Proceedings of International Conference
on Networking and Multimedia

A Predictive Mechanism for ABR Traffic Management in ATM Networks

Tai-Kuo Woo, National Defense Management College

Abstract

We propose a predictive mechanism for managing the ABR traffic in ATM networks. For ABR traffic management, the establishment of a feedback loop between the source/destination end systems, and the network lets the data sources know the state of the network, and utilize the available network bandwidth more efficiently. However, because of the propagation delay of sending back the resource management cells to the data sources, the network is either underutilized (when the network load is light), or congested (when the network load is heavy) by the excessive cells that enter the network during the propagation period. The proposed predictive mechanism based on Trellis diagram predicts the contents of the resource management cells and makes rate adjustment beforehand. We also show how the predictive mechanism can be used as an "add-on" to the current congestion control algorithms such as EFCI, PRCA, and EPRCA. A performance evaluation demonstrates that the predictive approach reduces excessive cells and increases network utilization.

1. Introduction

In ATM Networks, a connection between the data source and the network requires a traffic contract, which specifies the bandwidth requirements (e.g., peak rate, average rate), and quality of service requirements (e.g., the tolerance on cell loss probability and delay). In terms of bandwidth allocation, the constant bit rate (CBR) gets the bandwidth normally at the peak cell rate. The variable bit rate (VBR) gets the bandwidth at the sustainable cell rate (SCR). Both CBR and VBR are guaranteed services. The third type, named available bit rate (ABR), is best effort service. In ABR, each data source has a vague bandwidth requirement, and can only use whatever bandwidth left, after CBR/VBR bandwidth requirements are met[3]. An ABR connection is assigned a minimum cell rate (MCR), and may use additional bandwidth if available. Normally, the destination end system or the network traffic control located in the switch sends a resource management cell periodically to notify the data source to increase or reduce its data rate. This strategy works well for conventional low speed packet switching networks. For a high speed network like ATM, the resource management cell used to inform the data resource requires a long propagation delay . Consider a network that spans an area of 3000 km, the switch experiences a long queue length at the switch port, and notifies the data source to reduce data rate using a resource management cell. The propagation delay of the resource management cell is approximately 0.015 sec. During the propagation period for a high speed network , say OC-3 (155.52 Mbps), the amount of cells that enter the network is around 55000. The large amount of data makes the network congestion state even worse, and thus, data cells are lost and retransmission of cells makes the delay even longer [2,4].

On the other hand, if the network load is light, the destination end system sends a resource management cell to notify the data source to increase its rate. The long propagation delay would mean that the network is underutilized during the period of propagating the resource management cells back to the source end system.

In this paper, we propose a predictive approach to adjust the available bandwidth of ABR service. It works by allowing a data source to predict the network condition based on previously received resource management cells. The mechanism is based on Trellis diagram. By comparing the number of time quanta of last occurrence of the current network state with that of next occurrence of the current network state, the mechanism predicts the contents of the incoming resource management cell.

The "predictive mechanism" in EFCI, PRCA, and EPRCA is rather primitive. Here, we briefly describe the existing ABR flow control mechanisms.

(1) EFCI (explicit forward congestion indication)[1]: When receiving data from the source, the switch indicates network congestion by setting the EFCI bit of the pay load type field of ATM cells. Feedback in the form of a resource management cell is returned to the data source periodically. The source increases its rate by default, and decreases its rate when receiving a resource management cell. In other words, the data source always predicts that the value of CI field in the incoming resource management cell is 0. Note that CI =0 indicates that the network is not in a congested state. If the data source does not receive a resource management cell within a fixed time interval, the prediction is correct. If it receives a resource management cell, the prediction is wrong.

(2) PRCA (proportional rate control algorithm)[5]: In the absence of congestion, a resource management cell is sent back to the data source. A source always decreases its rate unless it receives a resource management cell. In other words, the predictive mechanism of the data source always expects the incoming resource management cell with CI = 1.

