi không có lỗi 2 bit xảy ra trong cùng một cột tại cùng thời điểm. Rõ ràng xác suất xảy ra trường hợp này nhỏ hơn nhiều so với xác suất xảy ra lỗi 2 bit trong một ký tự. Việc dùng kiểm tra theo ma trận cải thiện đáng kể các đặc trưng phát hiện lỗi của kiểm tra chẵn lẻ

Vị trí bit truyền	Dãy bit truyền đi							Kiểm p theo hang
1	1	0	0	0	0	0	1	0
2	1	0	0	1	0	1	0	1
3	1	0	1	0	0	1	1	0
Kiểm tra p theo cột	1	0	1	1	0	0	0	0

Ví dụ: sau khi truyền nhận được dãy truyền như dưới đây dựa vào sự thay đổi của các bit p dọc và ngang sẽ xác định được vị trí lỗi tương ứng (hình dưới)

Vị trí bit truyền	Dãy bit truyền đi							Kiểm p theo hang
1	1	0	0	**1**	0	0	1	1
2	1	0	0	1	**1**	1	0	0
3	1	0	1	0	0	1	1	0
Kiểm tra p theo cột	1	0	1	0	1	0	0	0

Tuy nhiên phương pháp này cũng không hoàn toàn hiệu quả. Giả sử bit thứ nhất và bit thứ 3 của ký tự thứ nhất bị sai kiểm tra hàng sẽ không bị sai, nhưng kiểm tra chẵn lẻ của cột sẽ phát hiện bị sai, nhưng không xác định được vị trí sai ???

Vị trí bit truyền	Dãy bit truyền đi							Kiểm p theo hang
1	1	0	0	0	1	0	0	0
2	1	0	0	1	0	1	0	1
3	1	0	1	0	0	1	1	0
Kiểm tra p theo cột	1	0	1	1	1	0	0	1

Bây giờ ta lai giả thiết rằng bit thứ nhất và bit thứ 3 của ký tự thứ 3 cũng bị sai đồng thời với bit thứ nhất và bit thứ 3 của ký tự thứ nhất, lúc đó kiểm tra chẵn lẻ của cột và hàng đều không thấy sai nên đã không phát hiện được sai???

Vị trí bit truyền	Dãy bit truyền đi							Kiểm p theo hang
1	1	0	0	0	1	0	0	0
2	1	0	0	1	0	1	0	1
3	1	0	1	0	1	1	0	0
Kiểm tra p theo cột	1	0	1	1	0	0	0	0

4.2.4 Phương pháp mã dư thừa - mã vòng (CRC)

Một từ mã được viết dưới dạng một đa thức

$$C(x) = C_{n-1} X^{n-1} + C_{n-2} X^{n-2} + ...+ C_1 X + C_0$$

Phương pháp kiểm tra tín hiệu bằng mã vòng được thực hiện như sau: Tín hiệu cần phát đi gồm k bit sẽ được bên phát thêm vào n bit nữa để kiểm tra được gọi là Frame Check Sequence (FCS). Như vậy tín hiệu phát đi bao gồm k+n bit. Bên thu khi nhận được tín hiệu nay sẽ đem chia cho một đa thức được gọi là đa thức sinh đã biết trước (bên phát và bên thu đều cùng chọn đa thức này). Nếu kết quả chia không dư coi như tín hiệu nhận được là đúng. Vấn đề được đặt ra là n bit thêm vào sẽ được xác định như thế nào khi đã biết khung tin cần truyền đi, và biết đa thức sinh đã được chọn, n bit thêm vào đó được gọi là CRC (Cyclic Redundancy Check). Phương pháp tạo ra CRC bao gồm việc dịch thông báo sang trái c bit (c chính là bậc của đa thức đã chọn trước) sau đó thực hiện phép chia cho đa thức được chọn này. Kết quả dư lại của phép chia chính là CRC. Bên thu sau khi nhân được thông báo cũng đem chia cho hàm biết trước như bên phát. Nếu kết quả bằng 0. phép truyền không sai số.

