

信州大学工学部

学士論文

消失通信路の有限長解析

指導教員 西新 幹彦 准教授

学科 電気電子工学科

学籍番号 12T2093A

氏名 山川 誠史

2016 年 2 月 19 日

目次

1	はじめに	1
2	通信路符号化と有限長解析	1
3	消失通信路の有限長解析	2
3.1	復号誤り確率の上界	3
3.2	消失通信路における Achievability Bound	5
3.3	消失通信路における Converse Bound	6
3.4	有限長で達成可能なレート	8
4	まとめ	28
	謝辞	28
	参考文献	28
付録 A	レートの測定	30
A.1	DT Bound からレートを測るプログラム	30
A.2	Converse Bound からレートを測るプログラム	31

1 はじめに

情報を送って受け取る行為が盛んに行われる情報社会において一回の通信で送信できる情報の限界には興味もたれる。通信路容量と呼ばれる送信可能なレートの上限は符号語長を無限とみなした上で導出されているが、実際には符号語長は有限であるため送信可能なレートを符号語長が有限であることを考慮したうえで評価することに近年関心もたれている。これは有限長解析と呼ばれている。Yury Polyanskiy, H. Vincent Poor, and Sergio Verdú[1] は様々な通信路に対して有限長解析を行い、送信可能なレートの上下界を導出している。特に消失通信路に関しては具体的に二元消失通信路の有限長解析の数値結果を示している。本研究では様々なアルファベットサイズの消失通信路について有限長解析をする。

2 通信路符号化と有限長解析

情報通信の最も抽象的なモデルでは、送信者が出力したメッセージが符号器によって符号語に変換されて通信路に入力される。一般に通信路の出力はその入力に依存して確率的に決まる。これは、通信路の出力は一般にその入力と異なることを表しており、その意味で通信路の出力は入力にノイズが加えられたものであると解釈される。復号器は通信路の出力を受け取り、通信路に入力された符号語を推定する。推定された符号語は受信者に渡される。推定された符号語と真の符号語が異なることを復号誤りと言う。符号器と復号器は復号誤り確率を小さくするためと符号化を効率的に行うために設置される。その概略図を図1に示す。通信路の使用1回あたりの伝送情報量は符号化レートと呼ばれ、符号語の数を M 、符号語長を n とした時 $\frac{1}{n} \log M$ と表される。

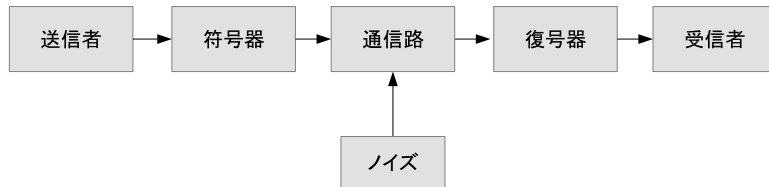


図1 通信路

通信路の入力アルファベットと出力アルファベットをそれぞれ $\mathcal{A}$, $\mathcal{B}$ とする。通信路にシンボル $x \in \mathcal{A}$ を入力したときに出力が $y \in \mathcal{B}$ となる確率は条件付き確率 $P_{Y|X}(y|x) \triangleq \Pr\{Y = y|X = x\}$ で表すことができる。確率変数 X, Y はそれぞれ通信路の入力と出力を表す。もち

るん任意の $x \in \mathcal{A}$ に対して

$$\sum_{y \in \mathcal{B}} P_{Y|X}(y | x) = 1 \quad (1)$$

が成り立つ．通信路に長さ n の文字列 $\mathbf{x} = (x_1, x_2, \dots, x_n) \in \mathcal{A}^n$ を入力したとき，出力として $\mathbf{y} = (y_1, y_2, \dots, y_n) \in \mathcal{B}^n$ が得られる確率が

$$P_{\mathbf{Y}|\mathbf{X}}(\mathbf{y} | \mathbf{x}) = \prod_{i=1}^n P_{Y_i|X_i}(y_i | x_i) \quad (2)$$

と表されるとき，この通信路を定常無記憶通信路という．すなわち定常無記憶通信路では，文字単位の入出力のペアたちが互いに独立である．本研究では定常無記憶通信路を考える．

通信路を n 回使用する符号を次のように表す．メッセージの数を M_n とおく．一般性を損なうことなくメッセージ全体を $\{1, 2, \dots, M_n\}$ と表す．符号器は各メッセージ i を長さ n の文字列 $\mathbf{c}_i \in \mathcal{A}^n$ に変換する． $\mathbf{c}_i$ を i の符号語という．通信路の出力として受け取られる長さ n の文字列に対してもとの符号語を推定する復号器を考え，その復号誤り確率を ϵ_n とおく．

上記の準備のもと，通常の通信路符号化の問題は次のように定式化される．

定義 1

$$\text{レート } R \text{ が達成可能} \iff \lim_{n \rightarrow \infty} \epsilon_n = 0 \text{ かつ } \liminf_{n \rightarrow \infty} \frac{1}{n} \log M_n \\ \text{となる符号の列が存在する.}$$

定義 2 与えられた通信路の通信路容量 C を

$$C \triangleq \sup\{R \mid R \text{ が達成可能}\}$$

と定める．

定義より，通信路容量より小さいレートならば，十分大きい n に対して任意に小さい復号誤りで通信が可能であることを意味している．しかし実際に符号器を実装する場合，符号語長 n を無制限に大きくすることはできない．そのため実際に送信可能なレートの上限は通信路容量よりも下回ることになる．そこで通常の通信路容量に代わり，符号語長 n を固定したときの符号化レートと復号誤り確率の関係を調べる研究が生まれた．これが本論文の主題である有限長解析である．

3 消失通信路の有限長解析

3.4 節では消失通信路の送信可能なレートの上下界を導出するが，その準備としてまず 3.2 節と 3.3 節で消失通信路の復号誤り確率の上下界を求める．復号誤り確率の上界を

Achievability Bound, 下界を Converse Bound と呼ぶ。また, 消失通信路の Achievable Bound を求める準備として 3.1 節で復号誤り確率の基本的な上界を求める。

3.1 復号誤り確率の上界

この節では復号誤り確率の上界を文献 [1] に基づき導出する。準備として次の補題を用意する。

補題 1 $z \geq 0, \gamma > 0$ に関して,

$$e^{-[\log \frac{z}{\gamma}]^+} = \mathbf{1}\{z \leq \gamma\} + \frac{\gamma}{z} \mathbf{1}\{z > \gamma\} \quad (3)$$

が成り立つ。

(証明) $e^{-[\log \frac{z}{\gamma}]^+}$ を計算する。ここで $[\]^+$ とは, $a > 0$ ならば $[a]^+ = a$ となり $a \leq 0$ ならば $[a]^+ = 0$ となる関数である。 $z = 0, \gamma > 0$ の時 e の指数部分が 0 となるため $e^{-[\log \frac{z}{\gamma}]^+} = 1$ となり, $0 < z \leq \gamma$ の時も同様に e の指数部分が 0 となるため $e^{-[\log \frac{z}{\gamma}]^+} = 1$ となる。また, $0 < \gamma < z$ の時は e の指数部分は $-\log \frac{z}{\gamma}$ となるため $e^{-[\log \frac{z}{\gamma}]^+} = \frac{\gamma}{z}$ となる。よって任意の $z \geq 0, \gamma > 0$ で式 (3) が成り立つ。□

次の定理は符号語長が 1 の場合の復号誤り確率に対する基本的な上界である。後にアルファベットを拡大して任意の符号語長に関する結果として適用される。また, 次の定理で使われている $i(x; y)$ とは入力と出力の分布を $P_X(x), P_Y(y)$, 同時分布を $P_{XY}(x, y)$ とした時に

$$i(x; y) \triangleq \log \frac{P_{XY}(x, y)}{P_X(x)P_Y(y)} \quad (4)$$

と定義したものである。

定理 1 任意の分布 P_X に対して, 符号語の数 M と平均誤り率 ϵ が

$$\epsilon \leq \mathbb{E}[e^{-[i(X; Y) - \log \frac{M-1}{2}]^+}] \quad (5)$$

を満たす符号が存在する。

(証明) 補題 1 で $z = \frac{P_{XY}(X, Y)}{P_X(X)P_Y(Y)}$ と置くと式 (3) の左辺の e の指数部分は

$$-\left[\log \frac{P_{XY}(X, Y)}{P_X(X)P_Y(Y)} - \log \gamma\right]^+ = -[i(X; Y) - \log \gamma]^+$$

となり, 両辺の期待値を計算すると

$$\mathbb{E}[e^{-[i(X; Y) - \log \gamma]^+}] = \mathbb{E}\left[\mathbf{1}\left\{\frac{P_{XY}(X, Y)}{P_X(X)P_Y(Y)} \leq \gamma\right\}\right] + \mathbb{E}\left[\frac{\gamma}{z} \mathbf{1}\left\{\frac{P_{XY}(X, Y)}{P_X(X)P_Y(Y)} > \gamma\right\}\right] \quad (6)$$

となる．式 (6) の右边を第 1 項と第 2 項に分けて変形すると，第 1 項は

$$\mathbb{E} \left[\mathbf{1} \left\{ \frac{P_{XY}(X, Y)}{P_X(X)P_Y(Y)} \leq \gamma \right\} \right] = \Pr \left\{ \frac{P_{XY}(X, Y)}{P_X(X)P_Y(Y)} \leq \gamma \right\} \quad (7)$$

となる．第 2 項を変形すると

$$\mathbb{E} \left[\frac{\gamma}{z} \mathbf{1} \left\{ \frac{P_{XY}(X, Y)}{P_X(X)P_Y(Y)} > \gamma \right\} \right] \quad (8)$$

$$= \mathbb{E} \left[\gamma \frac{P_X(X)P_Y(Y)}{P_{XY}(X, Y)} \mathbf{1} \left\{ \frac{P_{XY}(X, Y)}{P_X(X)P_Y(Y)} > \gamma \right\} \right] \quad (9)$$

$$= \sum_{x, y} \gamma \frac{P_X(x)P_Y(y)}{P_{XY}(x, y)} \mathbf{1} \left\{ \frac{P_{XY}(X, Y)}{P_X(X)P_Y(Y)} > \gamma \right\} P_{XY}(x, y) \quad (10)$$

$$= \gamma \sum_{x, y} \mathbf{1} \left\{ \frac{P_{XY}(X, Y)}{P_X(X)P_Y(Y)} > \gamma \right\} P_X(x)P_Y(y) \quad (11)$$

となる．ここで Y と同じ分布で X とは独立な確率変数 $\bar{Y}$ を導入すると， $P_X(x)P_Y(y) = P_{X\bar{Y}}(x, y)$ と書ける．よって式 (11) は

$$\gamma \sum_{x, y} \mathbf{1} \left\{ \frac{P_{XY}(X, Y)}{P_X(X)P_Y(Y)} > \gamma \right\} P_{X\bar{Y}}(x, y) = \gamma \mathbb{E} \left[\mathbf{1} \left\{ \frac{P_{XY}(X, \bar{Y})}{P_X(X)P_Y(\bar{Y})} > \gamma \right\} \right] \quad (12)$$

$$= \gamma \Pr \left\{ \frac{P_{XY}(X, \bar{Y})}{P_X(X)P_Y(\bar{Y})} > \gamma \right\} \quad (13)$$

と書ける．式 (7) と式 (13) を式 (6) に代入して $\gamma = \frac{M-1}{2}$ とすると，式 (6) は

$$\mathbb{E} \left[e^{-[i(X; Y) - \log \gamma]^+} \right] = \Pr \left\{ i(X; Y) \leq \log \frac{M-1}{2} \right\} + \frac{M-1}{2} \Pr \left\{ i(X; \bar{Y}) > \log \frac{M-1}{2} \right\} \quad (14)$$

と変形できる．

分布 P_X に従い $\mathcal{A}$ から 1 つ選んだものを符号語長 1 の符号語として M 個の符号語 $(c_1, \dots, c_M)$ を独立に作り，それを符号表と呼ぶ．このように作る符号をランダム符号と呼ぶ．次に以下に示す手順で復号器を作る． Z_x を

$$Z_x(y) \triangleq \mathbf{1} \left\{ i(x; y) > \log \frac{M-1}{2} \right\} \quad (15)$$

と定義する．次に出力 y を受け取った時 $i(c_j; y)$ を計算して式 (15) が 1 となる一番若い番号のメッセージを復号する．また，式 (15) が 1 になる c_j が存在しないときは復号誤りとする． j 番目のメッセージが復号誤りとなるのは， c_j を式 (15) に代入して 1 にならない，または j 番目より若い番号のメッセージが復号されるときなので， j 番目のメッセージを送った時に復