(3) EPRCA (enhanced PRCA)[6]: A data source issues a resource management cell for each fixed number of user data cells. The destination end system may change the CI bit of the resource management cell to one to indicate network congestion if the EFCI bit of the user data cell is one. The rate adjustment strategy is similar to that of PRCA.

(4) The Intelligent Congestion Control[7]: The approach distinguishes the cell rates among VC connections. The mechanisms mentioned in (1) through (3) adjust the rate without considering the cell rate of each individual VC connection. The intelligent mechanism tries to calculate the optimal bandwidth of each connection and adjusts the rates of VC connections accordingly. The rate adjustment scheme is similar to that of PRCA.

Note that we do not attempt to propose a new congestion control algorithm. Rather, we plan to use a predictive mechanism as an "add-on" to the congestion algorithms such as EPRCA, PRCA, or EFCI. This paper is organized as follows. In Section 2, we delineate the predictive mechanism and describe the steps of predicting. In Section 3, the effectiveness of using the predictive mechanism to reduce the excessive cells is demonstrated. A conclusion is given in Section 4.

2. The Predictive Mechanism

As mentioned earlier, there is a time lag between the network sends back the resource management cell to notify the data sources to adjust their rates and the data sources actually receive the message. The time lag does not cause any problem if the network speed is low, or if the network is confined to a small area, such as a LAN. In the former case, when the network speed is low, there are not much of cells can enter the network during the period of time lag. The small amount of excessive cells is less likely to make the network congestion worse. In the latter case, if the network is rather small, the propagation delay is short, and the amount of excessive cells is also limited. However, in ATM networks, the transmission rate is rather high, and the area the network covers is broad. The amount of excessive cells can be tremendous, and can make the congestion condition even worse. On the other hand, if the load is light, the network is under utilized during the period of time.

A predictive mechanism at the source is necessary for the sources to know beforehand what the network condition is, and make a quick response. As a result, the purposes of the predictive mechanism are two fold. First, the mechanism reduces the amount of excessive cells if the network is in a near congestion state. Second, the mechanism increases the network utilization, if the network load is light. We describe the concept of the predictive mechanism first. We then show how the predictive mechanism can be fitted into the current congestion algorithms including EFCI, PRCA, and EPRCA.

Figure 1 shows the predictive mechanism. For ease of explanation, we describe the network domain and user domain separately.

The Network Domain

The destination end system issues a resource management cell to notify the data source to increase or decrease its rate. Here, we use "1" to represent rate decrease, and "0" to represent rate increase. When the network load is light, a bit "0" is issued to the data source to increase rate. Or if the network load is heavy, a bit "1" is issued to the data source to reduce rate. The CI field of the resource management cell is used to carry the information.

The User Domain

The user domain must predict the contents of the incoming bits and adjust its rate before receiving the resource management cells. The predictive mechanism is based on the Trellis Diagram. In the Trellis diagram, the state of the network is represented by a sequence of consecutive bits that the network issues. In this paper, we take three consecutive bits to represent the state of the network (i.e., the CI bit of three consecutive resource management cells.) Later we will show how the length of bits can affect the accuracy of prediction. There are eight possible states represented by A through H. .

The Trellis diagram, as shown in Figure 2, illustrates the state transition when receiving a resource management cell. As shown in the figure, when the network is in state A, it can remain in state A if it predicts that a bit "0" will be received in the next time quantum. Otherwise, if it predicts a bit "1", then at least four time quanta are required to return to state A. Similarly, If a network is in State B, it either takes three quanta to return to B(if it predicts "0"), or takes three quanta or more to return to B (if it predicts "1").

Table 2 below shows the number of quanta required to return to the same network state if the mechanism predicts the CI field of the incoming resource management cell is "0" and "1," respectively.

The Algorithm

The fundamental principle is based on the idea that the network will return to the same state at about the same number of time quanta. The mechanism predicts what's coming based on the algorithm outlined as follows.