Tính xâu phải truyền đi FCS theo 4 bước:

Bước 1: Chuyển thông báo nhị phân thành đa thức M(x), sau đó chọn đa thức sinh cho trước $G(x) = X^c + 1$ có bậc cao nhất là c (c là độ dài của xâu kiểm tra CRC)

Bước 2: Nhân M(x) với X^c

Bước 3: Thực hiện $M(x)*X^c/G(x)$ được kết quả là thương Q(x) và số dư R(x). R(x) chính là xâu kiểm tra CRC

Bước 4: Thành lập xâu truyền đi là: $M(x)*X^c + R(x)$

Ví dụ: xâu cần truyền là: 110101 hãy sử dụng phương pháp mã CRC để kiểm tra lỗi khi truyền

Giải:

Bước 1: Dãy nhị phân 110101 có thể được dạng đa thức là $M(x) = X^5 + X^4 + X^2 + 1$

Bước 2: Chọn đa thức sinh có số mũ là 3: $G(x) = X^3 + 1$

Bước 3: Tính $M(x)*X^3/G(x) = Q(x) + R(x) = (X^8 + X^7 + X^5 + X^3)/X^3 + 1$

$X^8 + X^7 + X^5 + X^3$ | $X^3 + 1$

$X^8 \quad\quad X^5$ | $X^5 + X^4 + X + 1$

$X^7 + \quad X^3$

$X^7 + X^4$

$X^4 + X$

$X^3 + X$

$X^3 + 1$

$X + 1$

Kết quả cho: $Q(x) = X^5 + X^4 + X + 1$

$R(x) = X + 1$

⇨ Xâu kiểm tra CRC là: $X + 1 = 011$ (dài 3 ký tự)

Bước 4: Xâu phải truyền có mã CRC là: $M(x)*X^c + CRC = 110101$**011**

Thu và kiểm tra CRC

Để kiểm tra sai số khi truyền bên thu đem khối thông tin thu được chia cho G(x) theo modul 2, nếu số dư nhận được là 0 thì truyền đúng, ngược lại dữ liệu đã bị truyền sai

Ví dụ: Thông tin truyền đi là: 110101011

Thông tin nhận được là: 110101011

Có nghĩa thông tin truyền đúng tức là R(x) = 0

Kiểm tra CRC như sau:

Chuyển thông tin nhận được thành đa thức: x8 +x7 +x5 + x3 + x1 + 1

Đa thức sinh của hai bên phát và thu là: G(x) = x3 + 1

Chia đa thức nhận được cho G(x), kết quả R(x) = 0, vậy không có truyền sai

$X^8 + X^7 + X^5 + X^3 + X + 1$ | $X^3 + 1$

$X^8 \quad\quad X^5$ | $X^5 + X^4 + X + 1$

$X^7 + \quad X^3$

$X^7 + X^4$

$X^4 + X^3 + X$

$X^4 \quad + X$

$X^3 \quad + 1$

$X^3 \quad + 1$

$0 = R(x)$

4.2.5 Phát hiện và sửa sai theo mã Hamming

Mã Hamming là một mã sửa lỗi tuyến tính được đặt tên theo tên của người phát minh ra nó, Richard Hamming. Mã Hamming có thể phát hiện và sửa được sai cho một số bit nhất định

Nguyên tắc:

Một từ mã Hamming gồm m bit nhị phân dữ liệu và p bit kiểm tra chẵn lẻ ($2^p - 1 \geq m + p$), p bit kiểm tra được bố trí tại các vị trí thích hợp để phát hiện chính xác vị trí lỗi là các số mũ của 2 ví dụ: 2^0, 2^1, 2^2,…

Tất cả các vị trí bit khác được dùng cho dữ liệu sẽ được mã hóa. (các vị trí 3, 5, 6, 7, …

Mỗi bit chẵn lẻ tính giá trị chẵn lẻ cho một số bit trong m bit. Vị trí của bit chẵn lẻ quyết định chuỗi các bit mà nó luân phiên kiểm tra và bỏ qua

Vị trí 1 (n=1): bỏ qua 0 bit(n-1), kiểm 1 bit(n), bỏ qua 1 bit(n), kiểm 1 bit(n), bỏ qua 1 bit(n), v.v. → p1 (2^0)

Vị trí 2(n=2): bỏ qua 1 bit(n-1), kiểm 2 bit(n), bỏ qua 2 bit(n), kiểm 2 bit(n), bỏ qua 2 bit(n), v.v. → p2 (2^1)

Vị trí 4(n=4): bỏ qua 3 bit(n-1), kiểm 4 bit(n), bỏ qua 4 bit(n), kiểm 4 bit(n), bỏ qua 4 bit(n), v.v. → p3 (2^2)

Vị trí 8(n=8): bỏ qua 7 bit(n-1), kiểm 8 bit(n), bỏ qua 8 bit(n), kiểm 8 bit(n), bỏ qua 8 bit(n), v.v. → p4 (2^3)

Vị trí 16(n=16): bỏ qua 15 bit(n-1), kiểm 16 bit(n), bỏ qua 16 bit(n), kiểm 16 bit(n), bỏ qua 16 bit(n), v.v. → (2^4)

Vị trí 32(n=32): bỏ qua 31 bit(n-1), kiểm 32 bit(n), bỏ qua 32 bit(n), kiểm 32 bit(n), bỏ qua 32 bit(n), v.v. → (2^5)

và tiếp tục như trên.

