

信州大学工学部

学士論文

割り込みのある待ち行列通信システムで  
情報源を符号化したときの情報鮮度

指導教員 西新 幹彦 准教授

学科 電子情報システム工学科  
学籍番号 22T2138B  
氏名 古市 亮太

2026年3月24日

## 目次

1	はじめに	1
2	AoI とその評価方法	1
2.1	AoI の定義 . . . . .	2
2.2	AoI の導出方法 . . . . .	3
3	サービスレートによる符号化	5
3.1	多値情報源を符号化した時の AoI . . . . .	5
3.2	AoI の最適化 . . . . .	7
4	パケット数による符号化	10
5	シミュレーションによる検証	18
6	まとめ	19
	謝辞	19
	参考文献	19
	付録 A 漸化式の解の証明	21
	付録 B ソースコード	22

## 1 はじめに

テレビ局や配信事業者が、スタジアムやコンサート会場で行われるスポーツ試合や音楽パフォーマンスを撮影し、遠隔地の視聴者にライブ中継する場合を考える。現場を捉えるカメラやマイクと、家庭のテレビやスマートフォンは互いに離れていて、その間には衛星回線やインターネットなどの通信路が設置されている。カメラやマイクで捉えた映像や音声の情報は、エンコードや伝送の過程を通過する間に古くなり、視聴者の画面やスピーカーに再生される時点では、程度はともかく過去の一瞬を切り取ったものになってしまう。さらに、表示されている映像も、次のフレームが届くまで時間の経過とともに古くなり続ける。しかし「ライブ」体験の本質からして、視聴者が感じる臨場感は、実際の現場で起きている事象とできるだけ同期している、つまり情報が新しいことが望ましい。このような情報の新しさは情報鮮度 (Age of Information: AoI) と呼ばれる。AoI が小さいほど情報が新しいことを意味する。以下 AoI と呼ぶ。

待ち行列理論の立場に基づく従来研究 [1] では、センサでサンプリングされた情報が送信機に到着した順番に処理される FIFO (First In First Out) のシステムの AoI について考えられている。また、別の研究 [2] では、送信機の処理中に新しい情報が到着したら古い情報の処理を中断し割り込みをして優先的に処理する LIFO (Last In First Out) を用いたシステムの AoI について考えられている。その後、この問題は、システム中に符号器が追加され、情報理論的な最適化問題として考えられるようになった。そして研究 [3] で情報を符号語に変換し、シンボルごとサービスレートが変わる通信モデルを設定し、二値情報源での AoI の解析が行われた。さらに研究 [4] で、同じ通信モデルでシンボル数が一般に  $n$  個の情報源に対して、AoI の解析が行われた。

先行研究 [4] では、各シンボルに対して個別にサービスの分布が定められており、各分布は対応するシンボルの符号語長を期待値とした指数分布となっている。この研究で得られた AoI の式に対し、本研究では、平均サービス時間に関する制約の下でサービスレートを最適化した。また、サービスレートを固定し、符号語長の違いをパケット数で表現する通信モデルを考えた。具体的には、各シンボルを  $l_i$  個のパケットに分割し、サーバはこれらのパケットを一定レートで順次処理する。このときの AoI を解析した。

## 2 AoI とその評価方法

図 1 のような通信モデルを考える。情報源からサーバ、サーバからモニタまでの経路が通信路である。即ち、図の矢印は通信路を表している。このシステムでは情報源からポアソン過程に従い出力されたシンボルは、発生と同時に図 1 の矢印で表された通信路を通りサーバでシン



図1 システム図

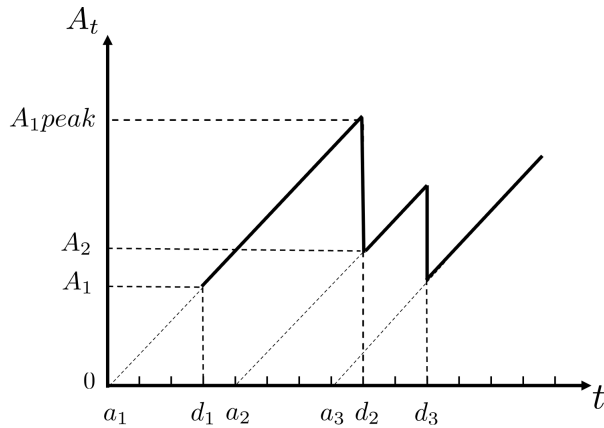


図2 AoIの遷移

ボルの処理が行われる．その後，遠隔地のモニタへ表示される．一般に FIFO より LIFO を用いたモデルのほうが AoI が小さいことが明らかである．よって，サーバにシンボルが保持されている状態で新たなシンボルが情報源から発生し，サーバに到着した場合，既にサーバで処理されているシンボルは棄却され，新しいシンボルがサーバで処理されモニタに表示される．

## 2.1 AoI の定義

本研究では，モニタに表示されている情報の年齢を考察する．情報の年齢は情報が発生してからの経過時間のことである．以降これを単に AoI と呼ぶ．

定義より，時刻  $t$  におけるモニタの AoI  $A_t$  は表示されている情報の発生時刻を  $a$  とすると

$$A_t = t - a \quad (1)$$

と書ける．モニタの AoI の時間的変化の具体例を図 2 に示す．実線がモニタの AoI である．時刻  $a_1$  に情報源から発生したシンボル  $X_1$  は時刻  $d_1$  でサーバの処理が終了し，モニタに表示される．シンボル  $X_1$  が表示された瞬間の値は，

$$A_1 = d_1 - a_1 \quad (2)$$

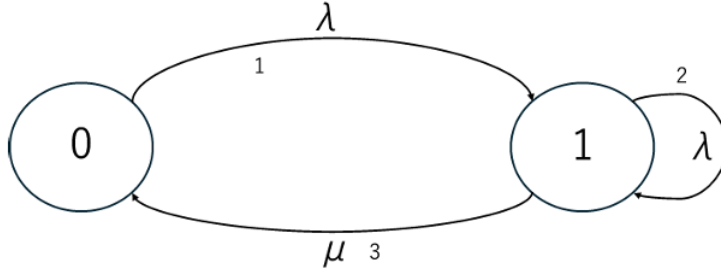


図3 図1の通信システムの状態遷移図

表1 図3のマルコフ連鎖の遷移表

$l$	$q_l \rightarrow q'_l$	$\lambda^{(l)}$	$x A_t$	$A_t$	$v_{ql} A_t$
1	$0 \rightarrow 1$	$\lambda$	$[x_0 \ 0]$	$\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$	$\begin{bmatrix} v_{00} & 0 \end{bmatrix}$
2	$1 \rightarrow 1$	$\lambda$	$[x_1 \ 0]$	$\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$	$\begin{bmatrix} v_{10} & 0 \end{bmatrix}$
3	$1 \rightarrow 0$	$\mu$	$[x_1 \ *]$	$\begin{bmatrix} 0 & * \\ 1 & * \end{bmatrix}$	$\begin{bmatrix} v_{11} & * \end{bmatrix}$

と書ける。また、時刻  $a_2$  で情報源から発生したシンボル  $X_2$  は、時刻  $d_2$  でモニタに表示される。この時、モニタではシンボル  $X_1$  からシンボル  $X_2$  に情報が更新される。このとき、AoIの値が  $A_{1peak}$  から  $A_2$  に瞬間的に下がる。このようにモニタではシンボルが更新されていくため、図2のようなノコギリ状の実線のグラフとなる。これが、AoIの時間的変化である。

本研究では、AoIの時間平均を評価する。AoIの時間平均  $A_{ave}$  は、

$$A_{ave} = \lim_{T \rightarrow \infty} \frac{1}{T} \int_0^T A_t dt \quad (3)$$

と書ける。

## 2.2 AoIの導出方法

この節では、AoIの導出方法の具体例を示す。図1の通信モデルを考える。情報源からのシンボルが、到着レート  $\lambda$  のポアソン過程に従って到着すると仮定する。また、サーバでのサービスレートをシンボルの種類によらず  $\mu$  とする。

このシステムでは、サーバは2つの状態をとる。状態0はサーバが空の状態、状態1はサーバでシンボルが処理されている状態を表すとする。互いの状態の遷移関係を自己遷移も含め表