号誤りが起きる確率は,

$$\begin{aligned} & \Pr \left\{ \left\{ z_{c_j(Y)=0} \cup \bigcup_{i < j} \{ z_{c_i(Y)} = 1 \} \right\} \middle| X = c_j \right\} \\ & \leq \Pr \left\{ i(c_j; Y) \leq \log \frac{M-1}{2} \middle| X = c_j \right\} + \sum_{i < j} \Pr \left\{ i(c_i; Y) > \log \frac{M-1}{2} \middle| X = c_i \right\} \end{aligned}$$

と書ける. 次にこの式の j を固定して, ランダム符号の c_j の値に関して平均をとると

$$\begin{aligned} & \sum_{c_j} P_X(c_j) \Pr \left\{ i(c_j; Y) \leq \log \frac{M-1}{2} \middle| X = c_j \right\} \\ & + \sum_{c_j} P_X(c_j) \sum_{i < j} \Pr \left\{ i(c_i; Y) > \log \frac{M-1}{2} \middle| X = c_j \right\} \\ & = \Pr \left\{ i(X; Y) \leq \log \frac{M-1}{2} \right\} + (j-1) \Pr \left\{ i(X; \bar{Y}) > \log \frac{M-1}{2} \right\} \end{aligned}$$

となり, さらに j についても平均をとると,

$$\frac{1}{M} \sum_{j=1}^M \left(\Pr \left\{ i(X; Y) \leq \log \frac{M-1}{2} \right\} + (j-1) \Pr \left\{ i(X; \bar{Y}) > \log \frac{M-1}{2} \right\} \right) \quad (16)$$

$$= \Pr \left\{ i(X; Y) \leq \log \frac{M-1}{2} \right\} + \frac{M-1}{2} \Pr \left\{ i(X; \bar{Y}) > \log \frac{M-1}{2} \right\} \quad (17)$$

となる. よって式 (14) と式 (17) から式 (5) が成り立つ. $\square$

本論文の中では式 (5) を DT Bound と呼ぶ. DT Bound とは Dependence Testing Bound の略である.

3.2 消失通信路における Achievability Bound

消失確率 δ の D 進の消失通信路の復号誤り確率の上界を求める. 本論文の中では Achievable Bound として DT Bound を用いる. 以下の定理は D 進の消失通信路の DT Bound である.

定理 2 消失確率 δ の D 進の消失通信路において復号誤り確率 ϵ が

$$\epsilon \leq \sum_{t=0}^n \binom{n}{t} \delta^t (1-\delta)^{n-t} D^{-[n-t-\log_D \frac{M-1}{2}]^+} \quad (18)$$

となる符号が存在する.

(証明) $\mathcal{A} = \{0, 1, \dots, D-1\}^n, \mathcal{B} = \{0, 1, \dots, D-1\}^n$ とする. 入力が一様分布で, $\mathbf{x}^n, \mathbf{y}^n$ をそれぞれ通信路の入出力とすると, $i(\mathbf{x}^n; \mathbf{y}^n) = \log_D \frac{P_{\mathbf{X}^n \mathbf{Y}^n}(\mathbf{x}^n, \mathbf{y}^n)}{P_{\mathbf{X}^n}(\mathbf{x}^n) P_{\mathbf{Y}^n}(\mathbf{y}^n)}$ は消失していない箇

所の個数となる．なぜなら

$$\begin{aligned} i(a; a) &= \log_D \frac{P_{XY}(x, y)}{P_X(x)P_Y(y)} = \log_D \frac{\frac{1}{D}(1-\delta)}{\frac{1}{D}\frac{1}{D}(1-\delta)} = 1 \\ i(a; e) &= \log_D \frac{P_{XY}(x, y)}{P_X(x)P_Y(y)} = \log_D \frac{\frac{1}{D}\delta}{\frac{1}{D}\delta} = 0 \end{aligned}$$

となるからである．ここで $a = 0, 1, \dots, D-1$ である．したがって $\mathbf{y}^n$ に t 回消失が含まれ $\mathbf{x}^n$ と消失していない箇所では一致しているとき $i(\mathbf{x}^n; \mathbf{y}^n) = n - t$ となる．式 (5) は対数と指数の底を同じ数で置き換えても成り立つ．式 (5) の対数と指数の底を D に置き換えた式に $i(\mathbf{x}^n; \mathbf{y}^n)$ の値を代入すると、

$$\epsilon \leq \mathbb{E} \left[D^{-[n-t-\log_D \frac{M-1}{2}]^+} \right] \quad (19)$$

となる．また t 回消失する確率は $\binom{n}{t} \delta^t (1-\delta)^{n-t}$ だから、期待値の計算をすると式 (18) に変形できる．□

また、式 (18) に $D = 2$ を代入することにより文献 [1] に記載されている二元消失通信路の DT Bound と同じ結果が導出できる．

3.3 消失通信路における Converse Bound

この節では D 進の消失通信路の復号誤り確率の下界である Converse Bound を次の定理として示す．

定理 3 消失確率 δ の D 進の消失通信路の復号誤り確率 ϵ はたとえ偶然的にエンコーダーが消失のありかを知ったとしても、

$$\epsilon \geq \sum_{l=\lfloor n-\log_D M \rfloor + 1}^n \binom{n}{l} \left(1 - \frac{D^{n-l}}{M}\right) \delta^l (1-\delta)^{n-l} \quad (20)$$

を満たす．

(証明) まず M 個の入力のうち識別可能な出力が J 個であった場合を考える．その時正しく復号できる確率は、入力した符号語が A で、出力が B であったとして C が入力されたと判断

したとすると

$$\Pr\{C = A\} = \sum_{a=1}^M \Pr\{C = A = a\} \quad (21)$$

$$= \sum_{a=1}^M \sum_{b=1}^J \Pr\{C = A = a, B = b\} \quad (22)$$

$$= \sum_{a=1}^M \sum_{b=1}^J P_A(a) P_{B|A}(b | a) P_{C|A,B}(c | a, b) \quad (23)$$

$$= \sum_{a=1}^M \sum_{b=1}^J P_A(a) P_{B|A}(b | a) P_{C|B}(c | b) \quad (24)$$

$$= \sum_{b=1}^J \frac{1}{M} \sum_{a=1}^M P_{C|B}(a | b) \quad (25)$$

$$\leq \sum_{b=1}^J \frac{1}{M} \sum_{a=1}^M P_{C|B}(c | b) \quad (26)$$

$$= \frac{J}{M} \quad (27)$$

となり上から評価できる． D 進の消失通信路で消失する箇所がわかっていて消失した回数が $z \in \{0, 1, \dots, n\}$ として考える場合，式 (27) 中の J の値は $J = D^{n-z}$ となるからこの条件の下での復号誤り確率はを導出すると

$$\Pr\{ \text{正しく復号} \mid \text{消失した箇所がわかっている, } z \text{ 回消失} \} \leq \frac{D^{n-z}}{M} \quad (28)$$

$$\Pr\{ \text{復号誤り} \mid \text{消失した箇所がわかっている, } z \text{ 回消失} \} \geq \left[1 - \frac{D^{n-z}}{M} \right]^+ \quad (29)$$

となる．式 (29) の左辺を p_D とすると復号誤り確率 ϵ は

$$\begin{aligned} \epsilon &= \Pr\{ \text{復号誤り} \} \\ &= \sum_{\text{消失している箇所}} \sum_{z \text{ 回消失}} p_D \Pr\{ \text{消失した箇所が分かっている, } z \text{ 回消失} \} \\ &\geq \sum_{z=0}^n \binom{n}{z} \left[1 - \frac{D^{n-z}}{M} \right]^+ \delta^z (1 - \delta)^{n-z} \\ &= \sum_{l=\lfloor n - \log_D M \rfloor + 1}^n \binom{n}{l} \left(1 - \frac{D^{n-l}}{M} \right) \delta^l (1 - \delta)^{n-l} \end{aligned}$$

となり， D 進の消失通信路の復号誤り確率は式 (20) を満たす．□

また，式 (20) に $D = 2$ を代入することにより文献 [1] に記載されている二元消失通信路の Converse Bound と同じ結果が導出できる．

3.4 有限長で達成可能なレート

3.2 節と 3.3 節で求めた DT Bound と Achievability Bound の式からアルファベットサイズが 2 から 9 それぞれの時の消失通信路について誤り確率が 10^{-3} の時各符号語長に対して送信可能なレートの上下界を図 2 から図 9 に示した．また，同様に誤り確率が 10^{-6} の時についても送信可能なレートの上下界を図 10 から図 17 に示した．各アルファベットサイズと誤り確率のときのレートの求め方は付録 A に記した．また図 2 から図 17 の一部分を拡大した図を図 18 から図 33 に示した．図 18 から図 33 を見ると消失通信路はアルファベットサイズを大きくするほど DT Bound が Converse Bound に対して値が近くなる傾向が見られた．次に通信路容量で正規化したとき誤り確率が 10^{-3} と 10^{-6} の時の DT Bound と Converse Bound を図 34 と図 36, 図 35 と図 37 に示した．またそれぞれを拡大した図を図 38 から図 41 に示した．図 38 と図 40 から DT Bound はアルファベットサイズを大きくするほど通信路容量から遠のく結果が見られた．それに対して，図 39 と図 41 から Converse Bound はアルファベットサイズを大きくするほど通信路容量に近づく結果が見られた．