(1) First, the mechanism goes through a warm up period. Wait until all states have appeared at least once, the mechanism then begins the predicting process.

(2) Find the number of quanta have elapsed between the current state and the last occurrence of the current state.

(3) In Table 2, find the number of quanta required for next occurrence of current state for predicting "0" and "1," respectively. If the number of time quanta calculated in Step (2) is in the range of predicting "1", then the mechanism predicts "1." Else the mechanism predicts "0."

We use the following examples to illustrate the algorithm.

Suppose the network is sending a bit stream of 111000111000... consecutively. During the first six bits, the mechanism is in the warm up period. The transition of the network state is shown as follows.

$$A \xrightarrow{1} B \xrightarrow{1} D \xrightarrow{1} H \xrightarrow{0} G \xrightarrow{0} E \xrightarrow{0} A$$

It then begins the predicting process. From Table 2, if the mechanism predicts "0", the network state will return to state A immediately. However, if the mechanism predicts "1", it will take more than one quanta to return to state A. Since the last time state A occurs is six quanta back, the mechanism predicts "1." The path of the network states is shown in Figure 3.

Predicting "1" leads the network state to B. At state B, the last occurrence of state B is six quanta back. Once again, the mechanism predicts "1." The network is now in state D. Because state D appears six quanta back, according to Table 2, the mechanism predict "1." Similarly, we can show that the network state will go through states H, G, E, etc.

We now use another string to show the predicting process.

11111000111110001111100011111000...

During the warm up period, the state transition is shown as follows.

$$A \xrightarrow{1} B \xrightarrow{1} D \xrightarrow{1} H \xrightarrow{1}$$

$$H \xrightarrow{1} H \xrightarrow{0} G \xrightarrow{0} E \xrightarrow{0} A$$

Table 3 shows the predicting process beginning with state A.

Note that the number in the middle column with an asterisk indicates that the predicting value was wrong and was corrected when receiving the actual value from the network. The state leading shows the state after receiving the correct value. Figure 4 below shows the Trellis diagram.

Assume the bit string is periodic in nature. Within each period, the string begins with a substring of 1's followed by a substring of 0's. The lengths of the substrings 1's and 0's are a and b, respectively. By going through the Trellis diagram, we can obtain the error bit ratio for bit "1" and "0," respectively, as shown in the table below.

Effect of Order of the Predictive (OP) Mechanism on Predicting Errors

The order of the predictive mechanism (OP) is defined as the number of consecutive bits used to represent the state of the network. In general, for a periodic string

$$\underset{a}{\underleftrightarrow{111\ldots111}}\underset{b}{\underleftrightarrow{000\ldots000}}\underset{a}{\underleftrightarrow{111\ldots111}}\underset{b}{\underleftrightarrow{000}}\ldots($$

where the lengths of "1" and "0" in each period are a and b, respectively, there are two kinds of predicting errors that can occur: (1) intrasubstring errors (2) intersubstring errors, as shown in Figure 5. An intrasubstring error occurs within a substring whereas an intersubstring error occurs in the transition of two substrings.

The following two lemmas state how the order of the predictive mechanism (OP) affects the number of error predicting bits.

<Lemma 1> If OP is greater than or equal to both lengths of substrings (a and b), neither intrasubstring errors nor intersubstring errors can occur.

<Proof> The mechanism can make a prediction error only if the state that includes the error bit occurs previously. If OP is greater than the length of the substring, then intersubstring error cannot occur. This is because an intersubstring error would increase the length of the substring by one, which cannot be a state that happens previously. If OP is equal to the length of the substring, intersubstring error cannot occur either since the last occurrence of the state that contains exactly the substring is many quanta away. The claim that intrasubstring errors cannot occur is based on the fact that the state that contains the error bit does not exist previously.

<Lemma 2> If OP is less than the length of a substring (a or b), there is an intrasubstring bit error and an intersubstring bit error.