したのが図 3 である．このような図を状態遷移図と呼ぶ．それぞれの状態の遷移を区別するため図 3 のように遷移番号をつける．遷移番号 1 は，サーバが空の状態の情報源からサーバにシンボルが到着した遷移を表す．このときの遷移は到着レート  $\lambda$  で起こる．同様に遷移番号 2 は，サーバでシンボル処理中に新しいシンボルがサーバに到着したことを表している．このときの遷移も到着レート  $\lambda$  で起こる．この時，サーバで処理中のシンボルは棄却され，新しいシンボルの処理が行われる．状態 3 はサーバで処理が終了したことを表している．このときの遷移のレートは  $\mu$  である．この状態遷移図から表 1 のような表が得られる． $l$  は遷移番号， $q_l$  は遷移前の状態， $q'_l$  は遷移後の状態， $\lambda(l)$  は遷移のレートである．状態遷移の直前におけるモニタとサーバの年齢をそれぞれ  $x_0, x_1$  と定義し，これらを横ベクトル  $\mathbf{x} = [x_0, x_1]$  で表す．ここで， $\mathbf{x}A_l$  はサーバの状態が変化した直後のモニタとサーバの年齢を表し， $A_l$  は状態遷移を記述するための写像行列である．また，状態  $q_l$  における年齢ベクトルの期待値を  $v_{q_l} = [v_{q_l,0}, v_{q_l,1}]$  と定義する．\* は値を持たないことを意味する．

表 1 で得られた値から， $Q$  を状態の集合， $q$  を遷移後の状態， $b_q$  を各状態の傾きとする．傾きとはシンボルがサーバまたはモニタなどで年齢をとることを指すので傾きは 0 か 1 である． $b_0 = [1 \ 0]$  は状態 0 のときモニタの傾きが 1 でサーバの傾きが 0 ということを表している．未知数  $v_{q_l}$ ， $\pi_q$  の連立方程式

$$v_q \sum_{l \in I_q} \lambda^{(l)} = b_q \pi_q + \sum_{l \in I'_q} \lambda^{(l)} v_{q_l} A_l, \quad q \in Q \quad (4)$$

を立てる．またこの式は，状態  $q$  における年齢ベクトルの期待値  $v_q$  に関する平衡方程式を表している．左辺は状態  $q$  から出ていく遷移による期待値の減衰を，右辺は状態  $q$  に入ってくる遷移による期待値の流入を表しており，両者が平衡状態で等しくなることを示している． $\pi_q$  は定常確率である．定常状態のときに成り立つ式は，

$$\pi_q \sum_{l \in I_q} \lambda^{(l)} = \sum_{l \in I'_q} \lambda^{(l)} \pi_{q_l}, \quad q \in Q \quad (5)$$

と書ける． $\pi_q$  は確率であるため，

$$\sum_{q \in Q} \pi_q = 1 \quad (6)$$

が成り立つ．

さらに，求めるモニタの時間平均 AoI を  $\Delta$  とすると，

$$\Delta = \sum_{q \in Q} v_{q,0} \quad (7)$$

となる．実際に，表 2 から得られた値を式 (4) および式 (5) に代入すると，

$$[v_{00} \ v_{01}]\lambda = [1 \ 0]\pi_0 + \mu[v_{11} \ *] + \lambda[v_{10} \ 0], \quad (8)$$

$$[v_{10} \ v_{11}](\lambda + \mu) = [1 \ 1]\pi_1 + \lambda[v_{00} \ 0] + \lambda[v_{10} \ 0], \quad (9)$$

$$\pi_0\lambda = \mu\pi_1, \quad (10)$$

$$\pi_1(\mu + \lambda) = \lambda\pi_0 + \lambda\pi_1 \quad (11)$$

を得る．式 (10), (11) について解くと，

$$\pi_0 = \frac{\mu}{\mu + \lambda}, \quad (12)$$

$$\pi_1 = \frac{\lambda}{\mu + \lambda} \quad (13)$$

が得られ，これらを用いて式 (8), (9) について解くと  $v_{00}, v_{10}$  は，

$$v_{00} = \frac{1}{\mu} \left( \frac{\lambda^2 + \mu^2 + 3\lambda\mu}{(\mu + \lambda)^2} \right), \quad (14)$$

$$v_{10} = \frac{\mu}{\lambda} \left( \frac{\mu + 2\lambda}{(\mu + \lambda)^2} \right) \quad (15)$$

と表される．よって求めたい時間平均 AoI は，

$$\Delta = \sum_{q \in \mathcal{Q}} v_{q0} \quad (16)$$

$$= v_{00} + v_{10} \quad (17)$$

$$= \frac{\mu + \lambda}{\mu\lambda} \quad (18)$$

である．

### 3 サービスレートによる符号化

本章では，通信システムに符号器を追加し，各シンボルに応じてサービスレートが変化する通信モデルを扱う．1 節ではシンボルの種類を  $n$  個に一般化した時のモニタの AoI を導出した先行研究 [4] の紹介し，2 節ではその結果に対してサービスレートを最適化する．

#### 3.1 多値情報源を符号化した時の AoI

先行研究 [4] の紹介をする．図 4 のような通信システムを考える．情報源のシンボルは  $n$  種類である．シンボルは符号器にポアソン到着し，符号器によって符号語に変換されサーバで処

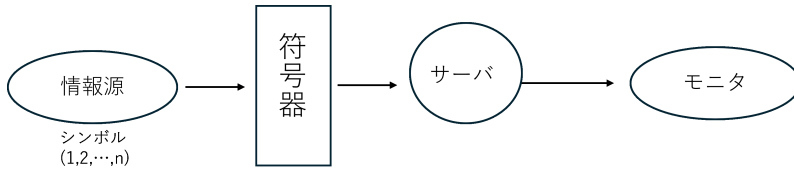


図4 サーバで割り込みのある符号器を追加した通信システム

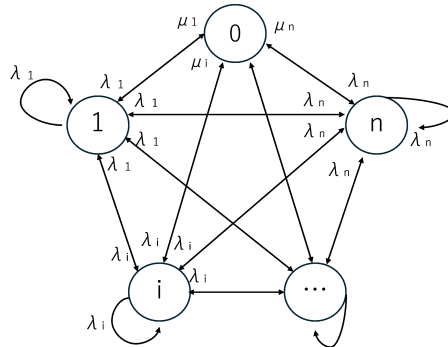


図5 図4の状態遷移図

理される．サーバの処理が終了するとモニタに映し出される．サーバでのサービス時間は符号語長を期待値とした指数分布によって決定する．符号語長は整数であるが，サービス時間を実数で表すため実数として扱う．システムでは，新しい情報を常に送りたい．よって，サーバでシンボルが処理されているときに新しいシンボルが到着した場合，処理しているシンボルを棄却し，新しいシンボルが処理される．AoI は 2 章で説明した方法で導出する．状態遷移図は図 5 のようになる．またそこから表 2 が得られる．このとき状態 0 がサーバが空の状態．状態 1 がシンボル 1 を処理している状態．同様にシンボルは  $n$  種類のため状態  $n$  まで存在する．任意の 2 つの状態間には互いに遷移が存在し，状態 0 を除くすべての状態には自己遷移が存在する．ここで，状態  $j$  から状態  $i$  への遷移にはラベル  $(j, i)$  を付与する．これらの遷移は，サーバが空の状態に対応するラベル  $(i, 0)$  の場合と，サーバでいずれかのシンボルが処理中である場合のラベル  $(j, i)$  の場合に区別できる．2 章と同様に連立方程式を立てて解くと，AoI  $\Delta$  は，

$$\Delta = \frac{1}{\sum_{i=1}^n \frac{\mu_i \lambda_i}{\mu_i + \lambda}}, \quad \lambda = \sum_{i=1}^n \lambda_i \quad (19)$$

と求まる [4].