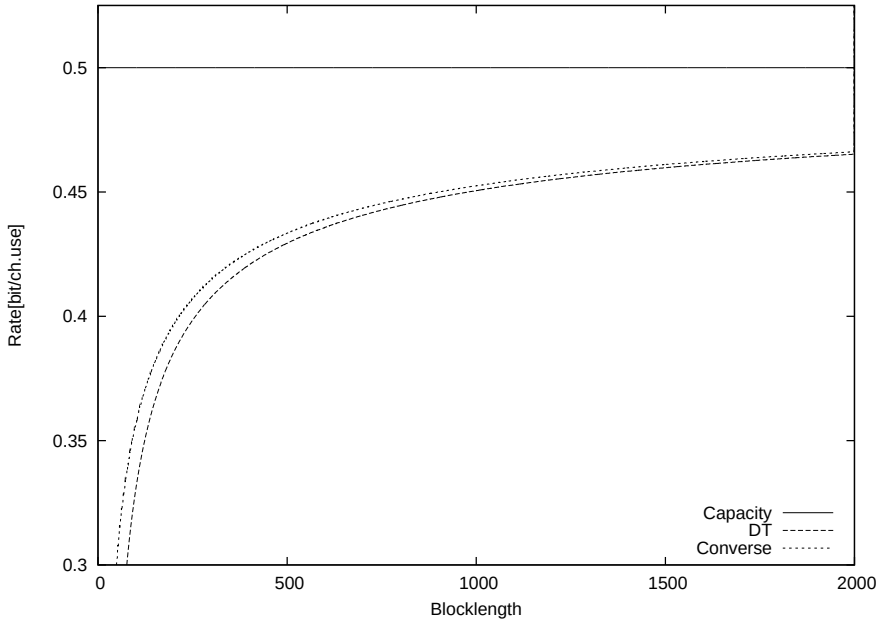


図 2 $\delta = 0.5$, $\epsilon = 10^{-3}$, $D = 2$ の消失通信路のレート

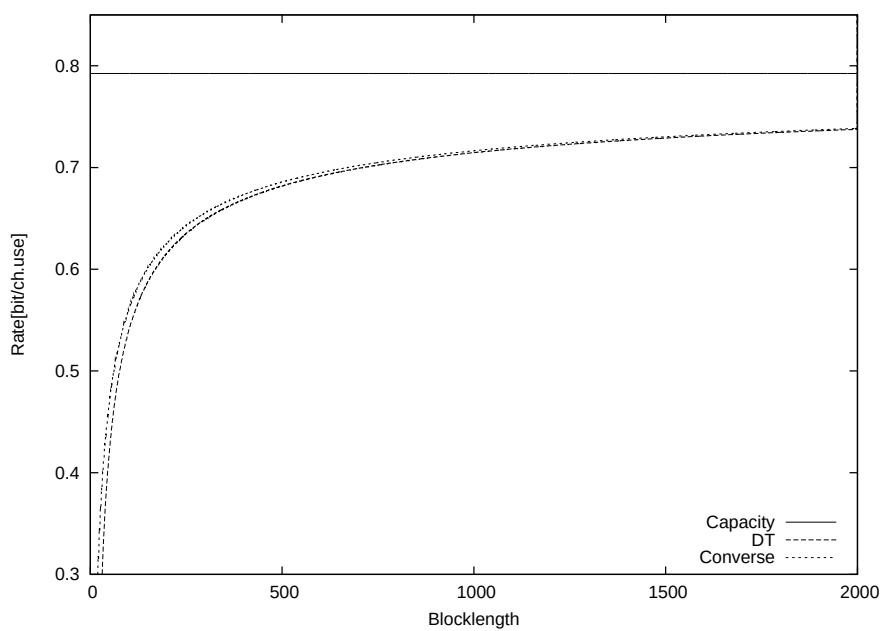


図3 $\delta = 0.5$, $\epsilon = 10^{-3}$, $D = 3$ の消失通信路のレート

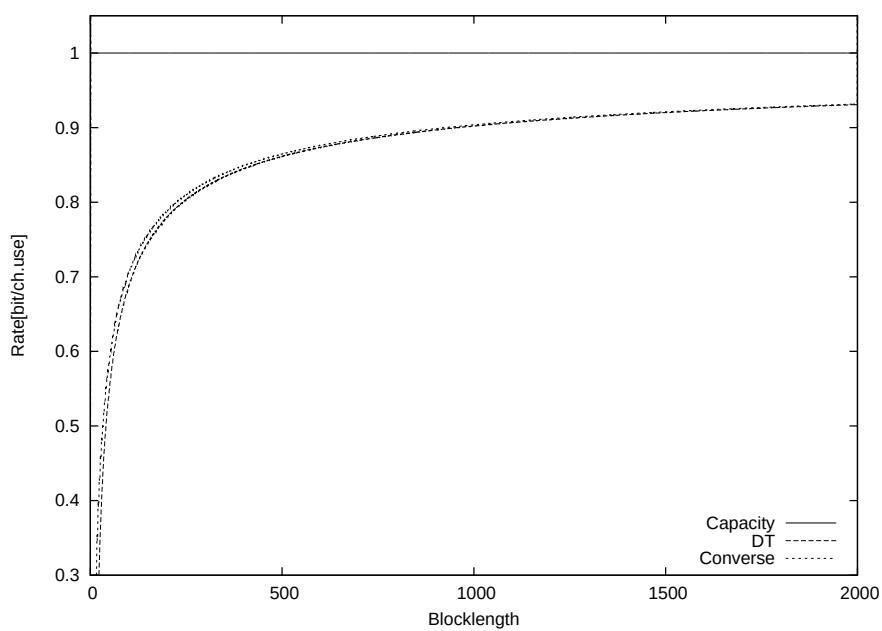


図4 $\delta = 0.5$, $\epsilon = 10^{-3}$, $D = 4$ の消失通信路のレート

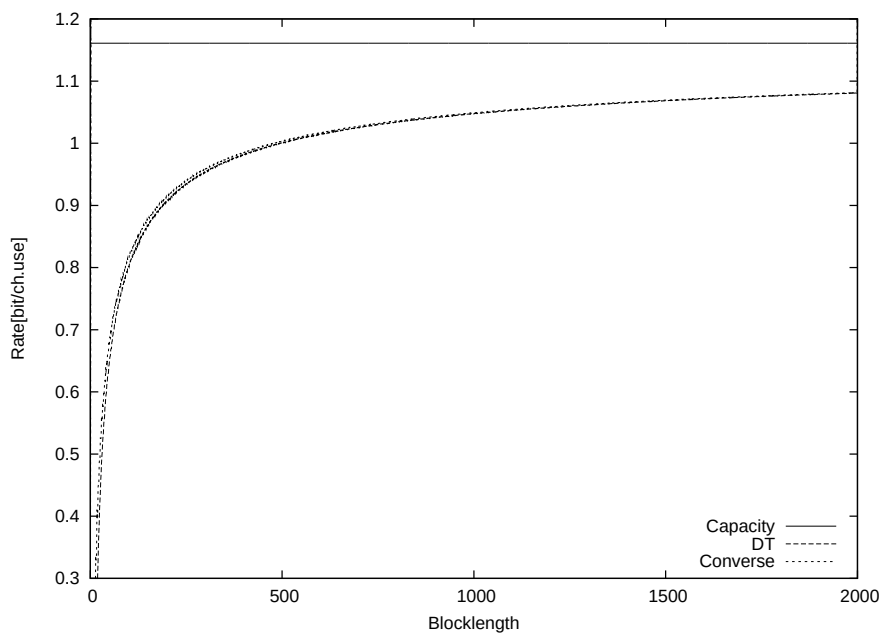


図 5 $\delta = 0.5$, $\epsilon = 10^{-3}$, $D = 5$ の消失通信路のレート

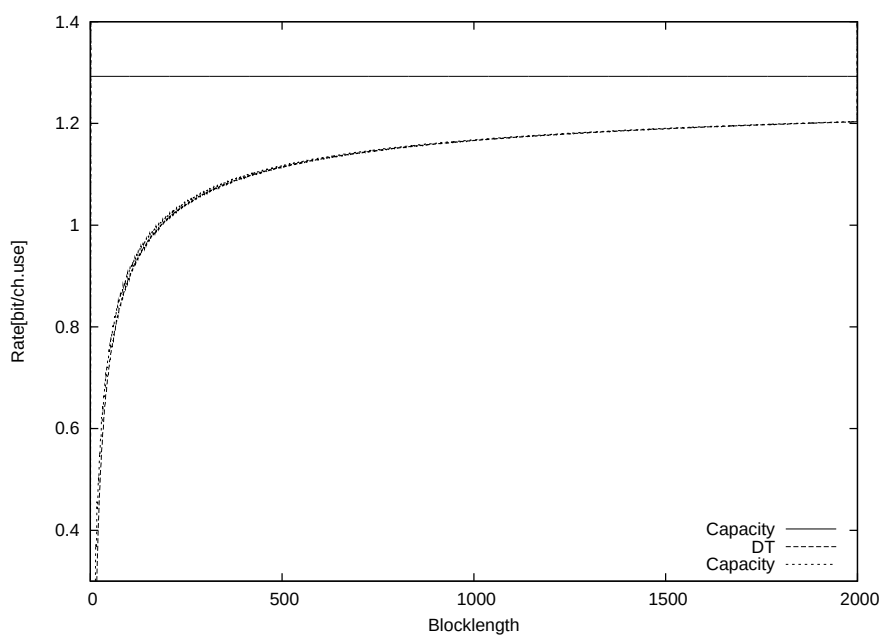


図 6 $\delta = 0.5$, $\epsilon = 10^{-3}$, $D = 6$ の消失通信路のレート

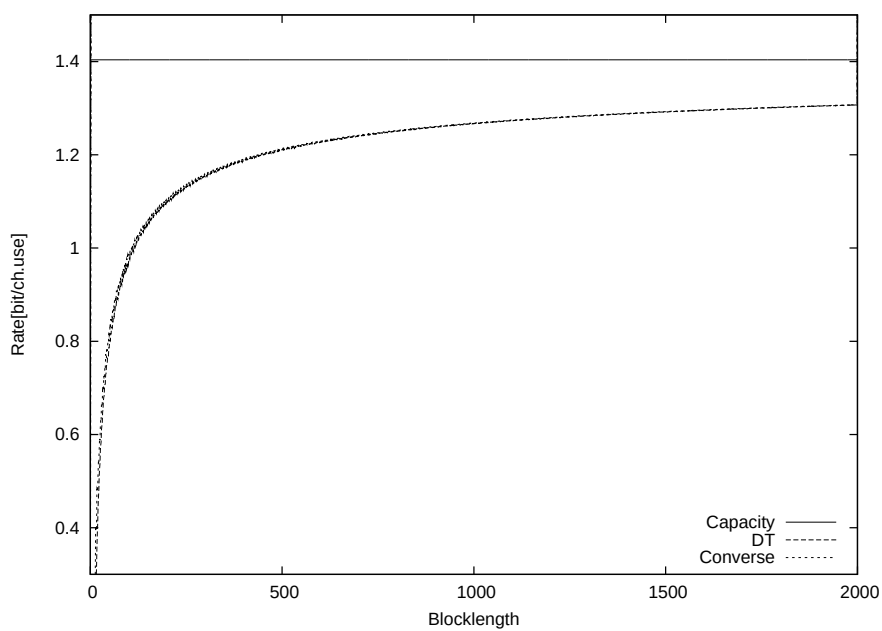


図 7 $\delta = 0.5$, $\epsilon = 10^{-3}$, $D = 7$ の消失通信路のレート

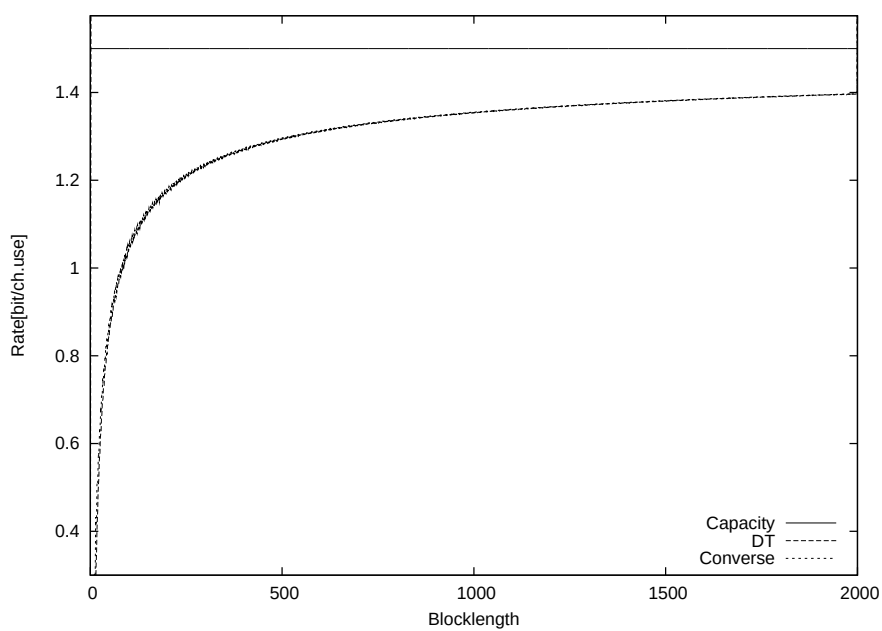


図 8 $\delta = 0.5$, $\epsilon = 10^{-3}$, $D = 8$ の消失通信路のレート

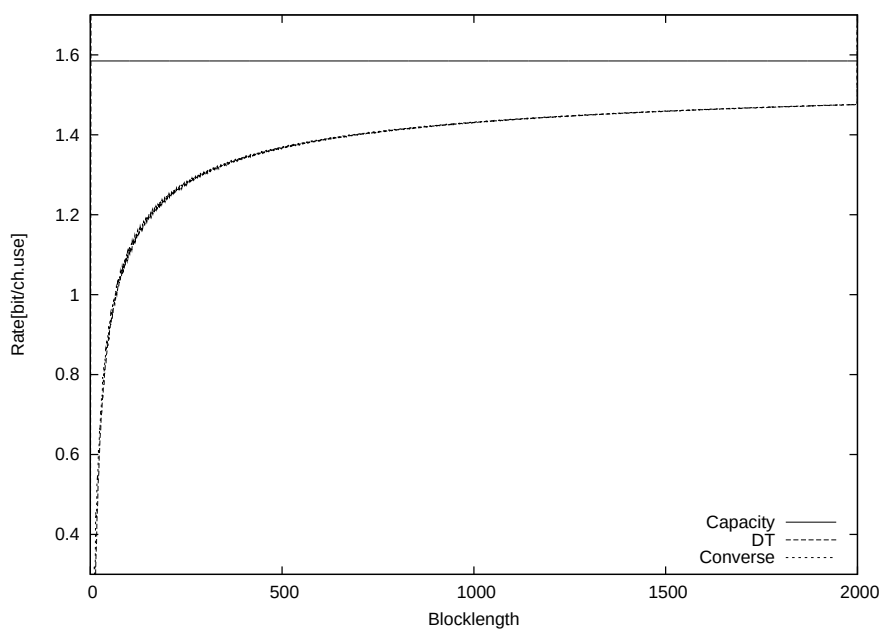


図 9 $\delta = 0.5$, $\epsilon = 10^{-3}$, $D = 9$ の消失通信路のレート

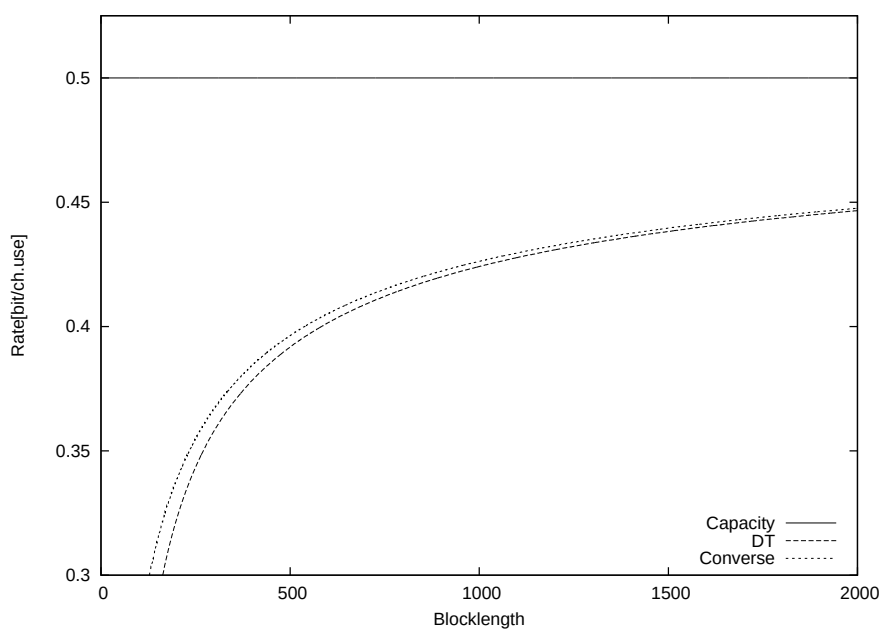


図 10 $\delta = 0.5$, $\epsilon = 10^{-6}$, $D = 2$ の消失通信路のレート

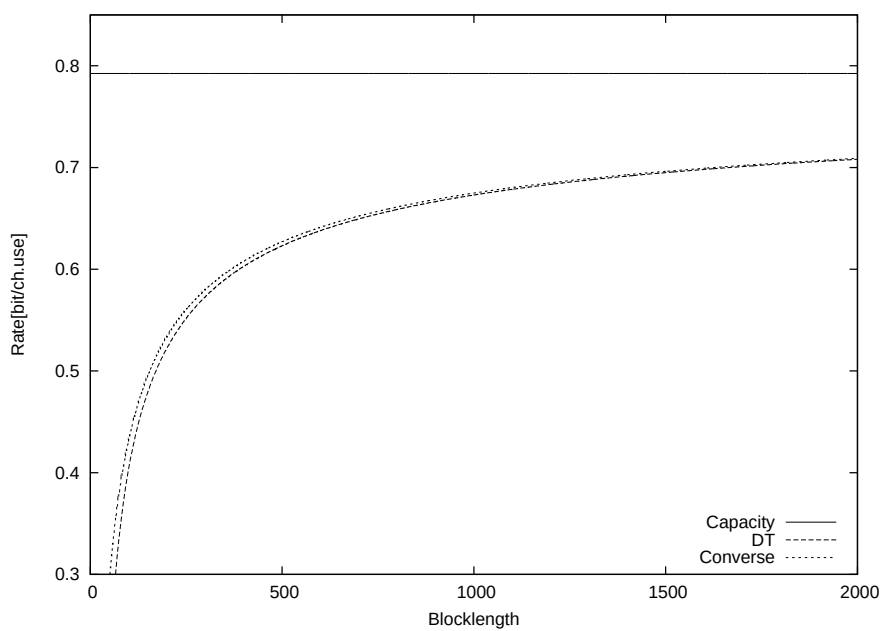


図 11 $\delta = 0.5$, $\epsilon = 10^{-6}$, $D = 3$ の消失通信路のレート

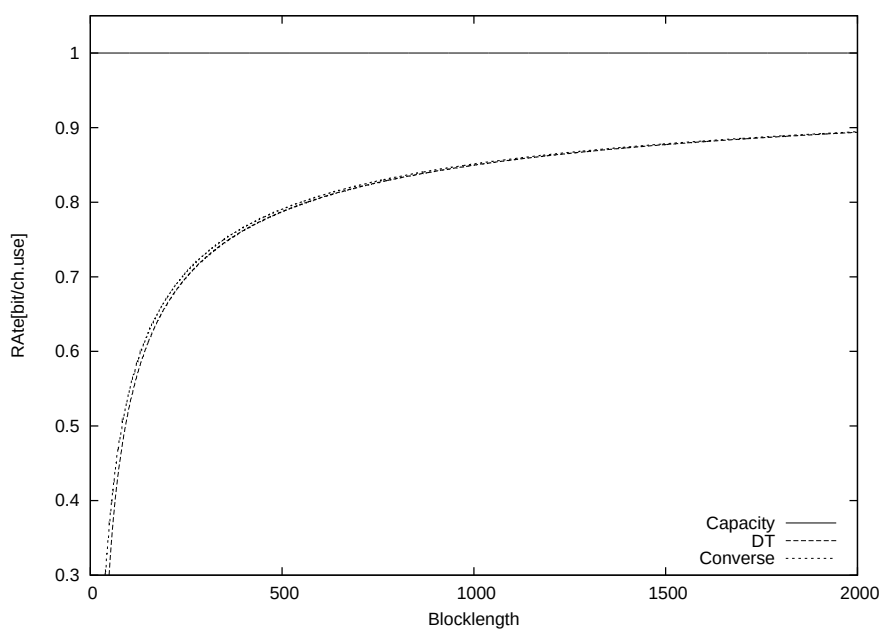