<Proof> Since the substring is longer than the OP, the string of OP can overlap with the substring to the left or to the right. Assuming that the destination end system sends back the resource management cells coincides with the substring from left to right. An intrasubstring error occurs when the OP string overlaps with the left side of the substring, and an intersubstring error occurs when the OP string overlaps with the right side of the string. In both cases, making a correct prediction would violate the fundamental principle of the predictive mechanism.

By simple calculations through Lemma 1 and Lemma 2, we can obtain the error rate ratio (ERR) as follows.

If $OP \ge a$, and $OP \ge b$, then $ERR=0$

If $OP \ge a$, and $OP \le b$, then $ERR = \frac{2}{a+b}$

If $OP \le a$, and $OP \ge b$, then $ERR = \frac{2}{a+b}$

If $OP < a$, and $OP < b$, then $ERR = \frac{4}{a+b}$

where ERR is the error bit rate ratio, that is, the total number of error predicting bits divided by the total number of predicting bits. Similarly, α, the error bit rate ratio on bit value "1, " is defined as the number of error predicting bits on "1" divided by the number of predicting bits on "1." And β, the error bit rate ratio on bit value "0," is defined as the number of error predicting bits on "0" divided by the number of predicting bits on "0."

Adapting the Predictive Mechanism to EPRCA

We let the data rate of the source be R. The destination end system checks the EFCI field of data cells and compute the percentage of cells with EFCI =1 as follows..

$$PCE = \frac{CE}{R} * 424 \qquad 0 \le PCE \le 1$$

where 424 is the constant that converts the data source rate (R) from bits per second to cells per second, CE is the number of data cells with EFCI = 1, and PCE is the percentage of data cells with EFCI =1.

The range of PCE is from 0 to 1, and is divided into a fixed number of notches. When PCE advances one notch, a resource management cell with CI =1 is sent to the source end system. Else, if PCE drops one notch, a resource management cell with CI = 0 is sent to the source end system. For instance, the range of PCE is divided into 10 notches. Table 5 below shows the resource management cells issued and their contents.

Note that, in EFCI, the resource management cells are generated in fixed time interval. In PRCA, the resource management cells are generated at a rate proportional to the data source rate. In both PRCA and EFCI, the resource management cells are generated by the destination end system. In EPRCA, the resource management cells are generated by the source end system. For the predictive approach, whether the source or the destination generates the resource management cells won't make any difference.

About the timing of generating the resource management cells, the EFCI mechanism generates a resource management cell on a fixed time interval. The PRCA generates resource management cells at a rate proportional to the data source rate. For the predictive approach, we suggest that a resource management cell is generated when the rate change has achieved a predetermined amount.

In EPRCA, the CI bit in the resource management cell indicates whether there is a congestion or not. For the predictive approach, the CI =1 and CI = 0, represent rate increase and decrease, respectively.

3. Performance Evaluation

As stated, excessive cells are defined as the data cells that enter the network during the period of propagation of the resource management cell back to the data source. If the propagation delay of the resource management cell is TP, and the current data source rate is required to reduce by the amount ΔR Mbps, then the amount of excessive cells is $\Delta R * TP / 424$.

This amount of excessive cells can be avoided if the predictive mechanism makes a correct prediction when the resource management cells are generated. However, if the predictive mechanism makes a wrong prediction and increases the data source rate, then more excessive cells enter the network. Assuming the amount of increment is also ΔR, the amount of excessive cells for each error prediction is $2\Delta R * TP / 424$.

Let the length of the string be L (i.e., the consecutive bits contained in resource management cells), the following parameters can be derived.

The number of 1's in the string: $aL/(a+b)$.

The number of 0's in the string: $bL/(a+b)$.

The number of error bits on "1": $\alpha a L/(a+b)$.

The number of error bits on "0": $\beta bL/(a+b)$.