Xác định vị trí lỗi bằng cách lấy mudul 2 (xor) của các bit chẵn lẻ của dữ liệu sau và trước khi truyền nếu = 0 thì không có lỗi, ngược lại thì vị trí lỗi là kết của phép môdul 2 chuyển sang hệ thập phân.

b) Ví dụ 1: cho m bit dữ liệu cần truyền là: 0110101 hãy chèn các bit cần thiết vào theo phương pháp mã sửa sai Hamming. (cho biết dãy truyền từ phải sang trái →) Giải:

- m = 7, p = 4 ($2^p - 1 \geq 7 + 4$) các bit p1, p2, p3, p4 được bố trí vào các vị trí 1,2,4,8
- Các vị trí 3,5,6,7,9,10,11 là các bit dữ liệu
- Xác định p1, p2, p3, p4 theo thuật toán kết quả là: 1000

  p1= 01011 = 1 (xor)

  p2= 01001 = 0

  p3= 110 = 0

p4= 101 = 0

- Rải các bit vừa tính được vào trong xâu gốc ta được xâu cần truyền là: **<u>10</u>0<u>0</u>1110<u>0</u>10**

Thứ tự bit	1	2	3	4	5	6	7	8	9	10	11
Vị trí bit chẵn lẻ và các bit dữ liệu	p_1	p_2	d_1	p_3	d_2	d_3	d_4	p_4	d_5	d_6	d_7
Nhóm dữ liệu (không có bit chẵn lẻ):			0		1	1	0		1	0	1
p_1	1		0		1		0		1		1
p_2		0	0			1	0			0	1
p_3				0	1	1	0				
p_4								0	1	0	1
Nhóm dữ liệu (với bit chẵn lẻ):	1	0	0	0	1	1	0	0	1	0	1

Xác định vị trí bit lỗi như sau:

- Giả sử sau khi truyền nhận được dãy nhị phân là: 10001100100.

Thứ tự bit	1	2	3	4	5	6	7	8	9	10	11
Vị trí bit chẵn lẻ và các bit dữ liệu	p_1	p_2	d_1	p_3	d_2	d_3	d_4	p_4	d_5	d_6	d_7
Giả sử dữ liệu sau khi nhận là:	1	0	0	0	1	1	0	0	1	0	0
p_1	0		0		1		0		1		0
p_2		1	0			1	0			0	0
p_3				0	1	1	0				
p_4								1	1	0	0
bit bị đảo lộn có nền thẫm											

- Bằng cách xác định các bit chẵn lẻ p1, p2, p3, p4 như trên ta được: p1=0, p2=1, p3=0, p4=1

- Thực hiện phép XOR các bit chẵn lẻ trước và sau khi truyền (chú ý phải viết ngược lại):

	p4	p3	p2	p1	
	1	0	1	0	
XOR	0	0	0	1	
	1	0	1	1	$= 11_{(10)} \rightarrow$ vị trí lỗi là 11

c) ví dụ 2: Cho chuỗi bit gốc m = 1101100111, hãy chèn các bit cần thiết vào theo phương pháp mã sửa sai Hamming. (cho biết dãy truyền từ trái sang phải ←)

Giải:

- m = 10, p = 4 (2^4 -1 > 10 + 4) các bit p1, p2, p3, p4 được bố trí vào các vị trí 1,2,4,8
- Các vị trí 3,5,6,7,9,10,11,12,13,14 là các bit dữ liệu
- Xác định p1, p2, p3, p4 theo thuật toán kết quả là:

p1= 110011 = 0

p2= 110111 = 1

p3= 110110 = 0

p4= 011011 = 0

Rải các bit vừa tính được vào trong xâu gốc ta được xâu cần truyền là: 1 1 0 1 1 0 **<u>0</u>** 0 1 1 **<u>0</u>** 1 **<u>1 0</u>**

Giả sử sau khi truyền xâu nhận được là: 1 1 0 1 1 0 **0 <u>1</u>** 1 1 **0** 1 **1 0** hãy xác định vị trí lỗi

Tìm p1, p2, p3, p4 của xâu nhận:

P1 = 111011 = 1

P2 = 111111 = 0

P3 = 111011 = 1

P4 = 011011 = 0

- Vị trí lỗi là

	p4	p3	p2	p1	
	0	1	0	1	
Xor	0	~~0~~	~~1~~	~~0~~	
	0	1	1	1	→ Vị trí lỗi là 7

4.3 NÉN SỐ LIỆU

4.3.1 Khái quát

Nội dung thông tin truyền đi bao gồm dữ liệu gốc dưới dạng chuỗi ký tự có chiều dài cố định. Trong đó dữ liệu được nén trước khi truyền đi, nén dữ liệu là một việc làm thiết yếu trong các

dịch vụ truyền dẫn công cộng, ví dụ truyền qua mạng PSTN, vì trong các mạng các mạng như vậy việc tính cước dựa vào thời gian và cự ly truyền. .