図 12 $\delta = 0.5$, $\epsilon = 10^{-6}$, $D = 4$ の消失通信路のレート

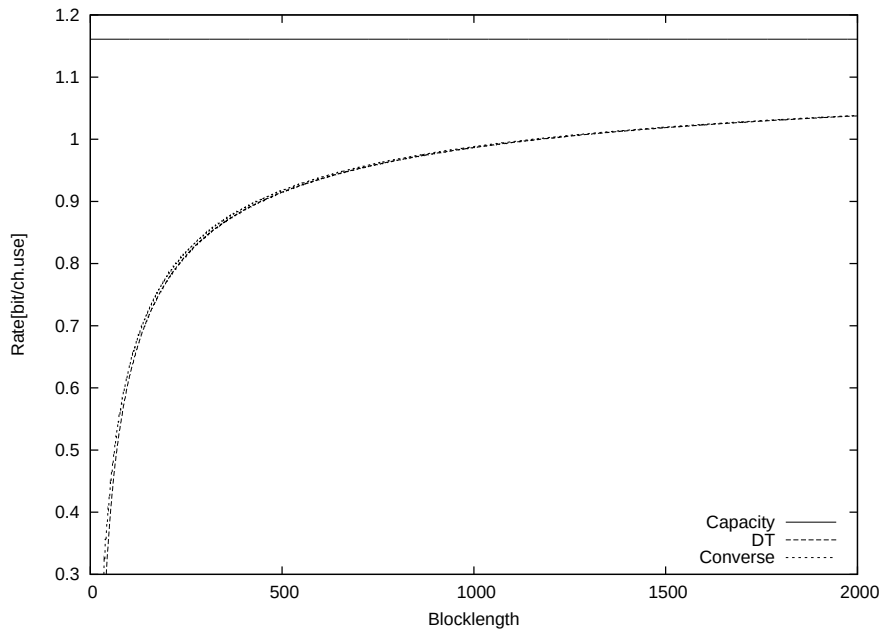


図 13 $\delta = 0.5$, $\epsilon = 10^{-6}$, $D = 5$ の消失通信路のレート

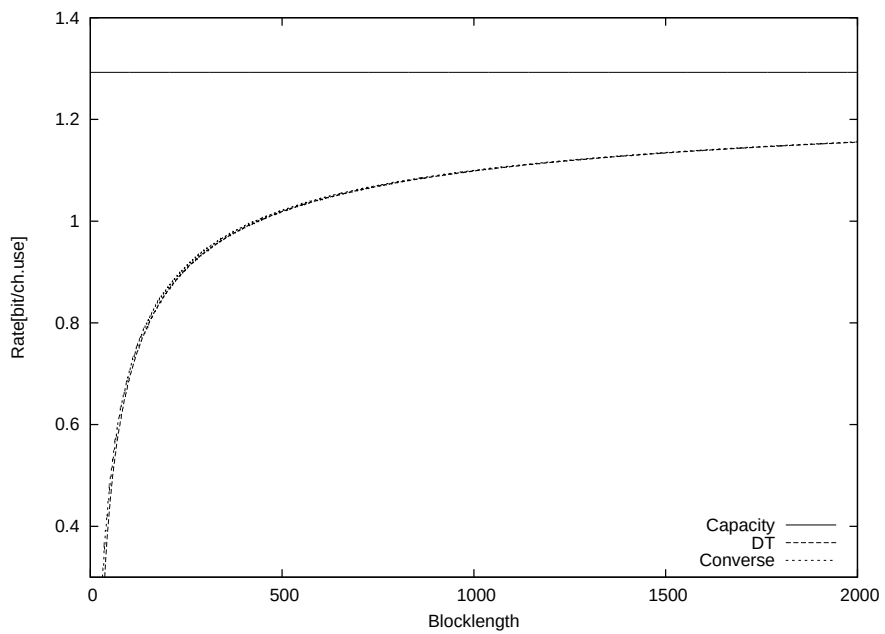


図 14 $\delta = 0.5$, $\epsilon = 10^{-6}$, $D = 6$ の消失通信路のレート

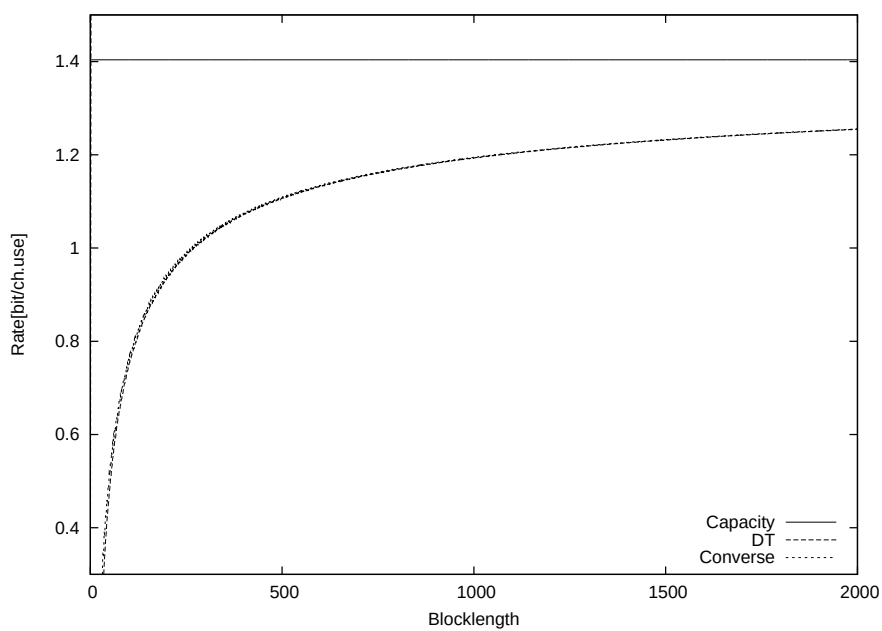


図 15 $\delta = 0.5$, $\epsilon = 10^{-6}$, $D = 7$ の消失通信路のレート

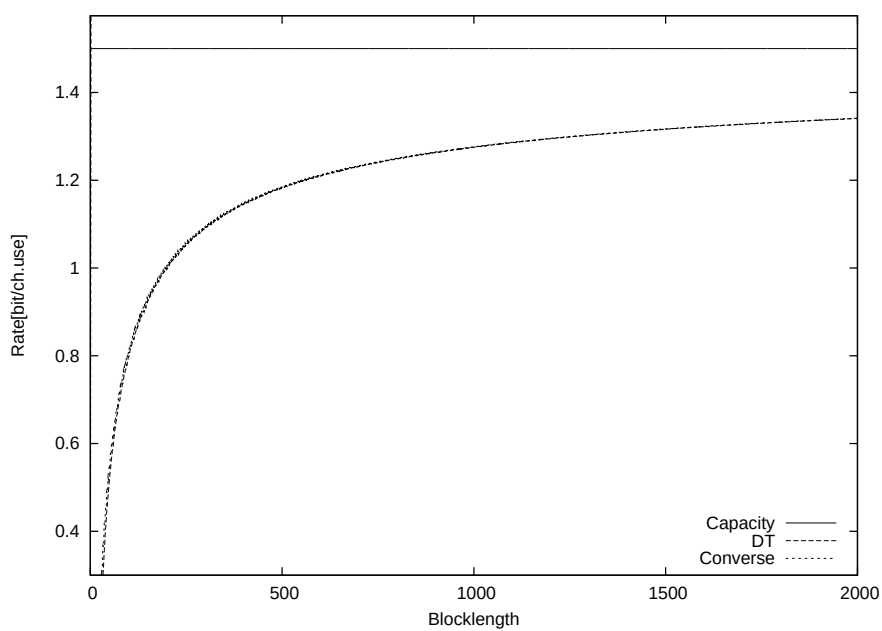


図 16 $\delta = 0.5$, $\epsilon = 10^{-6}$, $D = 8$ の消失通信路のレート

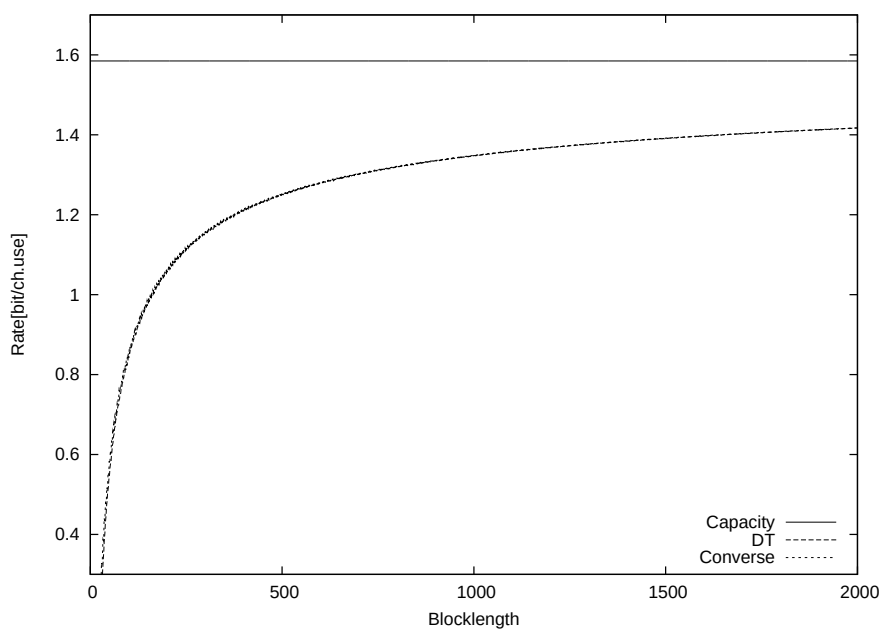


図 17 $\delta = 0.5$, $\epsilon = 10^{-6}$, $D = 9$ の消失通信路のレート

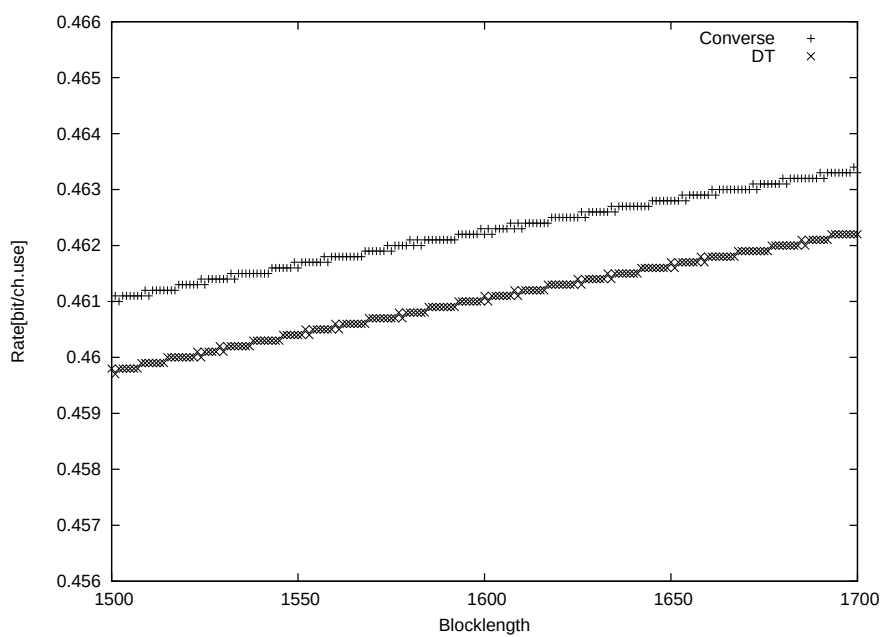


図 18 図 2 の拡大図

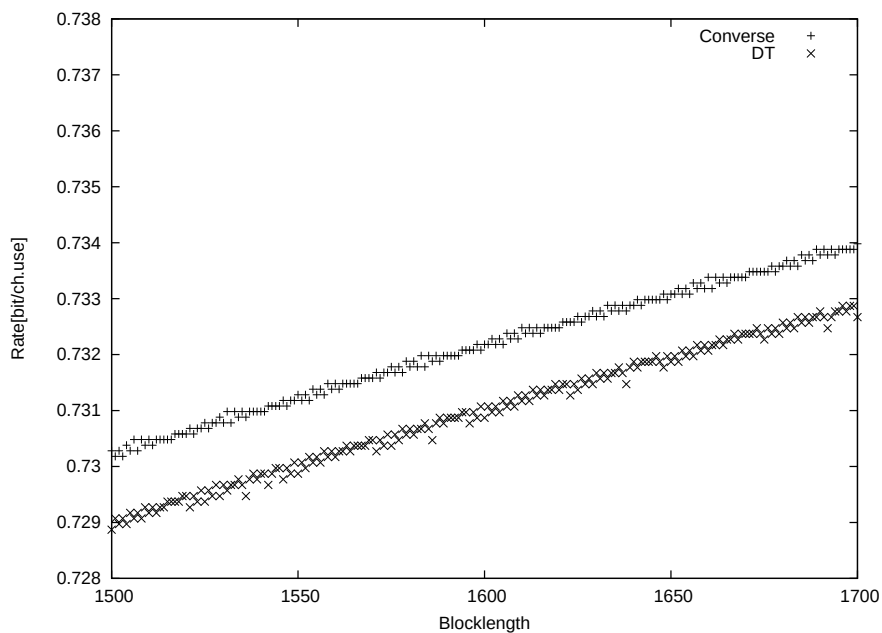


図 19 図 3 の拡大図

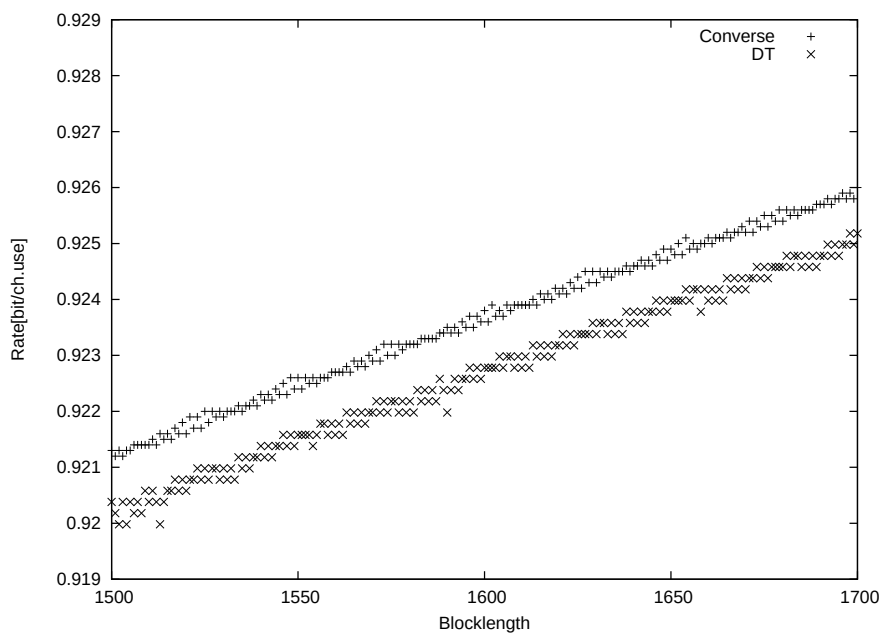


図 20 図 4 の拡大図

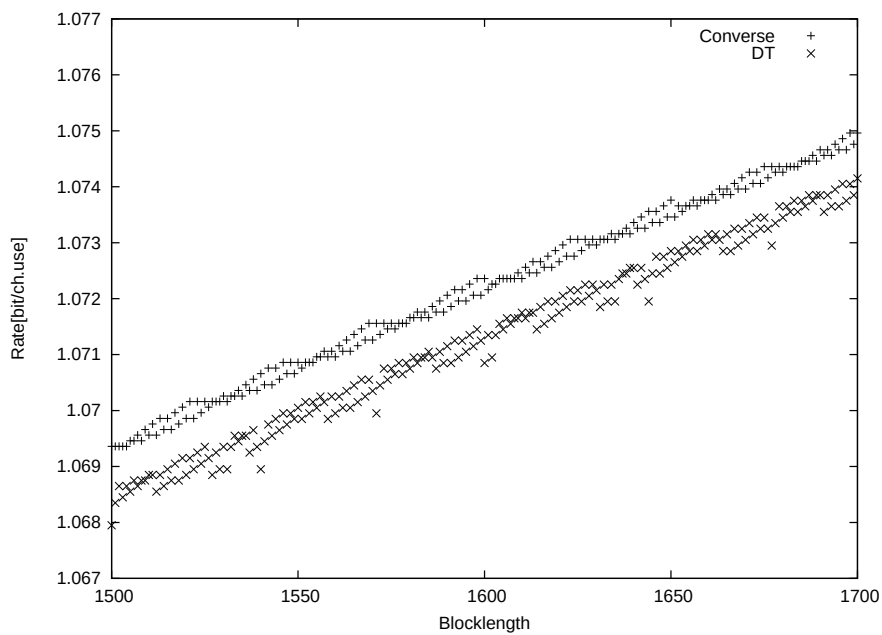


図 21 図 5 の拡大図

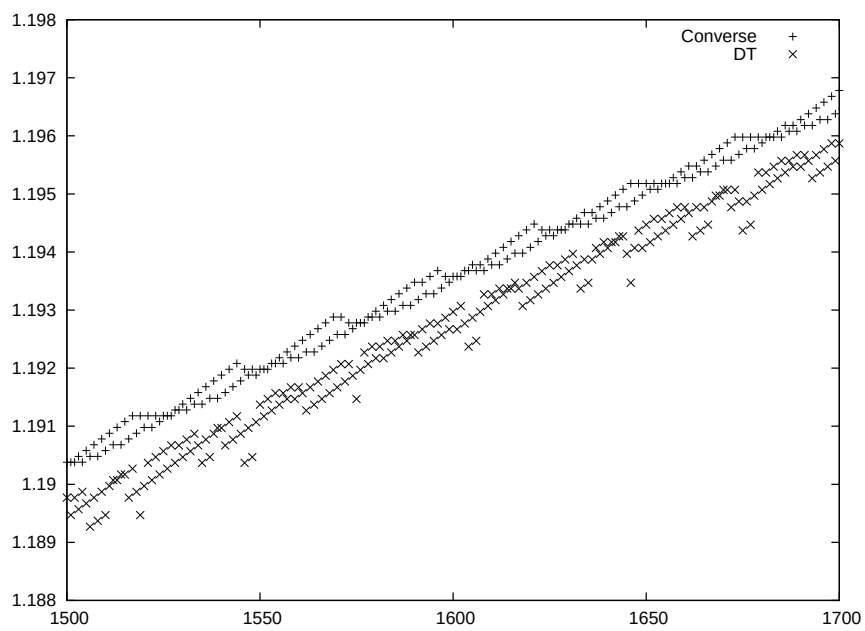


図 22 図 6 の拡大図

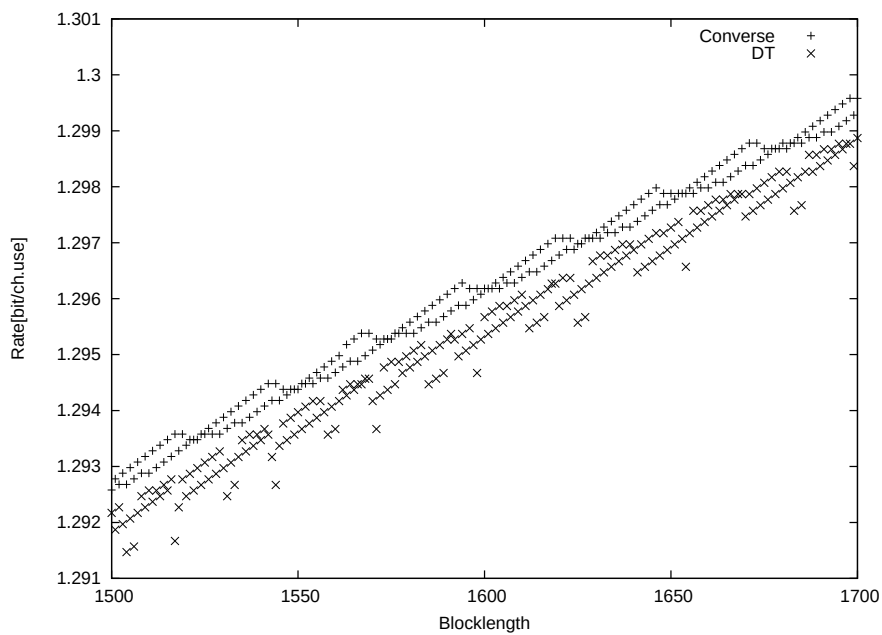


図 23 図 7 の拡大図

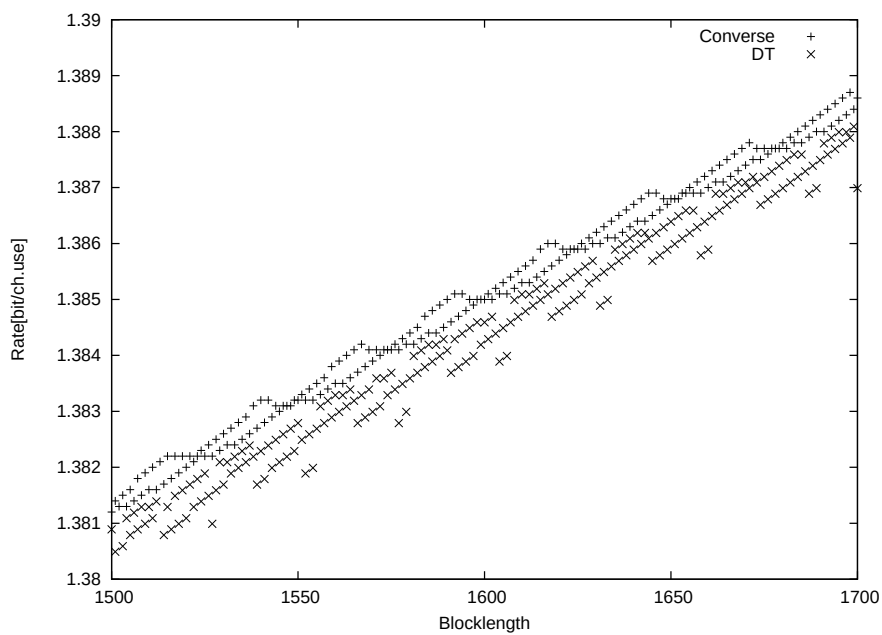


図 24 図 8 の拡大図

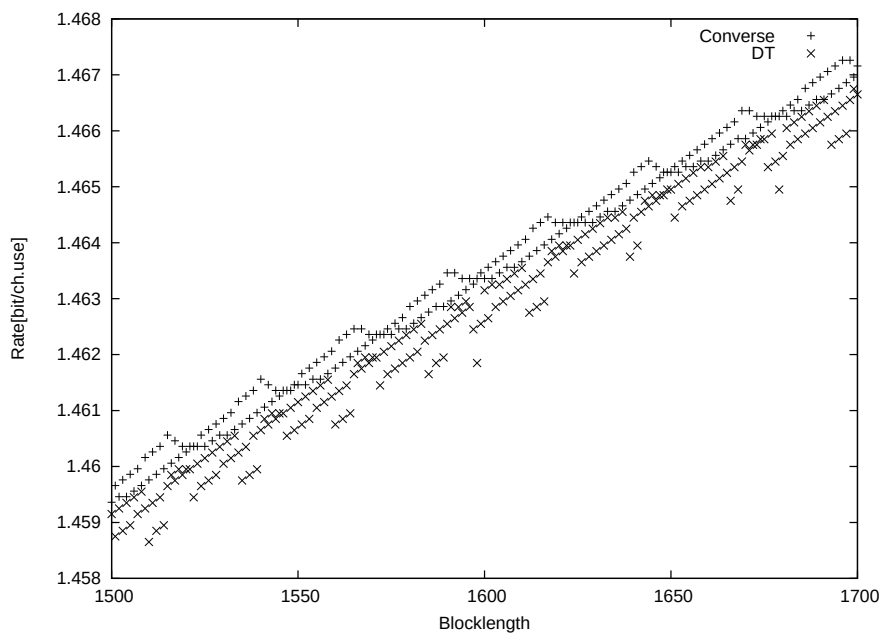


図 25 図 9 の拡大図

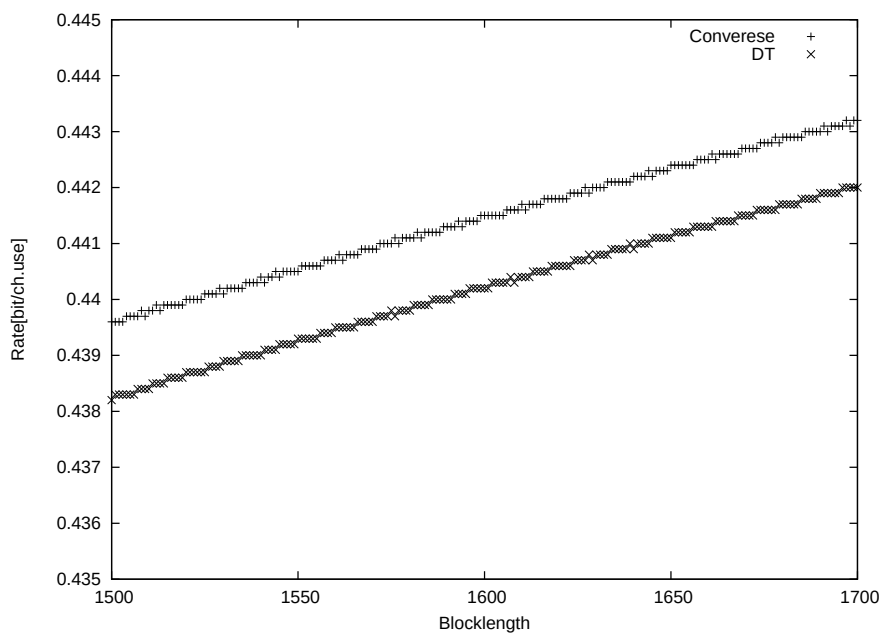


図 26 図 10 の拡大図