The amount of reduction of excessive cells due to the correct prediction of each incoming bit is $\Delta R * T$ where ΔR and T are the amount of rate reduction, and the duration of a time quantum, respectively. Since the amount of rate increment and decrement for each receiving resource management cell can be different, we use ΔR_i and ΔR_d to represent the amount of rate increment and rate decrement, respectively. The total amount of reduction of excessive cells due to correct prediction of incoming bits of the string thus is

$$\frac{\Delta R_i * T * a * L * (1-\alpha)}{a+b}$$

The amount of increase of excessive cells for errors in predicting the incoming bits of the string is

$$\frac{\Delta R_d * T * a * L * \alpha}{a+b}$$

As a result, the amount reduction of excessive cells for the predictive mechanism is

$$\frac{T * a * L * (\Delta R_i (1-\alpha) - \Delta R_d \alpha)}{a+b}$$

However, the amount of excessive cells that can enter the network without the predictive mechanism is

$$\frac{\Delta R_i * T * L * a}{a+b}$$

thus, the percentage of reduction of excessive cells ($PREC$) is calculated as follows.

$$\frac{\Delta R_i (1-\alpha) - \Delta R_d \alpha}{\Delta R_i}$$

The following parameters are used for computing $PREC$.

Time quantum=0.015 sec

(a, b) = (3,3), (5,3), (6,4), (7,5), (8,6)

Minimum Cell Rate (MCR) = 10 Mbps, Peak Rate (PR) = 100 Mbps

Each increment of rate ΔR_i = $(PR - MCR)/a$

Each decrement of rate ΔR_d = $(PR - MCR)/b$

Initial Cell Rate (ICR) = 100 Mbps

The percentage of reduction of excessive cells ($PREC$) for various values of (a, b) is shown in the table below.

4. Conclusion

We have shown a simple predictive mechanism for improving the "responsiveness" of the feedback control loop algorithm for managing the ABR traffic. The predictive algorithm is based on a Trellis diagram, where each state is represented by a fixed number of bits. The contents of the bits are taken from the CI field of the resource management cells. The fundamental principle of predicting is based on the idea that network state will appear approximately at about the same time quanta. We also demonstrate the improvement of reduction of excessive cells through the predictive mechanism.

References:

1. K. Ramakrishnan and R. Jain, A Binary Feedback Scheme for Congestion Avoidance in Computer Networks, ACM Transactions on Computer Systems, Vol. 8, May 1990, pp. 158-181.
2. J. Bae, and T. Suda, Survey of Traffic Control Schemes and Protocols in ATM Networks, Proceedings of the IEEE, Vol. 79, No. 2, Feb. 1991, pp. 170-189.
3. F. Bonomi and K Fendick, The Rate-Based Flow Control Framework for the Available Bit Rate ATM Services, IEEE Network, Vol. 9, Mar/Apr, 1995, pp. 25-39.
4. T. M. Chen, S. Liu, and V. Samalam, The Available Bit Rate Service for Data in ATM Networks, IEEE Communications Magazine, May 1996, pp. 56-71.
5. A. Barnhart, Use of Extended PRCA with Various Switch Mechanism, ATM Forum/94-0898, Sept, pp. 26-29, 1994.
6. L. Roberts, Enhanced PRCA (Proportional Rate Control Algorithm) ATM Forum/94-0735R1, Aug. 1994.
7. K. Y. Siu and H. Y. Tseng, Intelligent Congestion Control for ABR Service in ATM Networks, Computer Communications Review, Vol, 24, Oct. 1995, pp. 81-105