表2 図5のマルコフ連鎖の遷移表

$l$	$q_l \rightarrow q'_l$	$\lambda^{(l)}$	$x A_t$	$A_t$	$v_{q_l} A_t$
(1,0)	$1 \rightarrow 0$	$\mu_1$	$[x_1 \ *]$	$\begin{bmatrix} 0 & * \\ 1 & * \end{bmatrix}$	$\begin{bmatrix} v_{11} & * \end{bmatrix}$
(2,0)	$2 \rightarrow 0$	$\mu_1$	$[x_1 \ *]$	$\begin{bmatrix} 0 & * \\ 1 & * \end{bmatrix}$	$\begin{bmatrix} v_{21} & * \end{bmatrix}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
( $i$ ,0)	$i \rightarrow 0$	$\mu_i$	$[x_1 \ *]$	$\begin{bmatrix} 0 & * \\ 1 & * \end{bmatrix}$	$\begin{bmatrix} v_{i1} & * \end{bmatrix}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
(1,1)	$1 \rightarrow 1$	$\lambda_1$	$[x_0 \ 0]$	$\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$	$\begin{bmatrix} v_{11} & 0 \end{bmatrix}$
(2,1)	$2 \rightarrow 1$	$\lambda_1$	$[x_0 \ 0]$	$\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$	$\begin{bmatrix} v_{21} & 0 \end{bmatrix}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
( $n$ ,1)	$n \rightarrow 1$	$\lambda_1$	$[x_0 \ 0]$	$\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$	$\begin{bmatrix} v_{n1} & 0 \end{bmatrix}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
( $j$ , $i$ )	$j \rightarrow i$	$\lambda_i$	$[x_0 \ 0]$	$\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$	$\begin{bmatrix} v_{i0} & 0 \end{bmatrix}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
(0, $n$ )	$0 \rightarrow n$	$\lambda_n$	$[x_0 \ 0]$	$\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$	$\begin{bmatrix} v_{n,0} & 0 \end{bmatrix}$

### 3.2 AoI の最適化

先行研究では、サービスレート  $\mu_i$  は符号器の設定といえる。したがって、この設定によって AoI が変化する。本節では、 $\mu_i$  を変数として AoI の最小値を求める。

$\frac{1}{\mu_i}$  はシンボル  $i$  の平均サービス時間の期待値である。到着率が高いシンボルには、より多くの処理能力を割り当てたい。しかし、全体の処理能力は有限であるため、システム全体の平均サービス時間を  $\delta$  に固定する制約のもとで最適化を行う。 $p_i$  はシンボル  $i$  の到着確率とする。即ち、 $p_i = \frac{\lambda_i}{\lambda}$  である。制約条件を

$$\sum_{i=1}^n \frac{p_i}{\mu_i} = \delta \quad (20)$$

として考える。

上述の問題を制約付き最適化問題として定式化する．未定乗数を  $\alpha$  とし，ラグランジュ関数を

$$\mathcal{L}(\{\mu_i\}, \alpha) = \frac{1}{\sum_{i=1}^n \frac{\mu_i \lambda_i}{\mu_i + \lambda}} + \alpha \left( \sum_{i=1}^n \frac{p_i}{\mu_i} - \delta \right) \quad (21)$$

と定義する．最適解は，ラグランジュ関数の偏微分が 0 となる点で与えられる．すなわち，各  $i = 1, 2, \dots, n$  に対して，

$$\frac{\partial \mathcal{L}}{\partial \mu_i} = 0 \quad (22)$$

が成立する．具体的な計算を行うため， $f(\{\mu_i\}) = \sum_{i=1}^n \frac{\mu_i \lambda_i}{\mu_i + \lambda}$  とおく．まず，式 (21) の第 1 項の偏微分は，

$$\frac{\partial}{\partial \mu_i} \left( \frac{1}{f} \right) = -\frac{1}{f^2} \cdot \frac{\partial f}{\partial \mu_i} \quad (23)$$

$$= -\frac{1}{f^2} \cdot \frac{\lambda_k \lambda}{(\mu_i + \lambda)^2} \quad (24)$$

となる．第 2 項の偏微分は，

$$\alpha \frac{\partial}{\partial \mu_i} \left( \frac{p_i}{\mu_i} \right) = -\alpha \frac{p_i}{\mu_i^2} \quad (25)$$

である．したがって，式 (22) は，

$$-\frac{\lambda_i \lambda}{(\mu_i + \lambda)^2 f^2} - \alpha \frac{p_i}{\mu_i^2} = 0 \quad (26)$$

となる．ここで  $p_i = \frac{\lambda_i}{\lambda}$  より，

$$\alpha = -\frac{\lambda^2 \mu_i^2}{(\mu_i + \lambda)^2 f^2} \quad (27)$$

が得られる．任意の 2 つのシンボル  $k$  と  $l$  に対して，未定乗数  $\alpha$  は共通であるから，

$$\frac{\lambda^2 \mu_k^2}{(\mu_k + \lambda)^2 f^2} = \frac{\lambda^2 \mu_l^2}{(\mu_l + \lambda)^2 f^2} \quad (28)$$

が成立する．両辺を整理すると，

$$\frac{\mu_k^2}{(\mu_k + \lambda)^2} = \frac{\mu_l^2}{(\mu_l + \lambda)^2} \quad (29)$$

となる。両辺の平方根を取ると、

$$\frac{\mu_k}{\mu_k + \lambda} = \frac{\mu_l}{\mu_l + \lambda} \quad (30)$$

である。これを展開して整理すると、

$$\mu_k \mu_l + \mu_k \lambda = \mu_k \mu_l + \mu_l \lambda \Rightarrow \mu_k = \mu_l \quad (31)$$

が得られる。したがって、

$$\mu_1 = \mu_2 = \cdots = \mu_n = \mu \quad (32)$$

が成り立つ。即ち、すべてのシンボルに対してサービスレートは等しい。得られた結果を制約条件に代入すると、

$$\sum_{i=1}^n \frac{p_i}{\mu} = \delta \Rightarrow \frac{1}{\mu} \sum_{i=1}^n p_i = \delta \quad (33)$$

となる。 $\sum_{i=1}^n p_i = 1$  より、

$$\frac{1}{\mu} = \delta \Rightarrow \mu = \frac{1}{\delta} \quad (34)$$

となる。したがって、最適なサービスレートは、

$$\mu_1 = \mu_2 = \cdots = \mu_n = \frac{1}{\delta} \quad (35)$$

である。最適サービスレート  $\mu_i = \frac{1}{\delta}$  を AoI の式に代入すると、

$$\Delta = \frac{1}{\sum_{i=1}^n \frac{\frac{1}{\delta} \cdot \lambda_i}{\frac{1}{\delta} + \lambda}} \quad (36)$$

$$= \frac{1}{\sum_{i=1}^n \frac{\lambda_i}{1 + \delta \lambda}} \quad (37)$$

$$= \frac{1 + \delta \lambda}{\lambda} \quad (38)$$

$$= \delta + \frac{1}{\lambda} \quad (39)$$

となる。

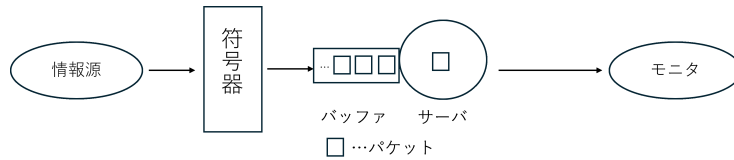


図 6 符号器でパケットに変換した通信システム

## 4 パケット数による符号化

先行研究 [4] では、 $n$  種類のシンボルに対して、異なるレートのポアソン過程に従って発生し、サーバのサービス時間もその種類によって異なるレートの指数分布に従う。サービスレートの違いは、符号器による各シンボルの符号語長の違いを表現している。これに対し、本研究では、各シンボルを複数のパケットに変換することで符号語長を表現する。つまり、シンボルの種類によって変換後のパケット数が変わる。その代わりに、サーバはどのパケットもレート  $\mu$  の指数分布に従って処理する。これは、通信路やサーバの物理的な処理能力が固定的である状況を想定している。

通信システムの構成は図 6 である。情報源から出力されたシンボル  $i$  は符号化の過程を経て  $l_i$  個のパケットに変換される。サーバはこれらのパケットを一つずつ処理し、処理が完了していないパケットはシステム内のバッファに保持される。シンボルの最終パケットの処理が完了したときモニタに映し出される。即ち、同一シンボルに属する全  $l_i$  個のパケットの処理が完了した時点で、初めてモニタの表示が更新される。

サーバがいずれかのシンボルのパケット処理中に、新しいシンボルのパケットが到着した場合、処理中のパケットとバッファのパケットはすべて棄却され、新しく到着したシンボルの最初のパケットからの処理が即座に開始される。これを割り込みという。