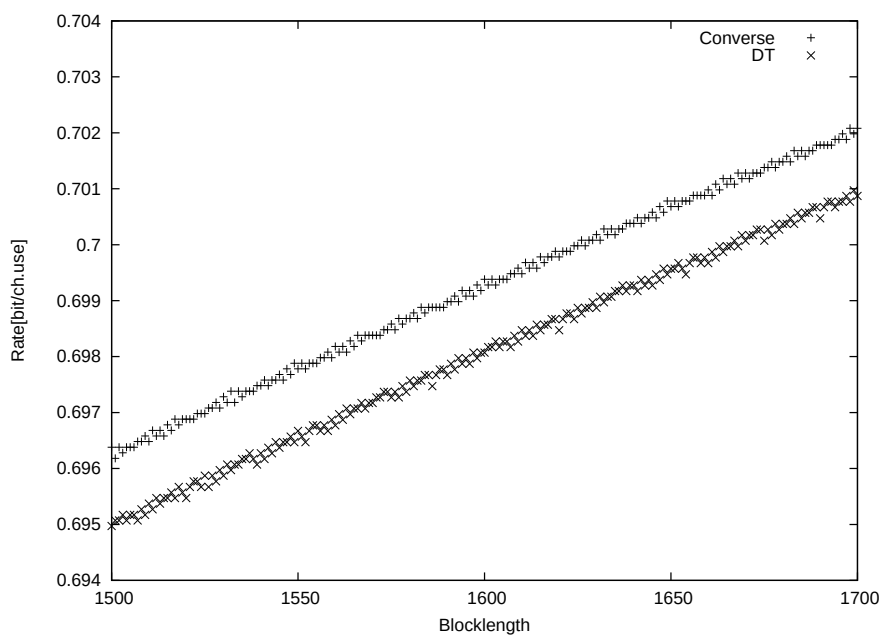


図 27 図 11 の拡大図

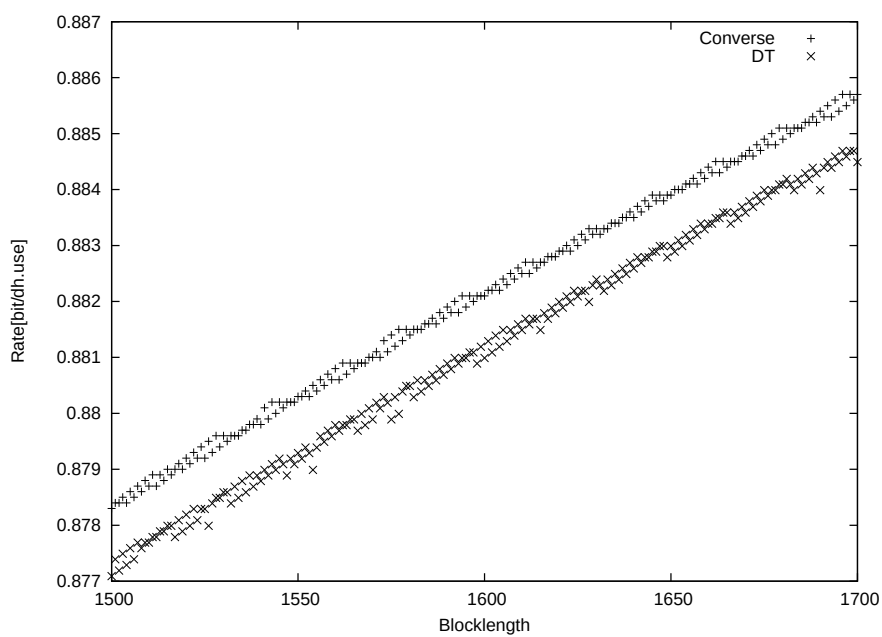


図 28 図 12 の拡大図

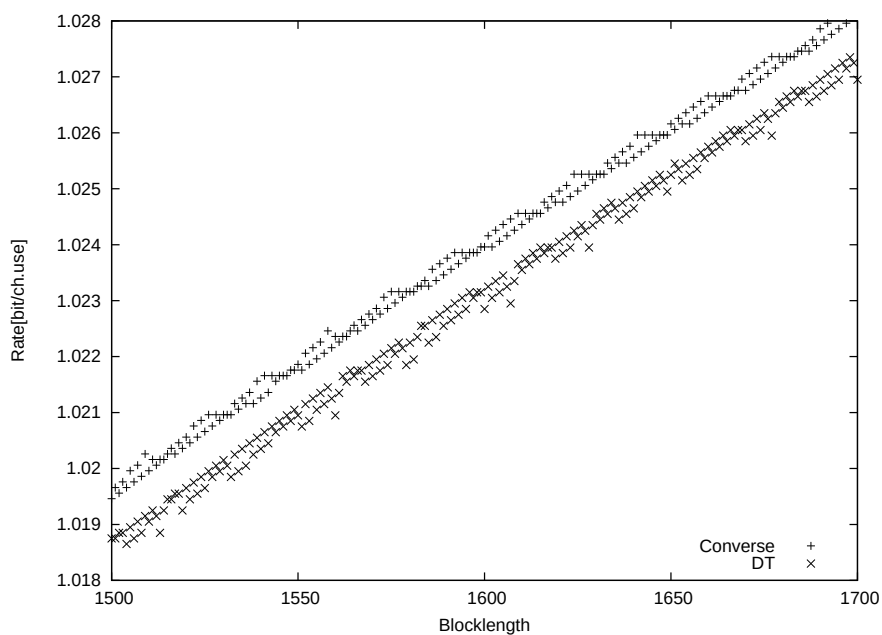


図 29 図 13 の拡大図

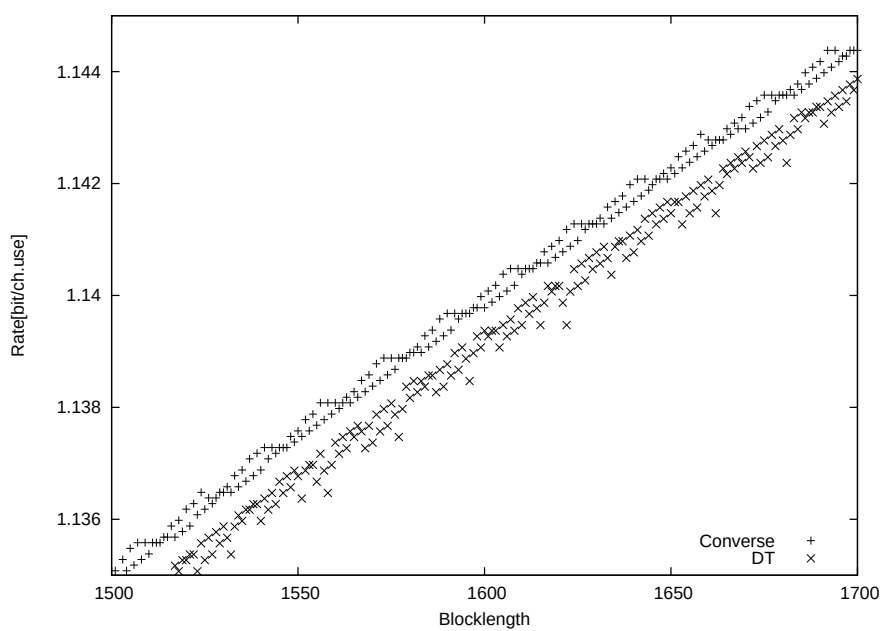


図 30 図 14 の拡大図

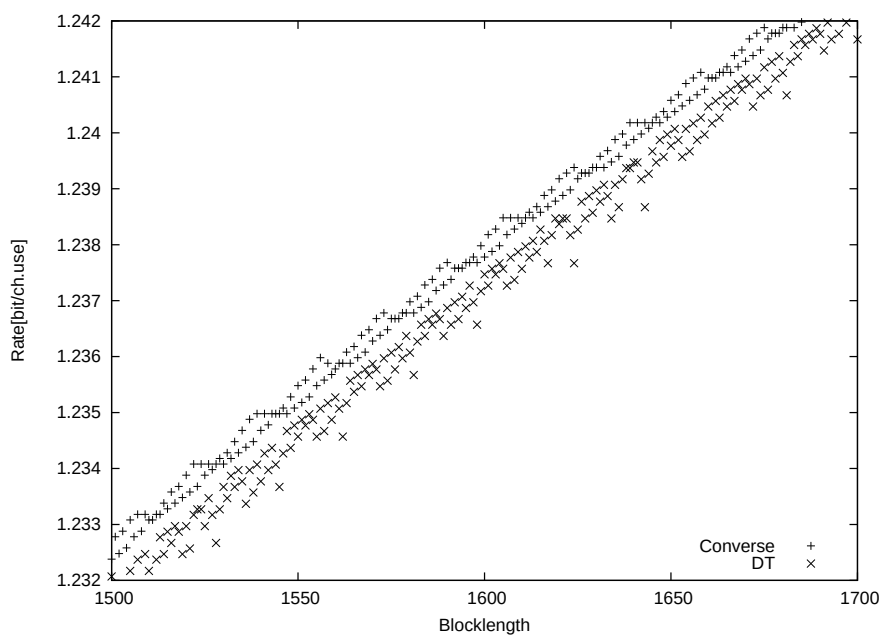


図 31 図 15 の拡大図

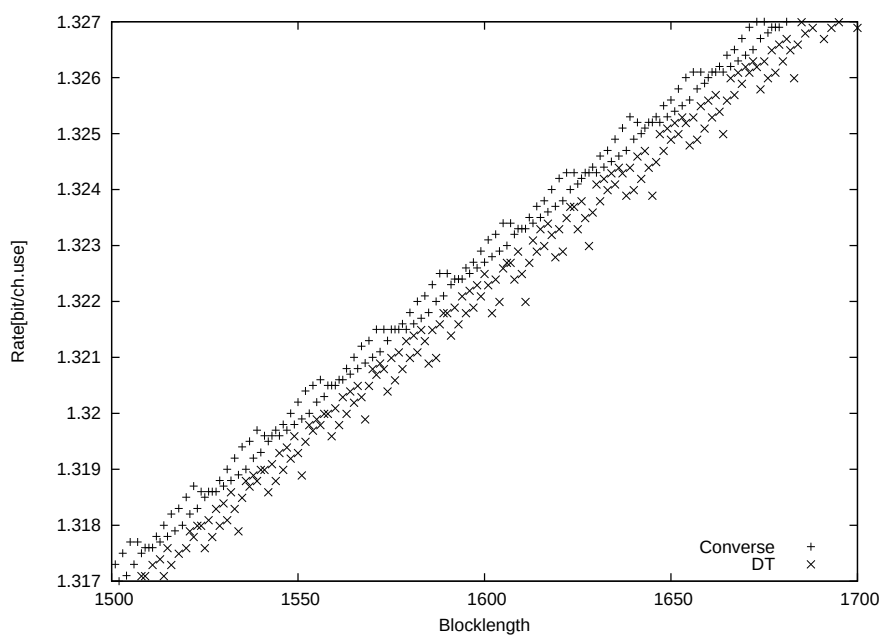


図 32 図 16 の拡大図

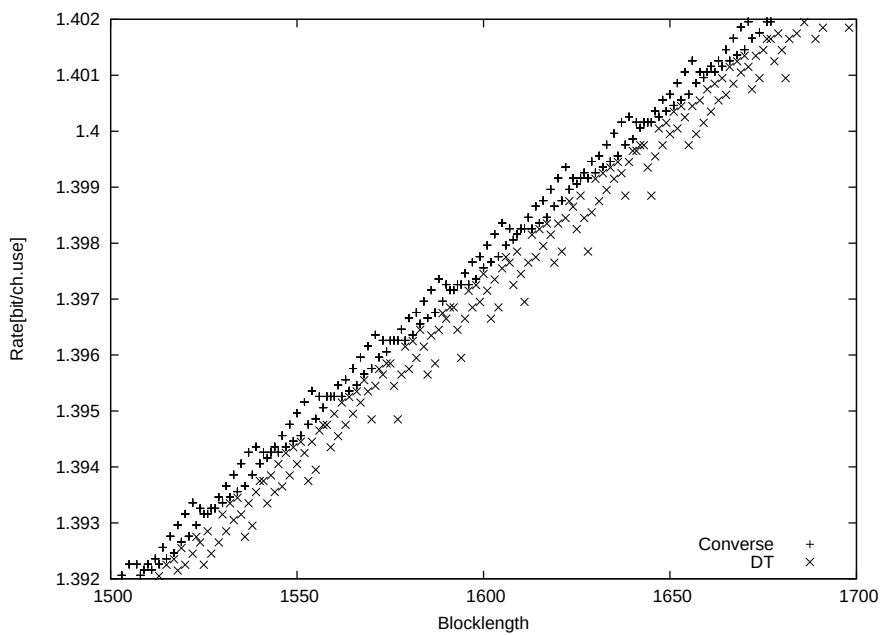


図 33 図 17 の拡大図

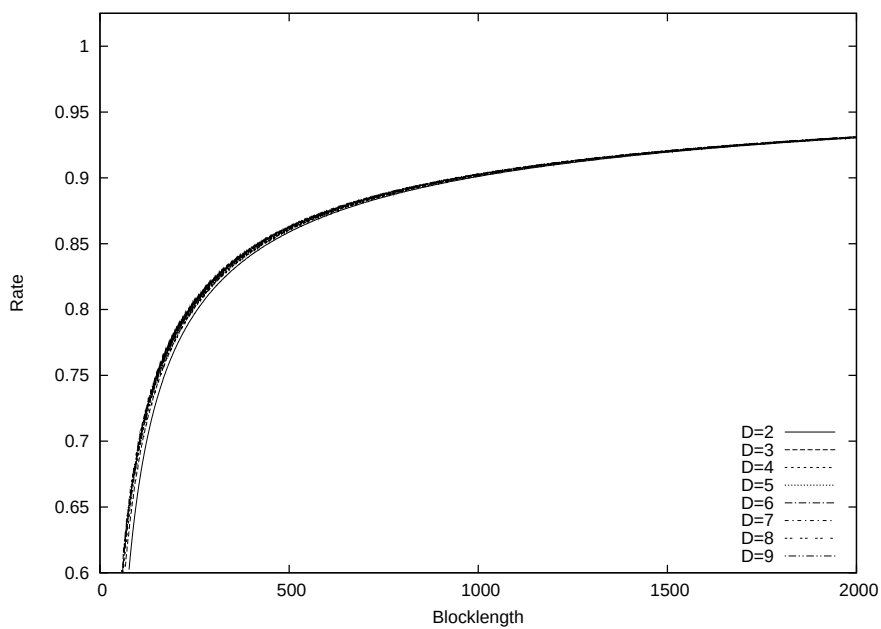


図 34 $\delta = 0.5$, $\epsilon = 10^{-3}$ の消失通信路の $DTBound$ のレートの正規化

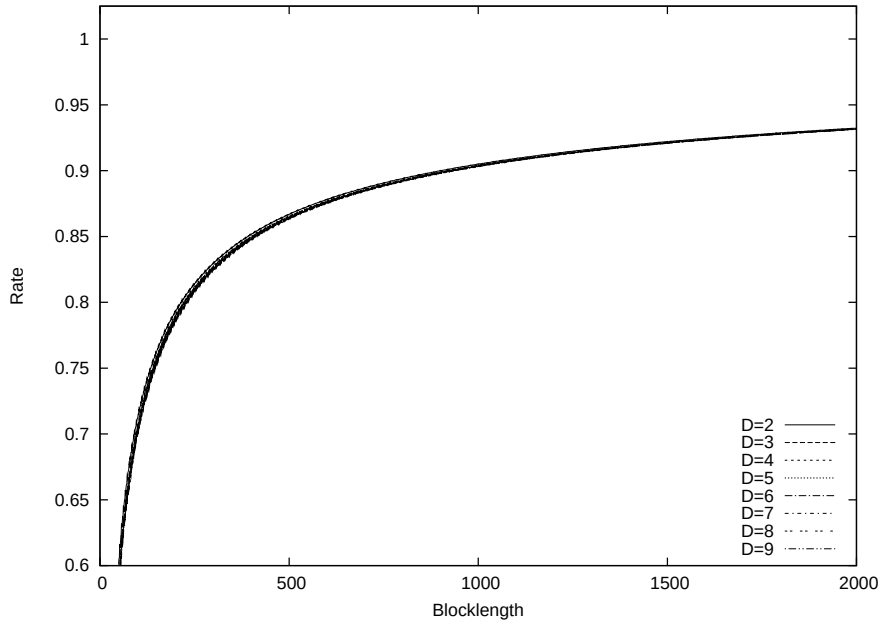


図 35 $\delta = 0.5$, $\epsilon = 10^{-3}$ の消失通信路の *Converse Bound* のレートの正規化

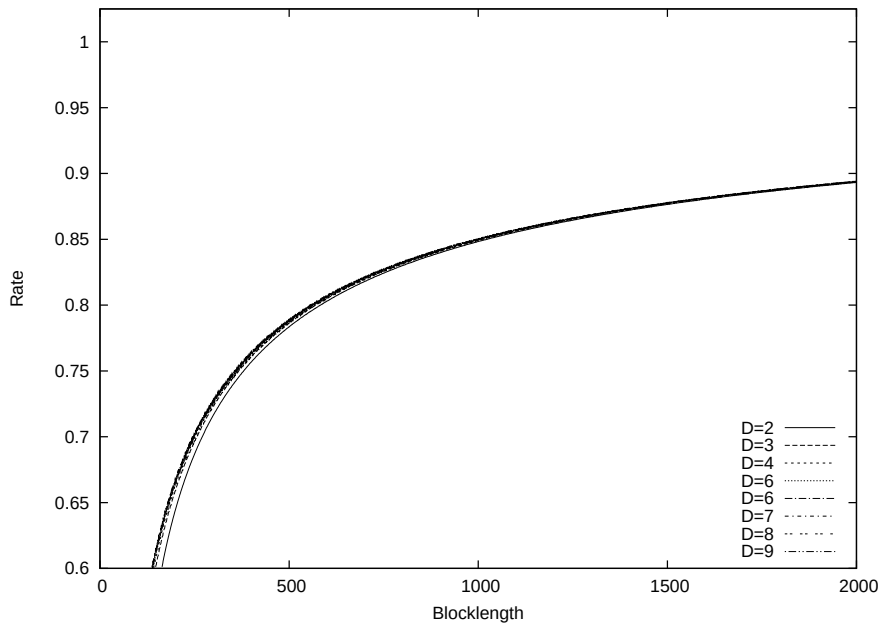


図 36 $\delta = 0.5$, $\epsilon = 10^{-6}$ の消失通信路のレートの *DT Bound* の正規化

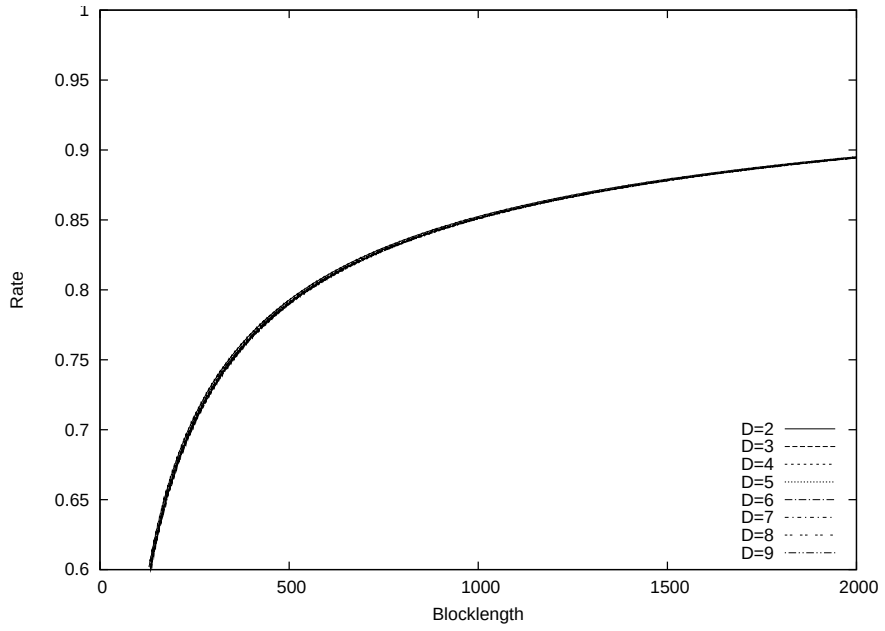


図 37 $\delta = 0.5$, $\epsilon = 10^{-6}$ の消失通信路のレートの *ConverseBound* の正規化

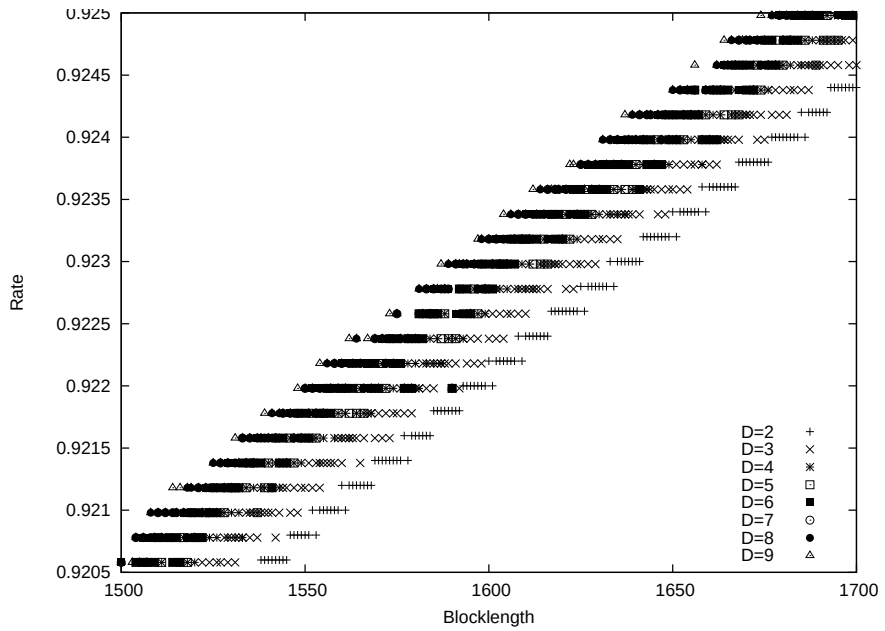


図 38 図 34 の拡大図

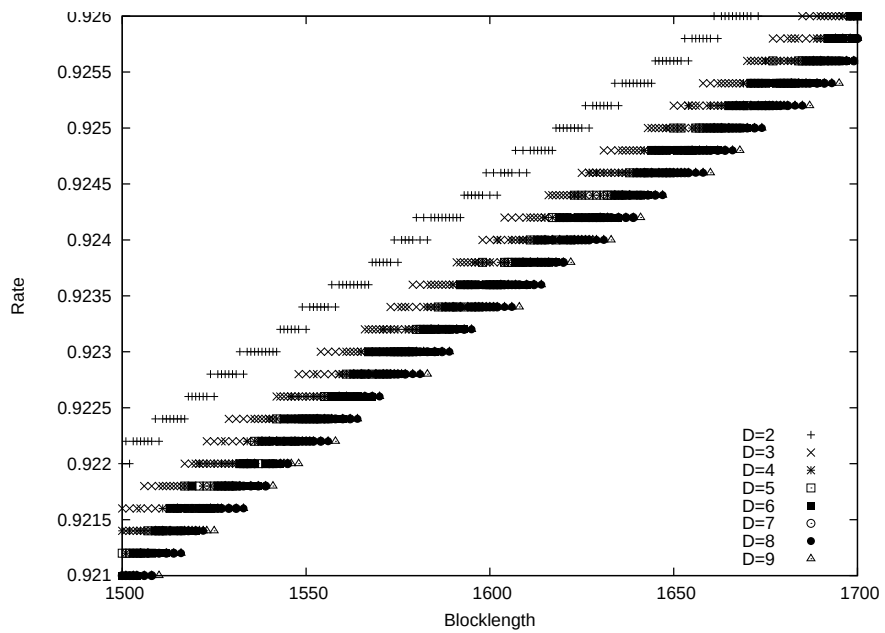


図 39 図 35 の拡大図

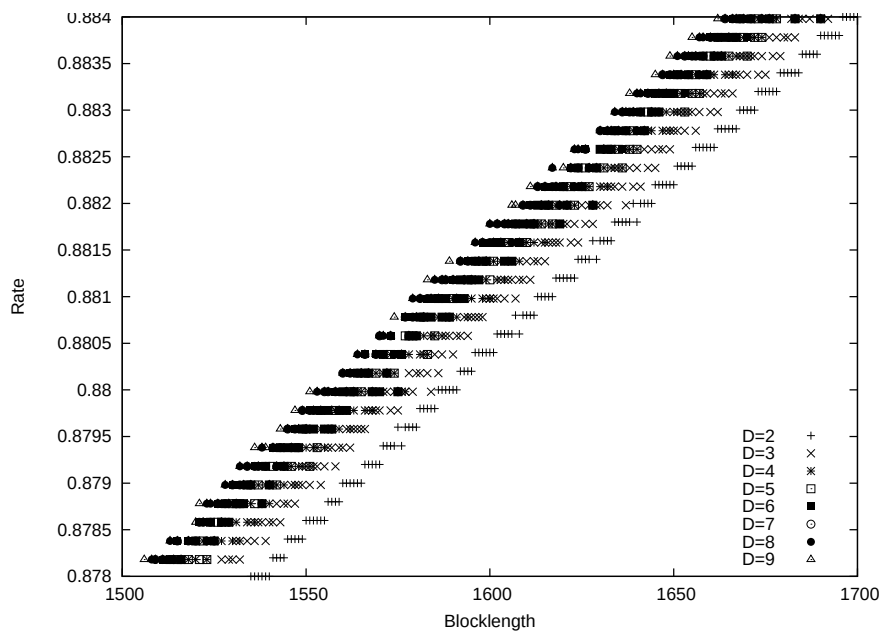


図 40 図 36 の拡大図

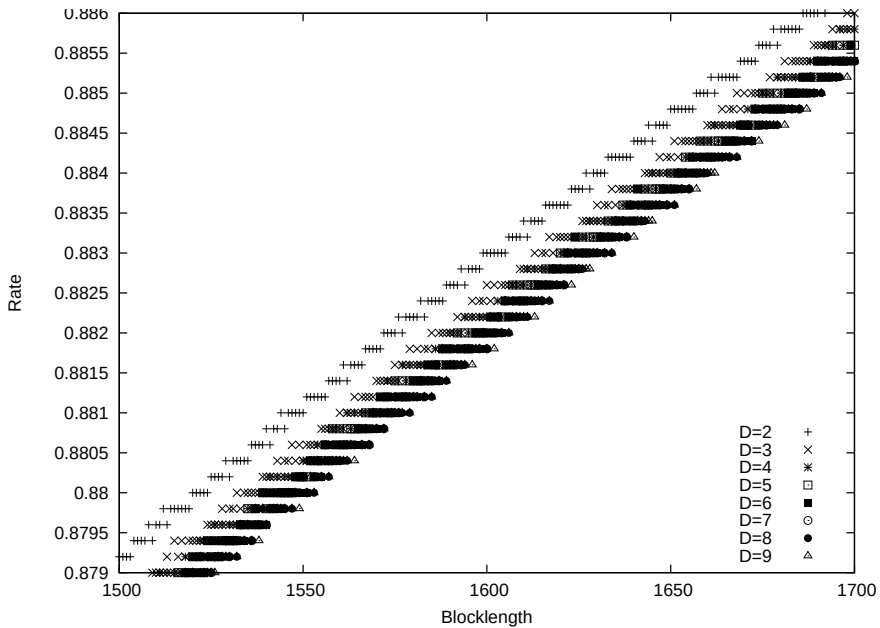


図 41 図 37 の拡大図