Trong thực tế chúng ta có thể dùng một loạt các giải thuật nén khác nhau, mỗi giải thuật sẽ phù hợp với một loại dữ liệu. Vài modem thông minh sẽ cung cấp đặc trưng nén thích nghi tự động thực hiện các giải thuật nén phù hợp với loại dữ liệu đang được truyền

4.3.2 Nén nhờ đơn giản mã cho các chữ số (Packed Decimal)

Khi dữ liệu chỉ bao gồm các ký tự số học đang được truyền, chúng ta có thể tiết kiệm đáng kể bằng cách giảm số bit trên mỗi ký tự từ 7 xuống 4 thông qua mã BCD, thay cho mã ASCII vì các số từ 0 – 9 đều có 3 bit cao là 011 (thực tế 3 bit này dùng để phân biệt giữa ký tự là số và không phải là số). Do vậy nếu dữ liệu chỉ là các ký tự số thì 3 bit này là dư thừa

4.3.3 Nén theo mã hóa quan hệ

Một phương pháp khác được sử dụng khi truyền dữ liệu số học kế tiếp chỉ khác nhau phần nhỏ về giá trị là chỉ gửi lượng khác nhau giữa các giá trị này cùng với một giá trị tham khảo. Điều này được gọi là mã hóa quan hệ và nó có thể đem lại hiệu quả đặc biệt trong các ứng dụng ghi nhân dữ liệu.

Ví dụ: nếu giám sát từ xa mục nước của dòng sông thường đọc mức nước theo các khoảng thời gian định trước . để tối thiểu thời gian cần truyền thay vì truyền giá trị chỉ mưc nước tuyệt đối, chúng ta chỉ cần truyền đi các giá trị khác nhau.

4.3.4 Nén bằng cách bỏ bớt các ký tự giống nhau (mã hoá độ dài loạt Run-Length Encoding)

Thông thường dữ liệu được truyền thường xuất hiện chuỗi lặp lại các ký tự giống nhau. Thiết bị điều khiển tại máy phát sẽ quét nội dung của dữ liệu trước khi truyền nếu gặp một chuỗi ký tự liên tiếp giống nhau thì chúng sẽ được thay thế bởi một cặp số và ký tự.

Ví dụ, xét chuỗi sau: AAAABBBAABBBBBCCCCCCCCDABCBAAABBBBCCCD

Chuỗi này gồm bốn chữ A theo sau bởi ba chữ B rồi lại theo sau bởi hai chữ A, rồi lại theo sau bởi năm chữ B...Do đó có thể sử dụng phương pháp mã hoá độ dài loạt để mã hóa. Vì vậy chuỗi kí tự trên được mã hoá lại như sau: 4A3BAA5B8CDABCB3A4B3CD. "4A" có nghĩa là "bốn chữ A"... Chú ý là không nên mã hoá các loạt chạy có độ dài 1 hoặc 2 vì cần đến hai kí tự để mã hoá. Đối với các tập tin nhị phân ý tưởng ở đây là lưu lại các độ dài loạt bằng cách tận dụng sự kiện các loạt chạy có thay đổi giữa 0 và 1. Ví dụ dãy nhị phân 1010 được biểu diễn bởi cặp số đếm và ký tự thì là như sau **1**1**1**0**1**1**1**0 nhưng để tránh phải lưu chính các số 0 và 1 đó, một phiên bản được tinh chế của phương pháp này được dùng đã thu được sự tiết kiệm đáng kể, lúc này dãy nhị phân trên sẽ được biểu diễn là 1010 (tiết kiệm được 4 ký tự dùng làm số đêm). Có nghĩa là ta tiết kiệm các bit trên một loạt chạy chỉ khi độ dài của đường chạy là lớn hơn số bit cần để biểu diễn chính nó trong dạng nhị phân. Việc mã hoá độ dài loạt cần đến các biểu diễn riêng biệt cho tập tin và cho bản đã được mã hoá của nó, vì vậy nó không thể dùng cho mọi tập tin, ví dụ, phương pháp nén tập tin kí tự sẽ không dùng được đối với các chuỗi kí tự có chứa số. Nếu dùng những kí tự khác để mã hoá các số đếm, thì nó sẽ không làm việc với các chuỗi chứa các