サーバの状態は、サーバが空の状態と特定シンボルのパケットを処理している状態である。状態 0 をサーバが空の状態、状態  $(i, k)$  は「シンボル  $i$  処理中、残り  $k$  パケット」を表す ( $k = l_i, l_i - 1, \dots, 1$ )。遷移のパターンは、

- サーバが空の状態から到着レート  $\lambda_i$  で 1 つ目のパケットの処理を開始する。
- サービスレート  $\mu$  で、中間のパケットの処理をする。
- パケットの最後の処理を終えてサーバが空になる。
- 割り込みが到着レート  $\lambda_i$  で発生する。

の 4 つのパターンが考えられる。これらをまとめると状態遷移は図 7 のような基本構造を並べたものであることが分かる。状態  $(j, l_j)$  に遷移後は、状態  $(i, l_i)$  と同じ遷移の仕方である。

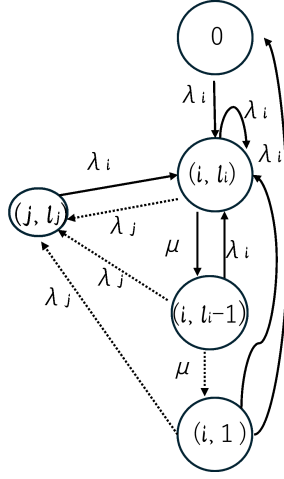


図7 符号器でパケットに変換した通信システムの状態遷移図

また、中間パケットと最終パケットは同じ遷移の仕方であることがわかる。この状態遷移に基づき、2章で説明した手順に従うと表3のような状態遷移表を得る。

2.2節の導出方法より、表3から得られる値を式(4)、式(5)に代入すると、

$$[v_{00} \ v_{01}] \lambda = [1 \ 0] \pi_0 + \sum_{i=1}^n \mu [v_{(i,1)1} \ *] \quad (40)$$

$$[v_{(i,l_i)0}, v_{(i,l_i)1}] (\mu + \lambda) = [1, 1] \pi_{(i,l_i)} + \lambda_i [v_{00} \ 0] + \sum_{j=1}^n \sum_{m=1}^{l_j} \lambda_j [v_{(j,m)0}, 0] \quad (41)$$

$$[v_{(i,k)0}, v_{(i,k)1}] (\mu + \lambda) = [1, 1] \pi_{(i,k)} + \mu [v_{(i,k+1)0}, v_{(i,k+1)1}], \quad 1 \leq k < l_i \quad (42)$$

$$\pi_0 \lambda = \sum_{i=1}^n \mu \pi_{(i,1)} \quad (43)$$

$$\pi_{(i,l_i)} (\mu + \lambda) = \lambda_i \pi_0 + \lambda_i \sum_{j=1}^n \sum_{k=1}^{l_j} \pi_{(j,k)} \quad (44)$$

$$\pi_{(i,k)} (\mu + \lambda) = \mu \pi_{(i,k+1)}, \quad 1 \leq k < l_i \quad (45)$$

表3 図7のマルコフ連鎖の遷移表

$q_t \rightarrow q'_t$	$\lambda^{(l)}$	$x A_t$	$A_t$	$v_{ql} A_t$
$0 \rightarrow (1, l_1)$	$\lambda_1$	$[x_0 \ 0]$	$\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$	$\begin{bmatrix} v_{00} & 0 \end{bmatrix}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$0 \rightarrow (i, l_i)$	$\lambda_i$	$[x_0 \ 0]$	$\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$	$\begin{bmatrix} v_{00} & 0 \end{bmatrix}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$0 \rightarrow (n, l_n)$	$\lambda_n$	$[x_0 \ 0]$	$\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$	$\begin{bmatrix} v_{00} & 0 \end{bmatrix}$
$(1, l_1) \rightarrow (1, l_1 - 1)$	$\mu$	$[x_0 \ x_1]$	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} v_{(1, l_1)0} & v_{(1, l_1)1} \end{bmatrix}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$(i, k) \rightarrow (i, k - 1)$	$\mu$	$[x_0 \ x_1]$	$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$	$\begin{bmatrix} v_{(i, k)0} & v_{(i, k)1} \end{bmatrix}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$(i, 1) \rightarrow 0$	$\mu$	$[x_0 \ 0]$	$\begin{bmatrix} 0 & * \\ 1 & * \end{bmatrix}$	$\begin{bmatrix} v_{(i, 1)1} & * \end{bmatrix}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$(i, k) \rightarrow (i, l_j)$	$\lambda_j$	$[x_0 \ 0]$	$\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$	$\begin{bmatrix} v_{(i, k)0} & 0 \end{bmatrix}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

$$\pi_0 + \sum_{i=1}^n \sum_{k=1}^{l_i} \pi_{(i, k)} = 1 \quad (46)$$

という連立方程式が立てられる。ただし、 $\sum_{i=1}^n \lambda_i = \lambda$ とおいた。まず、定常確率を求める。式(44)、式(46)より、

$$\pi_{(i, l_i)} = \frac{\lambda_i}{\mu + \lambda} \quad (47)$$

と書ける。式(45)より、

$$\pi_{(i,k)} = \frac{\mu}{\mu + \lambda} \pi_{(i,k+1)}, \quad 1 \leq k < l_i \quad (48)$$

と書ける. この漸化式を解くと,

$$\pi_{(i,k)} = \left( \frac{\mu}{\mu + \lambda} \right)^{l_i - k} \pi_{(i,l_i)} \quad (49)$$

$$= \left( \frac{\mu}{\mu + \lambda} \right)^{l_i - k} \frac{\lambda_i}{\mu + \lambda}, \quad 1 \leq k \leq l_i \quad (50)$$

を得る. さらに式 (43) より,

$$\pi_0 = \frac{\mu}{\lambda} \sum_{i=1}^n \pi_{(i,1)} \quad (51)$$

$$= \frac{\mu}{\lambda} \sum_{i=1}^n \left( \frac{\mu}{\mu + \lambda} \right)^{l_i - 1} \frac{\lambda_i}{\mu + \lambda} \quad (52)$$

$$= \frac{1}{\lambda} \sum_{i=1}^n \left( \frac{\mu}{\mu + \lambda} \right)^{l_i} \lambda_i \quad (53)$$

と書ける.

式 (42) より,  $1 \leq k < l_i$  に対して

$$v_{(i,k)0} = \frac{1}{\mu + \lambda} (\pi_{(i,k)} + \mu v_{(i,k+1)0}) \quad (54)$$

$$v_{(i,k)1} = \frac{1}{\mu + \lambda} (\pi_{(i,k)} + \mu v_{(i,k+1)1}) \quad (55)$$

と書ける. この漸化式を解くと,

$$v_{(i,k)0} = \sum_{m=k}^{l_i-1} \frac{\mu^{m-k} \pi_{(i,m)}}{(\mu + \lambda)^{m-k+1}} + \left( \frac{\mu}{\mu + \lambda} \right)^{l_i - k} v_{(i,l_i)0} \quad (56)$$

$$v_{(i,k)1} = \sum_{m=k}^{l_i-1} \frac{\mu^{m-k} \pi_{(i,m)}}{(\mu + \lambda)^{m-k+1}} + \left( \frac{\mu}{\mu + \lambda} \right)^{l_i - k} v_{(i,l_i)1} \quad (57)$$

と書ける. この導出は付録 B で述べる. 式 (40) より,

$$v_{00} = \frac{1}{\lambda} \left( \pi_0 + \sum_{i=1}^n \mu v_{(i,1)1} \right) \quad (58)$$

である.