4 まとめ

本研究では消失通信路に対してアルファベットサイズを変更しながら、符号語長が有限の時に送信可能なレートの上下界を求め、upper bound が通信路容量からどれだけ下回っているかを調べた。また、その結果からアルファベットサイズを大きくすることでどのような変化が見られるかを調べた。その結果アルファベットサイズを大きくすると送信可能なレートの lower bound と upper bound の間隔が狭くなる傾向が見られた。またアルファベットサイズを大きくするほど lower bound は通信路容量に近づき、upper bound は通信路容量から遠ざかることで間隔が狭くなる傾向が見られた。

謝辞

本研究を行うにあたって、細かく指導してくださった指導教員の西新幹彦准教授に感謝の意を表す。

参考文献

- [1] Yury Polyanskiy, H. Vincent Poor, and Sergio Verdú, “Channel Coding Rate in the Finite Blocklength Regime,” pp.2307–2359, May 2010.
- [2] 小林欣吾・森田啓義, 情報理論講義, 培風館, 2008.
- [3] 韓太舜, 情報理論における情報スペクトル的方法, 培風館, 1998.

付録 A レートの測定

A.1 DT Bound からレートを測るプログラム

```
#include<stdio.h>
#include<math.h>

double
binomial(int n, int t, double d)
{
    int i;
    double term;

    term = 1.0;
    for (i = 0; i < t / 2; i++){
        term *= ((double)(n - i) / (t - i)) * d;
    }
    term *= pow(1 - d, n - t);
    for (i = t / 2; i < t; i++){
        term *= ((double)(n - i) / (t - i)) * d;
    }
    return(term);
}

double
bound_DT(int n, double d, double log_m, int a)
{
    double bound = 0.0;
    int t;

    for (t = 0; t <= n; t++){