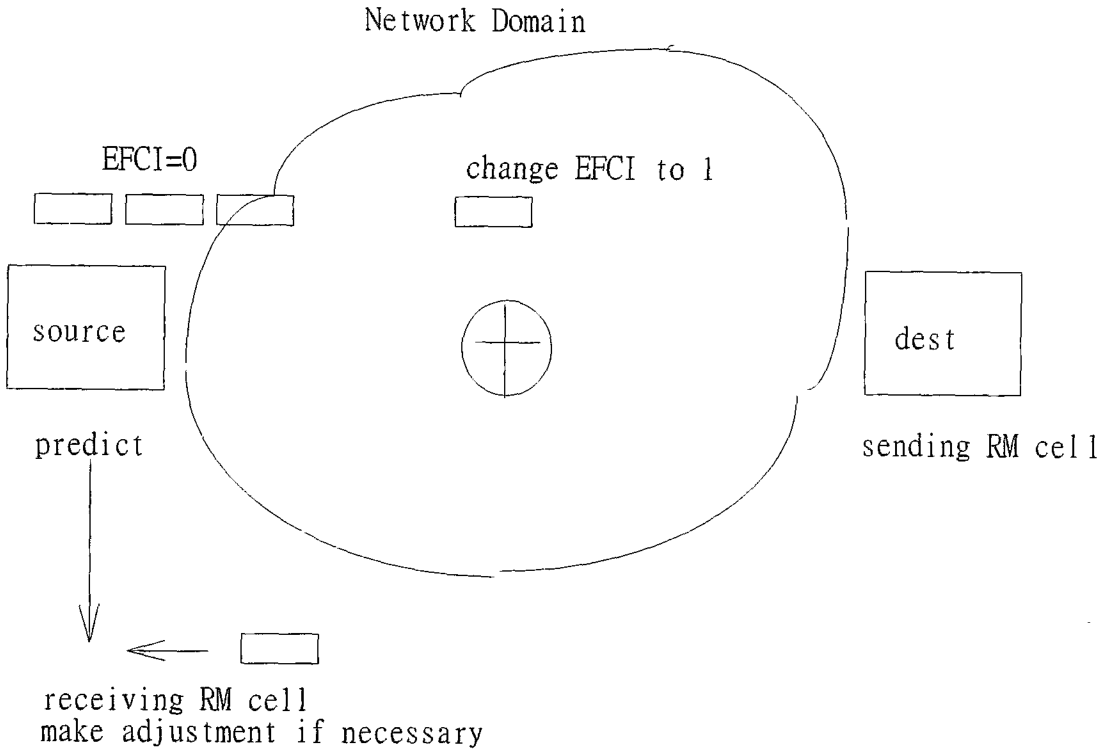

Figure 1: The Mechanism of Predictive Approach

A	B	C	D	E	F	G	H
000	001	010	011	100	101	110	111

Table 1: The States of the Trellis Diagram

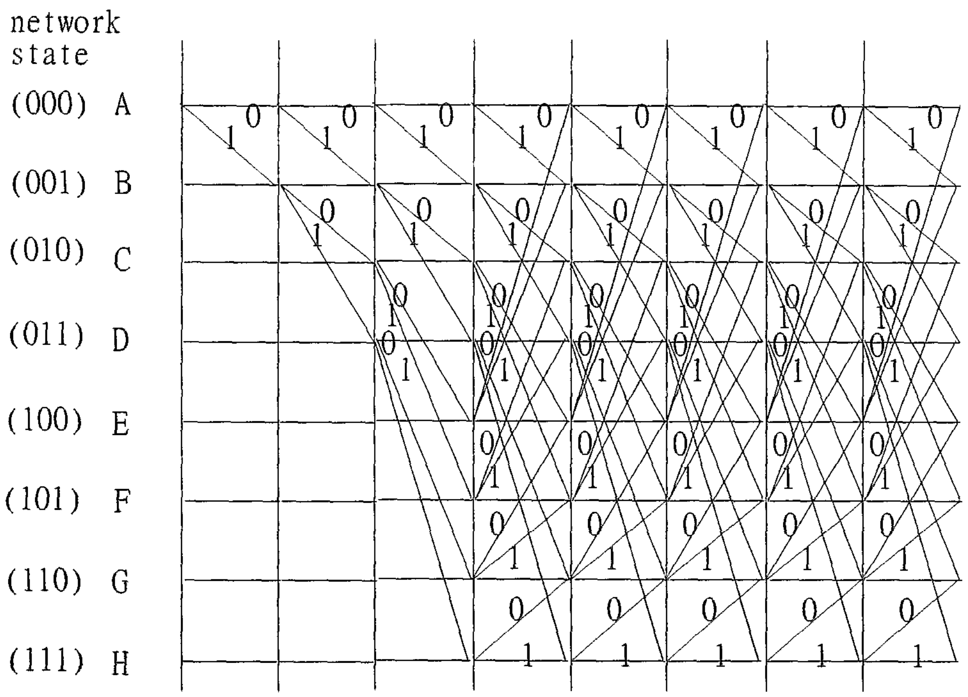

Figure 2: The Trellis Diagram of Eight States

Network state	Predicting bit	Number of Quanta
A	0	num=1
	1	num>=4
B	0	num=3
	1	num >=4
C	0	num >=3
	1	num =2