式 (57) に  $k = 1$  を代入して  $v_{(i,1)1}$  を求めると,

$$v_{(i,1)1} = \sum_{m=1}^{l_i-1} \frac{\mu^{m-1} \pi_{(i,m)}}{(\mu + \lambda)^m} + \left( \frac{\mu}{\mu + \lambda} \right)^{l_i-1} v_{(i,l_i)1} \quad (59)$$

である. 式 (41) より,

$$v_{(i,l_i)1}(\mu + \lambda) = \pi_{(i,l_i)} \quad (60)$$

$$v_{(i,l_i)1} = \frac{\pi_{(i,l_i)}}{\mu + \lambda} \quad (61)$$

である. これを代入すると,

$$v_{(i,1)1} = \sum_{m=1}^{l_i-1} \frac{\mu^{m-1} \pi_{(i,m)}}{(\mu + \lambda)^m} + \left( \frac{\mu}{\mu + \lambda} \right)^{l_i-1} \frac{\pi_{(i,l_i)}}{\mu + \lambda} \quad (62)$$

である. さらに, これを式 (58) に代入すると

$$v_{00} = \frac{1}{\lambda} \left( \pi_0 + \sum_{i=1}^n \mu \left( \sum_{m=1}^{l_i-1} \frac{\mu^{m-1} \pi_{(i,m)}}{(\mu + \lambda)^m} + \left( \frac{\mu}{\mu + \lambda} \right)^{l_i-1} \frac{\pi_{(i,l_i)}}{\mu + \lambda} \right) \right) \quad (63)$$

$$= \frac{1}{\lambda} \left( \pi_0 + \sum_{i=1}^n \sum_{m=1}^{l_i-1} \frac{\mu^m \pi_{(i,m)}}{(\mu + \lambda)^m} + \sum_{i=1}^n \mu \left( \frac{\mu}{\mu + \lambda} \right)^{l_i-1} \frac{\pi_{(i,l_i)}}{\mu + \lambda} \right) \quad (64)$$

と書ける. ここで,  $\pi_{(i,m)} = \left( \frac{\mu}{\mu + \lambda} \right)^{l_i-m} \frac{\lambda_i}{\mu + \lambda}$  を代入して,

$$v_{00} = \frac{1}{\lambda} \left( \pi_0 + \sum_{i=1}^n \sum_{m=1}^{l_i-1} \frac{\mu^m}{(\mu + \lambda)^m} \cdot \left( \frac{\mu}{\mu + \lambda} \right)^{l_i-m} \frac{\lambda_i}{\mu + \lambda} + \sum_{i=1}^n \mu \left( \frac{\mu}{\mu + \lambda} \right)^{l_i-1} \cdot \frac{1}{\mu + \lambda} \cdot \frac{\lambda_i}{\mu + \lambda} \right) \quad (65)$$

$$= \frac{1}{\lambda} \left( \pi_0 + \sum_{i=1}^n \frac{\lambda_i}{\mu + \lambda} \sum_{m=1}^{l_i-1} \left( \frac{\mu}{\mu + \lambda} \right)^{l_i} + \sum_{i=1}^n \frac{\lambda_i}{\mu + \lambda} \left( \frac{\mu}{\mu + \lambda} \right)^{l_i} \right) \quad (66)$$

$$= \frac{1}{\lambda} \left( \pi_0 + \sum_{i=1}^n \frac{\lambda_i}{\mu + \lambda} \left( \sum_{m=1}^{l_i-1} \left( \frac{\mu}{\mu + \lambda} \right)^{l_i} + \left( \frac{\mu}{\mu + \lambda} \right)^{l_i} \right) \right) \quad (67)$$

$$= \frac{1}{\lambda} \left( \pi_0 + \sum_{i=1}^n \frac{\lambda_i}{\mu + \lambda} \cdot l_i \cdot \left( \frac{\mu}{\mu + \lambda} \right)^{l_i} \right) \quad (68)$$

である．式 (53) より  $\pi_0 = \frac{1}{\lambda} \sum_{i=1}^n \left(\frac{\mu}{\mu+\lambda}\right)^{l_i} \lambda_i$  を代入すると，

$$v_{00} = \frac{1}{\lambda} \left( \frac{1}{\lambda} \sum_{i=1}^n \left(\frac{\mu}{\mu+\lambda}\right)^{l_i} \lambda_i + \sum_{i=1}^n \frac{\lambda_i}{\mu+\lambda} l_i \left(\frac{\mu}{\mu+\lambda}\right)^{l_i} \right) \quad (69)$$

$$= \frac{1}{\lambda} \sum_{i=1}^n \lambda_i \left(\frac{\mu}{\mu+\lambda}\right)^{l_i} \left(\frac{1}{\lambda} + \frac{l_i}{\mu+\lambda}\right) \quad (70)$$

したがって，

$$v_{00} = \sum_{i=1}^n \frac{\lambda_i}{\lambda} \left(\frac{\mu}{\mu+\lambda}\right)^{l_i} \left(\frac{1}{\lambda} + \frac{l_i}{\mu+\lambda}\right) \quad (71)$$

である．ここで，求める AoI の式は，

$$\Delta = v_{00} + \sum_{i=1}^n \sum_{k=1}^{l_i} v_{(i,k)0} \quad (72)$$

である．式 (41) より，

$$v_{(i,l_i)0} = \frac{\pi_{(i,l_i)} + \lambda_i v_{00} + \lambda_i \sum_{j=1}^n \sum_{m=1}^{l_j} v_{(j,m)0}}{\mu + \lambda} \quad (73)$$

$$= \frac{\pi_{(i,l_i)} + \lambda_i \Delta}{\mu + \lambda} \quad (74)$$

と書ける．式 (56) を式 (72) に代入して，

$$\Delta = v_{00} + \sum_{i=1}^n \sum_{k=1}^{l_i} v_{(i,k)0} \quad (75)$$

$$= v_{00} + \sum_{i=1}^n \sum_{k=1}^{l_i} \left[ \sum_{m=k}^{l_i-1} \frac{\mu^{m-k} \pi_{(i,m)}}{(\mu+\lambda)^{m-k+1}} + \left(\frac{\mu}{\mu+\lambda}\right)^{l_i-k} v_{(i,l_i)0} \right] \quad (76)$$

$$= v_{00} + \sum_{i=1}^n \left[ \sum_{m=1}^{l_i-1} \pi_{(i,m)} \sum_{k=1}^m \frac{\mu^{m-k}}{(\mu+\lambda)^{m-k+1}} + v_{(i,l_i)0} \sum_{k=1}^{l_i} \left(\frac{\mu}{\mu+\lambda}\right)^{l_i-k} \right] \quad (77)$$

と書ける．等比数列の和の公式より，

$$\sum_{k=1}^m \frac{\mu^{m-k}}{(\mu+\lambda)^{m-k+1}} = \frac{1}{\mu+\lambda} \sum_{r=0}^{m-1} \left(\frac{\mu}{\mu+\lambda}\right)^r \quad (78)$$

$$= \frac{1}{\lambda} \left[ 1 - \left(\frac{\mu}{\mu+\lambda}\right)^m \right] \quad (79)$$

$$\sum_{k=1}^{l_i} \left( \frac{\mu}{\mu + \lambda} \right)^{l_i - k} = \sum_{r=0}^{l_i - 1} \left( \frac{\mu}{\mu + \lambda} \right)^r \quad (80)$$

$$= \frac{\mu + \lambda}{\lambda} \left[ 1 - \left( \frac{\mu}{\mu + \lambda} \right)^{l_i} \right] \quad (81)$$

である。したがって、

$$\Delta = v_{00} + \sum_{i=1}^n \left[ \frac{1}{\lambda} \sum_{m=1}^{l_i - 1} \pi_{(i,m)} \left( 1 - \left( \frac{\mu}{\mu + \lambda} \right)^m \right) + v_{(i,l_i)0} \cdot \frac{\mu + \lambda}{\lambda} \left( 1 - \left( \frac{\mu}{\mu + \lambda} \right)^{l_i} \right) \right] \quad (82)$$

$$= v_{00} + \frac{1}{\lambda} \sum_{i=1}^n \sum_{m=1}^{l_i - 1} \pi_{(i,m)} \left( 1 - \left( \frac{\mu}{\mu + \lambda} \right)^m \right) + \frac{\mu + \lambda}{\lambda} \sum_{i=1}^n v_{(i,l_i)0} \left( 1 - \left( \frac{\mu}{\mu + \lambda} \right)^{l_i} \right) \quad (83)$$