        double term;
        term = binomial(n, t, d);
        if (n - t - log_m + 1 > 0){
            term *= pow(a, -(n - t - log_m + log(2, a)));
        }
        bound += term;
    }
    return(bound);
}

double
rate_DT(int n, double d, double e, int a)
{
    double log_m;
    double b;
    double step;

    log_m = 0.5 * n * log2(a);
    for (step = 0.1; step > 0.00001; step /= 10){
        while (e < (b = bound_DT(n, d, log_m / log2(a), a))){
            log_m -= step * n;
        }
        log_m += step * n;
    }
    log_m -= step * n;
    return(log_m / n);
}

int
main()
{
    int n;
    \\消失確率の値を入力
    double d = 0.5;
```



```

\\復号誤り確率の値を入力
double e = 1e-3;
double r;
\\アルファベットサイズの値を入力
int a = 2;

for (n = 1; n < 2001; n++){
r = rate_DT(n, d, e, a);
    if (r > 0.0){
        printf("%d,%g\\n", n, r);
    }
}

return(0);
}

```

A.2 Converse Bound からレートを測るプログラム

```

#include<stdio.h>
#include<math.h>

double
binomial(int n, int t, double d)
{
    int i;
    double term;

    term = 1.0;
    for (i = 0; i < t / 2; i++){
        term *= ((double)(n - i) / (t - i)) * d;
    }