D	0	num=3
	1	num >=4
E	0	num >=4
	1	num = 3
F	0	num=2
	1	num>=3
G	0	num>=4
	1	num=3
H	0	num>=4
	1	num=1

Table 2: The Number of Quanta Required to Return to the Same State

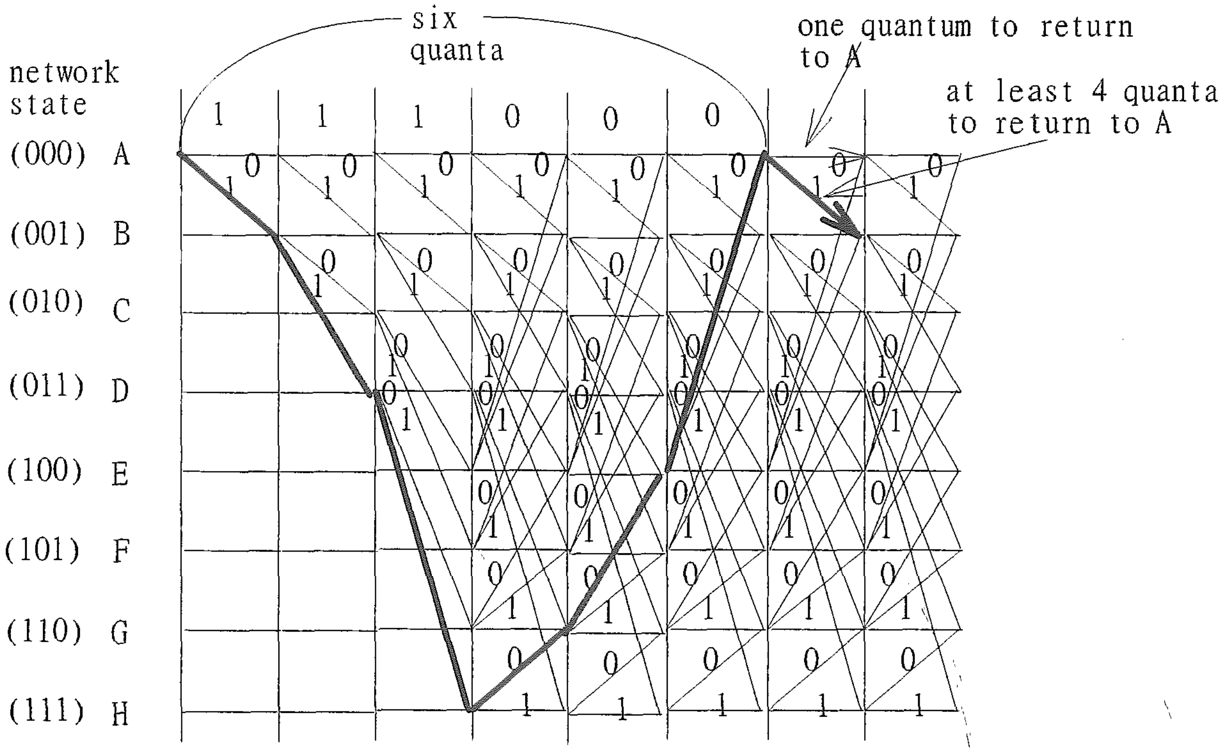

Figure 3 : The Predicting Path of the Trellis Diagram

Number of Quanta of Last Occurrence	Predicting”0“ or “1”	Leading to State
8	1	B
8	1	D
8	1	H
8	0*→(1)	H
1	1	H
1	1*→(0)	G