となる。ここに式 (74) を代入すると、

$$\begin{aligned} \Delta &= v_{00} + \frac{1}{\lambda} \sum_{i=1}^n \sum_{m=1}^{l_i - 1} \pi_{(i,m)} \left( 1 - \left( \frac{\mu}{\mu + \lambda} \right)^m \right) \\ &\quad + \frac{\mu + \lambda}{\lambda} \sum_{i=1}^n \frac{\pi_{(i,l_i)} + \lambda_i \Delta}{\mu + \lambda} \left( 1 - \left( \frac{\mu}{\mu + \lambda} \right)^{l_i} \right) \end{aligned} \quad (84)$$

$$\begin{aligned} &= v_{00} + \frac{1}{\lambda} \sum_{i=1}^n \sum_{m=1}^{l_i - 1} \pi_{(i,m)} \left( 1 - \left( \frac{\mu}{\mu + \lambda} \right)^m \right) \\ &\quad + \frac{1}{\lambda} \sum_{i=1}^n (\pi_{(i,l_i)} + \lambda_i \Delta) \left( 1 - \left( \frac{\mu}{\mu + \lambda} \right)^{l_i} \right) \end{aligned} \quad (85)$$

と書ける。 $\Delta$  について整理すると

$$\Delta \left[ 1 - \frac{1}{\lambda} \sum_{i=1}^n \lambda_i \left( 1 - \left( \frac{\mu}{\mu + \lambda} \right)^{l_i} \right) \right] \quad (86)$$

$$= v_{00} + \frac{1}{\lambda} \sum_{i=1}^n \left[ \sum_{m=1}^{l_i - 1} \pi_{(i,m)} \left( 1 - \left( \frac{\mu}{\mu + \lambda} \right)^m \right) \right] + \frac{1}{\lambda} \sum_{i=1}^n \pi_{(i,l_i)} \left( 1 - \left( \frac{\mu}{\mu + \lambda} \right)^{l_i} \right) \quad (87)$$

$$= v_{00} + \frac{1}{\lambda} \sum_{i=1}^n \sum_{m=1}^{l_i} \pi_{(i,m)} \left( 1 - \left( \frac{\mu}{\mu + \lambda} \right)^m \right) \quad (88)$$

である。また、式 (50) より  $\pi_{(i,m)} = \left(\frac{\mu}{\mu+\lambda}\right)^{l_i-m} \frac{\lambda_i}{\mu+\lambda}$  を代入して計算すると、

$$\begin{aligned}
& \Delta \left[ 1 - \frac{1}{\lambda} \sum_{i=1}^n \lambda_i \left( 1 - \left( \frac{\mu}{\mu+\lambda} \right)^{l_i} \right) \right] \\
&= v_{00} + \frac{1}{\lambda} \sum_{i=1}^n \sum_{m=1}^{l_i} \left( \frac{\mu}{\mu+\lambda} \right)^{l_i-m} \frac{\lambda_i}{\mu+\lambda} \left( 1 - \left( \frac{\mu}{\mu+\lambda} \right)^m \right) \\
&= v_{00} + \frac{1}{\lambda} \sum_{i=1}^n \left( \frac{\lambda_i}{\mu+\lambda} \sum_{m=1}^{l_i} \left( \frac{\mu}{\mu+\lambda} \right)^{l_i-m} - \frac{\lambda_i}{\mu+\lambda} \sum_{m=1}^{l_i} \left( \frac{\mu}{\mu+\lambda} \right)^{l_i} \right) \\
&= v_{00} + \frac{1}{\lambda} \sum_{i=1}^n \left( \frac{\lambda_i}{\lambda} \left( 1 - \left( \frac{\mu}{\mu+\lambda} \right)^{l_i} \right) - \frac{\lambda_i l_i}{\mu+\lambda} \left( \frac{\mu}{\mu+\lambda} \right)^{l_i} \right) \tag{89}
\end{aligned}$$

である。ここで左辺に注目して、 $\sum_{i=1}^n \lambda_i = \lambda$  より、

$$1 - \frac{1}{\lambda} \sum_{i=1}^n \lambda_i \left( 1 - \left( \frac{\mu}{\mu+\lambda} \right)^{l_i} \right) = 1 - \frac{1}{\lambda} \sum_{i=1}^n \lambda_i + \frac{1}{\lambda} \sum_{i=1}^n \lambda_i \left( \frac{\mu}{\mu+\lambda} \right)^{l_i} \tag{90}$$

$$= \frac{1}{\lambda} \sum_{i=1}^n \lambda_i \left( \frac{\mu}{\mu+\lambda} \right)^{l_i} \tag{91}$$

である。式 (71) を式 (89) に代入して、

$$\begin{aligned}
\Delta \cdot \frac{1}{\lambda} \sum_{i=1}^n \lambda_i \left( \frac{\mu}{\mu+\lambda} \right)^{l_i} &= \sum_{i=1}^n \frac{\lambda_i}{\lambda} \left( \frac{\mu}{\mu+\lambda} \right)^{l_i} \left( \frac{1}{\lambda} + \frac{l_i}{\mu+\lambda} \right) \\
&\quad + \frac{1}{\lambda} \sum_{i=1}^n \left( \frac{\lambda_i}{\lambda} \left( 1 - \left( \frac{\mu}{\mu+\lambda} \right)^{l_i} \right) - \frac{\lambda_i l_i}{\mu+\lambda} \left( \frac{\mu}{\mu+\lambda} \right)^{l_i} \right) \tag{92}
\end{aligned}$$

$$= \sum_{i=1}^n \frac{\lambda_i}{\lambda^2} \left( \frac{\mu}{\mu+\lambda} \right)^{l_i} + \sum_{i=1}^n \frac{\lambda_i}{\lambda^2} \left( 1 - \left( \frac{\mu}{\mu+\lambda} \right)^{l_i} \right) \tag{93}$$

$$= \frac{1}{\lambda^2} \sum_{i=1}^n \lambda_i \tag{94}$$

$$= \frac{1}{\lambda} \tag{95}$$

以上より、

$$\Delta = \frac{1}{\sum_{i=1}^n \lambda_i \left( \frac{\mu}{\mu+\lambda} \right)^{l_i}} \tag{96}$$

と求まる。

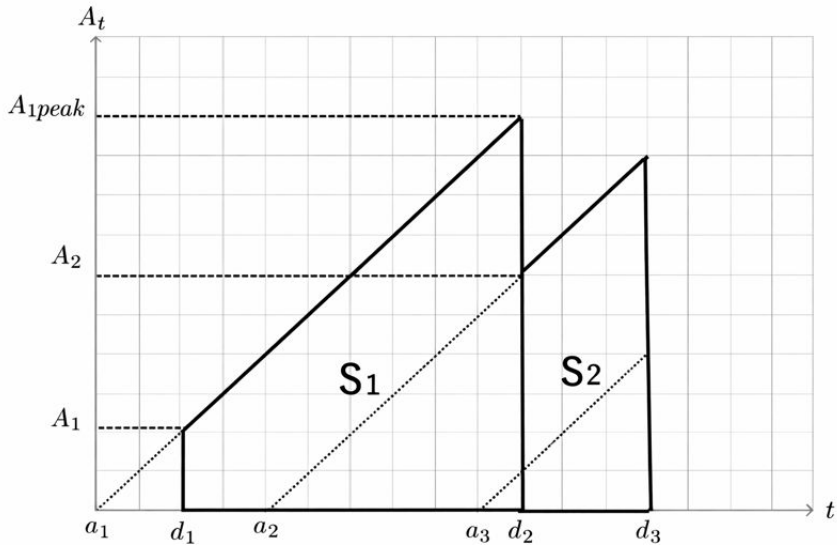


図 8 時間平均 AoI の導出

## 5 シミュレーションによる検証

4章で導出した AoI の式から求まる値と実験的に求めた値が一致することを確認するために図 6 の通信システムをシミュレートした。サービスレート  $\mu$  を一定値 1 とし、 $n$  種類のシンボルがそれぞれ  $\lambda_i$  でポアソン到着し、各シンボルはあらかじめ設定された自然数  $l_i$  個の packets に分割されるように設定した。サーバは packets を 1 つずつ処理し、そのサービス時間は平均  $\frac{1}{\mu}$  の指数分布に従う。シンボルの全 packets の処理完了をもってモニタの表示が更新され、処理中に新しいシンボルの最初の packets が到着した場合は、処理中のシンボルの残り packets をすべて棄却して新しいシンボルの処理を即座に開始する。サーバにシンボルが  $10^7$  個到着した時点で終了し、時間平均 AoI を計算する。これを 10 回繰り返し平均を求めた。