    term *= pow(1 - d, n - t);
    for (i = t / 2; i < t; i++){
        term *= ((double)(n - i) / (t - i)) * d;
    }
    return(term);
}

double
bound_converse(int n, double d, double log_m, int a)
{
    int t;
    double bound = 0.0;

    for (t = 0; t <= n; t++){
        double term;
        term = binomial(n, t, d) * (1 - (pow(a, n - t - log_m)));
        if (t > n - log_m){
            bound += term;
        }
    }
    return(bound);
}

double
rate_converse(int n, double d, double e, int a)
{
    double bound;
    double step;
    double log_m;

    log_m = 0.5 * n * log2(a);
    for (step = 0.1; step > 0.00001; step /= 10){
        while (e < (bound = bound_converse(n, d, log_m / log2(a), a))){

```

```

        log_m -= step * n;
    }
    log_m += step * n;
}
return(log_m / n);
}

int
main(){
    int n;
    \\消失確率の値を入力
    double d = 0.5;
    \\復号誤り確率の値を入力
    double e = 1e-3;
    double r;
    \\アルファベットサイズの値を入力
    int a = 2;

    for (n = 1; n < 2001; n++){
        r = rate_converse(n, d, e, a);
        if (r > 0.0){
            printf("%d,%g\n", n, r);
        }
    }

    return(0);
}

```