Table 3 : The Predicting Process for a Periodic String

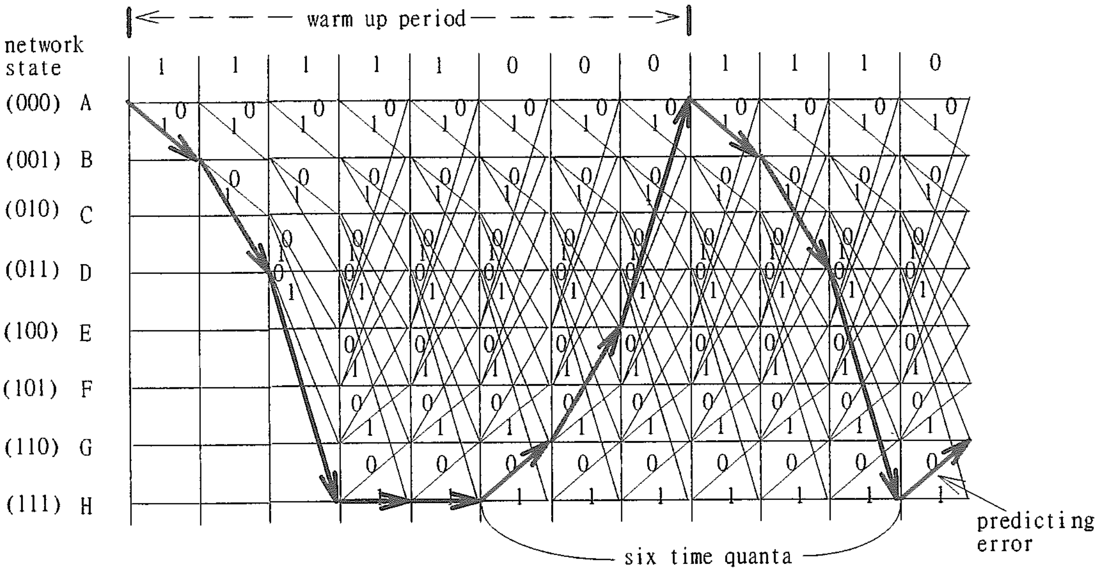

Figure 4: The Predicting Process in The Trellis Diagram for Bit String (1111100011111000.....)

(a,b)	(3,3)	(5,3)	(6,4)	(7,5)	(8,6)
α	0	1/5	1/3	2/7	1/4
β	0	1/3	1/2	2/5	1/3
ERR	0	1/4	2/5	1/3	2/7

Table 4: The Error Bit Rate Ratio for Various Lengths of Substrings

where *ERR* : error bit rate ratio

α : the error bit rate ratio on bit value "1"

β : the error bit rate ratio on bit value "0"

(a,b) : the lengths of substrings "1" and "0" in the periodic string, respectively

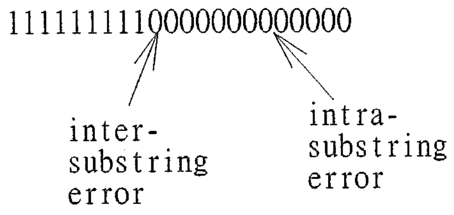

Figure 5: Types of Error Predicting Bits

PCE	0.2	03	0.4	0.3	0.2	0.4	6
RM Contents	nil	1	1	0	0	1	1

Table 5: An Example of Resource Management Contents

(a,b)	(3,3)	(5,3)	(6,4)	(7,5)	(8,6)
PREC	100	47	58	66	71

Table 6: The *PREC* for Various Values of (a,b)