AoI の時間平均は式 (3) で定義される。図 8 のように台形の面積  $S_k$  を考えると、式 (3) の積分は台形の面積の和といえる。モニタに更新されるたびに台形の面積  $S_k$  を計算する。AoI の時間平均  $A_{ave}$  は、経過時間  $T$  の間にモニタが  $m$  回更新されたとすると、

$$A_{ave} = \frac{1}{T} \sum_{k=1}^m S_k \quad (97)$$

と式 (3) を近似して書ける。

表 4 理論値とシミュレーション結果の比較

ケース番号	$n$	$(l_1, l_2, \dots, l_n)$ $(\lambda_1, \lambda_2, \dots, \lambda_n)$	理論値	シミュレーション値	比 (シミュレーション/理論値)
1	3	(1, 2, 2) (0.1, 0.4, 0.5)	3.636364	3.636375	1.000003
2	4	(1, 2, 2, 3) (0.1, 0.2, 0.3, 0.4)	4.444444	4.444872	1.000096
3	7	(2, 3, 4, 5, 5, 6, 7) (0.05, 0.08, 0.11, 0.14, 0.17, 0.20, 0.25)	22.654867	22.65284	0.999911
4	7	(2, 3, 4, 5, 5, 6, 7) (0.25, 0.20, 0.05, 0.10, 0.15, 0.10, 0.15)	9.884170	9.8800526	0.999584
5	10	(1, 2, 3, 3, 5, 5, 7, 8, 8, 9) (0.05, 0.07, 0.02, 0.10, 0.18, 0.08, 0.12, 0.10, 0.10, 0.18)	14.772072	14.777163	1.000345
6	10	(2, 2, 3, 5, 8, 9, 11, 11, 13, 14) (0.05, 0.07, 0.02, 0.10, 0.18, 0.08, 0.12, 0.10, 0.10, 0.18)	27.311219	27.313408	1.000080

$\mu = 1$  として、パラメータを変化させたときの理論値とシミュレーションから得た AoI を表 4 に示す. 表 4 より, シミュレーションの AoI の値と AoI の理論値で比をとると, ほとんど 1 である. 相対誤差は 0.03 % 以内とわかる. 即ち, 導出した式とシミュレーションが一致することを確認できた. 誤差が生じるのはシミュレーションで求める時間平均 AoI が, 式 (3) を近似したことや, シンボルが発生を乱数としていることが考えられる. また, 割り当てるパケットの長さによって AoI の値が変わることもわかる.

## 6 まとめ

本研究では, 先行研究 [4] の結果から, ラグランジュの未定乗数法を用いて平均サービス時間の制約付き最適化を行った. また, 割り込みのある待ち行列通信システムで  $n$  種類の情報源をパケット数による符号化したときの時間平均 AoI を複雑な状態遷移を基本構造を見つけることで導出した.

今後の課題として, 有限なバッファ付きの通信モデルでの AoI を求めることや, AoI の視点から見た最適な符号化の方法を見つけることが考えられる.

## 謝辞

本研究に際して, 様々なご指導をいただいた指導教員の西新幹彦教授, また助言をいただいた研究室の先輩方に深く感謝申し上げます.

## 参考文献

- [1] 井上文彰, 滝根哲哉, 「Age of Information (AoI) 基本概念と研究動向」, <http://www.ieice.org/ess/sita/forum/article/2018/201810111910.pdf>, 2025 年 6 月閲覧
- [2] Y. Inoue, H. Masuyama, T. Takine and T. Tanaka, “A general formula for the station

- ary distribution of the age of information and its application to single-server queues,”  
IEEE Trans. Inf. Theory, vol.65, no.12, pp.8305–8324, Dec. 2019 Nov. 1979.
- [3] 細海俊介, 「情報鮮度の視点から見たサーバで割り込みのある待ち行列通信システムの解析」, 信州大学大学院工学専攻電子情報システム工学分野修士論文 (指導教員: 西新幹彦), 2024 年 3 月.
- [4] 野田哲平, 「サーバで割り込みのある通信システムで多値情報源を符号化したときの情報鮮度の数学的解析」, 信州大学工学部学士論文 (指導教員: 西新幹彦), 2025 年 4 月.
- [5] R. D. Yates and S. K. Kaul, “The Age of Information: Real-Time Status Updating by Multiple Sources,” IEEE Trans. Inf. Theory, vol. 65, no. 3, pp. 1807–1827, Mar. 2019.

## 付録 A 漸化式の解の証明

4章の式(56)と式(57)の漸化式の解き方と、解の証明をする。状態 $(i, k)$ は「シンボル $i$ 処理中、残り $k$ パケット」を表す $(k = l_i, l_i - 1, \dots, 1)$ 。  $1 \leq k < l_i$  について、

$$v_{(i,k)0}(\mu + \lambda) = \pi_{(i,k)} + \mu v_{(i,k+1)0} \quad (98)$$

$$v_{(i,k)1}(\mu + \lambda) = \pi_{(i,k)} + \mu v_{(i,k+1)1} \quad (99)$$

の漸化式が成り立つ。

漸化式を繰り返し適用すると、

$$\begin{aligned} v_{(i,l_i-1)0} &= \frac{\pi_{(i,l_i-1)}}{\mu + \lambda} + \frac{\mu}{\mu + \lambda} v_{(i,l_i)0} \\ v_{(i,l_i-2)0} &= \frac{\pi_{(i,l_i-2)}}{\mu + \lambda} + \frac{\mu}{\mu + \lambda} v_{(i,l_i-1)0} \\ &= \frac{\pi_{(i,l_i-2)}}{\mu + \lambda} + \frac{\mu}{\mu + \lambda} \left( \frac{\pi_{(i,l_i-1)}}{\mu + \lambda} + \frac{\mu}{\mu + \lambda} v_{(i,l_i)0} \right) \\ &= \frac{\pi_{(i,l_i-2)}}{\mu + \lambda} + \frac{\mu \pi_{(i,l_i-1)}}{(\mu + \lambda)^2} + \left( \frac{\mu}{\mu + \lambda} \right)^2 v_{(i,l_i)0} \end{aligned}$$

と書ける。このパターンから、一般に  $1 \leq k \leq l_i$  について、

$$v_{(i,k)0} = \sum_{m=k}^{l_i-1} \frac{\mu^{m-k} \pi_{(i,m)}}{(\mu + \lambda)^{m-k+1}} + \left( \frac{\mu}{\mu + \lambda} \right)^{l_i-k} v_{(i,l_i)0}$$

と予測される。同様に、

$$v_{(i,k)1} = \sum_{m=k}^{l_i-1} \frac{\mu^{m-k} \pi_{(i,m)}}{(\mu + \lambda)^{m-k+1}} + \left( \frac{\mu}{\mu + \lambda} \right)^{l_i-k} v_{(i,l_i)1}$$

とかける。  $1 \leq k < l_i$  について、以下の式

$$v_{(i,k)0} = \sum_{m=k}^{l_i-1} \frac{\mu^{m-k} \pi_{(i,m)}}{(\mu + \lambda)^{m-k+1}} + \left( \frac{\mu}{\mu + \lambda} \right)^{l_i-k} v_{(i,l_i)0} \quad (100)$$

$$v_{(i,k)1} = \sum_{m=k}^{l_i-1} \frac{\mu^{m-k} \pi_{(i,m)}}{(\mu + \lambda)^{m-k+1}} + \left( \frac{\mu}{\mu + \lambda} \right)^{l_i-k} v_{(i,l_i)1} \quad (101)$$

が成り立つことを証明する。  $k = l_i - 1$  のとき、

$$\begin{aligned}
v_{(i,l_i-1)0} &= \frac{1}{\mu + \lambda} (\pi_{(i,l_i-1)} + \mu v_{(i,l_i)0}) \\
&= \frac{\pi_{(i,l_i-1)}}{\mu + \lambda} + \frac{\mu}{\mu + \lambda} v_{(i,l_i)0} \\
&= \sum_{m=l_i-1}^{l_i-1} \frac{\mu^{m-(l_i-1)} \pi_{(i,m)}}{(\mu + \lambda)^{m-(l_i-1)+1}} + \left( \frac{\mu}{\mu + \lambda} \right)^{l_i-(l_i-1)} v_{(i,l_i)0} \\
&= \sum_{m=l_i-1}^{l_i-1} \frac{\mu^{m-l_i+1} \pi_{(i,m)}}{(\mu + \lambda)^{m-l_i+2}} + \left( \frac{\mu}{\mu + \lambda} \right)^1 v_{(i,l_i)0}
\end{aligned}$$

である。よって式 (100) は  $k = l_i - 1$  で成立。ある  $k$  ( $2 \leq k \leq l_i - 1$ ) で式 (100) が成立すると仮定する。  $k - 1$  について、

$$\begin{aligned}
v_{(i,k-1)0} &= \frac{1}{\mu + \lambda} (\pi_{(i,k-1)} + \mu v_{(i,k)0}) \\
&= \frac{\pi_{(i,k-1)}}{\mu + \lambda} + \frac{\mu}{\mu + \lambda} v_{(i,k)0} \\
&= \frac{\pi_{(i,k-1)}}{\mu + \lambda} + \frac{\mu}{\mu + \lambda} \left[ \sum_{m=k}^{l_i-1} \frac{\mu^{m-k} \pi_{(i,m)}}{(\mu + \lambda)^{m-k+1}} + \left( \frac{\mu}{\mu + \lambda} \right)^{l_i-k} v_{(i,l_i)0} \right] \\
&= \frac{\pi_{(i,k-1)}}{\mu + \lambda} + \sum_{m=k}^{l_i-1} \frac{\mu^{m-(k-1)} \pi_{(i,m)}}{(\mu + \lambda)^{m-(k-1)+1}} + \left( \frac{\mu}{\mu + \lambda} \right)^{l_i-(k-1)} v_{(i,l_i)0} \\
&= \sum_{m=k-1}^{l_i-1} \frac{\mu^{m-(k-1)} \pi_{(i,m)}}{(\mu + \lambda)^{m-(k-1)+1}} + \left( \frac{\mu}{\mu + \lambda} \right)^{l_i-(k-1)} v_{(i,l_i)0}
\end{aligned}$$

である。これは式 (100) の  $k - 1$  の場合である。

以上より数学的帰納法より式 (100) は成り立つ。ここで、  $k = l_i$  の場合を確認する。このとき、

$$\begin{aligned}
\sum_{m=l_i}^{l_i-1} \frac{\mu^{m-l_i} \pi_{(i,m)}}{(\mu + \lambda)^{m-l_i+1}} &= 0 \\
\left( \frac{\mu}{\mu + \lambda} \right)^{l_i-l_i} v_{(i,l_i)0} &= \left( \frac{\mu}{\mu + \lambda} \right)^0 v_{(i,l_i)0} = v_{(i,l_i)0}
\end{aligned}$$

と書ける。したがって式 (100) の右辺は  $v_{(i,l_i)0}$  となり、左辺と一致する。よって  $k = l_i$  で成立する。また、  $v_{(i,k)1}$  についても全く同様の手順で証明できる。

## 付録 B ソースコード

5章でシミュレーションを行った。情報源を符号化した時のモニタ AoI を計算するプログラムを以下に記載する。

```

#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <time.h>
#include "MT.h"

#define Kaisu 10000000
#define SYMBOL_COUNT 3
#define SIMULATION_COUNT 20

double arrive_rates[SYMBOL_COUNT] = {0.1, 0.4, 0.5 }; //  $\lambda_1=0.1, \lambda_2=0.4, \lambda_3=0.5$ 
int packet_counts[SYMBOL_COUNT] = {1, 2, 2 }; //  $l_1=1, l_2=2, l_3=2$ 
double service_rate = 1.0; //  $\mu=1.0$ 

// 指数分布乱数
double Random_Exp(double Lambda) {
    return -log(genrand_real3()) / Lambda;
}

// シンボル選択
int symboli() {
    double rnd = genrand_real1();
    double total_arrive_rate = 0.0;

    for (int i = 0; i < SYMBOL_COUNT; i++) {
        total_arrive_rate += arrive_rates[i];
    }

    double cum_prob = 0.0;
    for (int i = 0; i < SYMBOL_COUNT; i++) {
        cum_prob += arrive_rates[i] / total_arrive_rate;
        if (rnd < cum_prob) {
            return i;
        }
    }
    return SYMBOL_COUNT - 1;
}

// サービス時間
double ui() {
    return Random_Exp(service_rate);
}

// 理論値計算
double theoretical_aoi() {
    double Lambda = 0.0;
    for (int i = 0; i < SYMBOL_COUNT; i++) {
        Lambda += arrive_rates[i];
    }

    double denominator = 0.0;
    for (int i = 0; i < SYMBOL_COUNT; i++) {
        double term = (arrive_rates[i] * service_rate / (service_rate + Lambda))
            * pow(service_rate / (service_rate + Lambda), packet_counts[i] - 1);
        denominator += term;
        printf("シンボル %d の寄与: %f\n", i, term);
    }

    printf("Λ = %f, 分母 = %f\n", Lambda, denominator);
    return 1.0 / denominator;
}

// シミュレーション
double simulate_AoI() {
    double current_time = 0.0;
    double service_end_time = INFINITY;
    double server_arrival_time = 0.0;
    double monitor_update_time = 0.0;
    double monitor_arrival_time = 0.0;

```

```

double cumulative_AoI = 0.0;
double first_update_time = 0.0;

int current_symbol = -1;
int remaining_packets = 0;
int arrival_count = 0;
int completion_count = 0;

double total_lambda = 0.0;
for (int i = 0; i < SYMBOL_COUNT; i++) {
    total_lambda += arrive_rates[i];
}

// 初期到着
current_symbol = symboli();
remaining_packets = packet_counts[current_symbol];
double next_arrival = Random_Exp(total_lambda);
service_end_time = current_time + ui();
server_arrival_time = current_time;
arrival_count++;

while (arrival_count < Kaisu && completion_count < Kaisu) {
    if (next_arrival <= service_end_time) {
        // 到着イベント (割り込み)
        current_time = next_arrival;
        current_symbol = symboli();
        remaining_packets = packet_counts[current_symbol];
        next_arrival = current_time + Random_Exp(total_lambda);
        service_end_time = current_time + ui();
        server_arrival_time = current_time;
        arrival_count++;
    } else {
        // サービス完了
        current_time = service_end_time;

        if (remaining_packets > 1) {
            // 中間パケット
            remaining_packets--;
            service_end_time = current_time + ui();
        } else {
            // 最終パケット - モニタ更新
            if (completion_count == 0) {
                first_update_time = current_time;
            } else {
                double a = current_time - monitor_arrival_time;
                double b = monitor_update_time - monitor_arrival_time;
                cumulative_AoI += 0.5 * (a*a - b*b);
            }

            monitor_update_time = current_time;
            monitor_arrival_time = server_arrival_time;
            current_symbol = -1;
            service_end_time = INFINITY;
            completion_count++;
        }
    }
}

return cumulative_AoI / (monitor_update_time - first_update_time);
}

int main() {
    init_genrand((unsigned)time(NULL));

    // パラメータ表示
    printf("=== パラメータ設定 ===\n");
    double Lambda = 0.0;
    for (int i = 0; i < SYMBOL_COUNT; i++) {
        printf("シンボル %d: λ=%f, l=%d\n", i, arrive_rates[i], packet_counts[i]);
    }
}

```

```

    Lambda += arrive_rates[i];
}
printf("μ=%f, λ=%f\n\n", service_rate, Lambda);

// 理論値計算
double theory = theoretical_aoi();
printf("\n理論値: Δ = %f\n", theory);

// シミュレーション
printf("\n=== シミュレーション実行 ===\n");
double total_AoI = 0.0;
for (int sim = 0; sim < SIMULATION_COUNT; sim++) {
    double AoI = simulate_AoI();
    total_AoI += AoI;
    printf("シミュレーション %2d: Δ = %f\n", sim+1, AoI);
}

double average_AoI = total_AoI / SIMULATION_COUNT;
printf("\n平均シミュレーション値: %f\n", average_AoI);
printf("理論値との差: %f\n", fabs(average_AoI - theory));
printf("相対誤差: %.2f%\n", fabs(average_AoI - theory) / theory * 100);

return 0;
}

```