

Adaptive Playout Buffer Algorithm for Enhancing Perceived Quality of Streaming Applications

ユーザ品質を考慮したストリーミング
アプリケーションのためのプレイアウト機構

大阪大学 大学院基礎工学研究科
村田研究室
藤本 康平

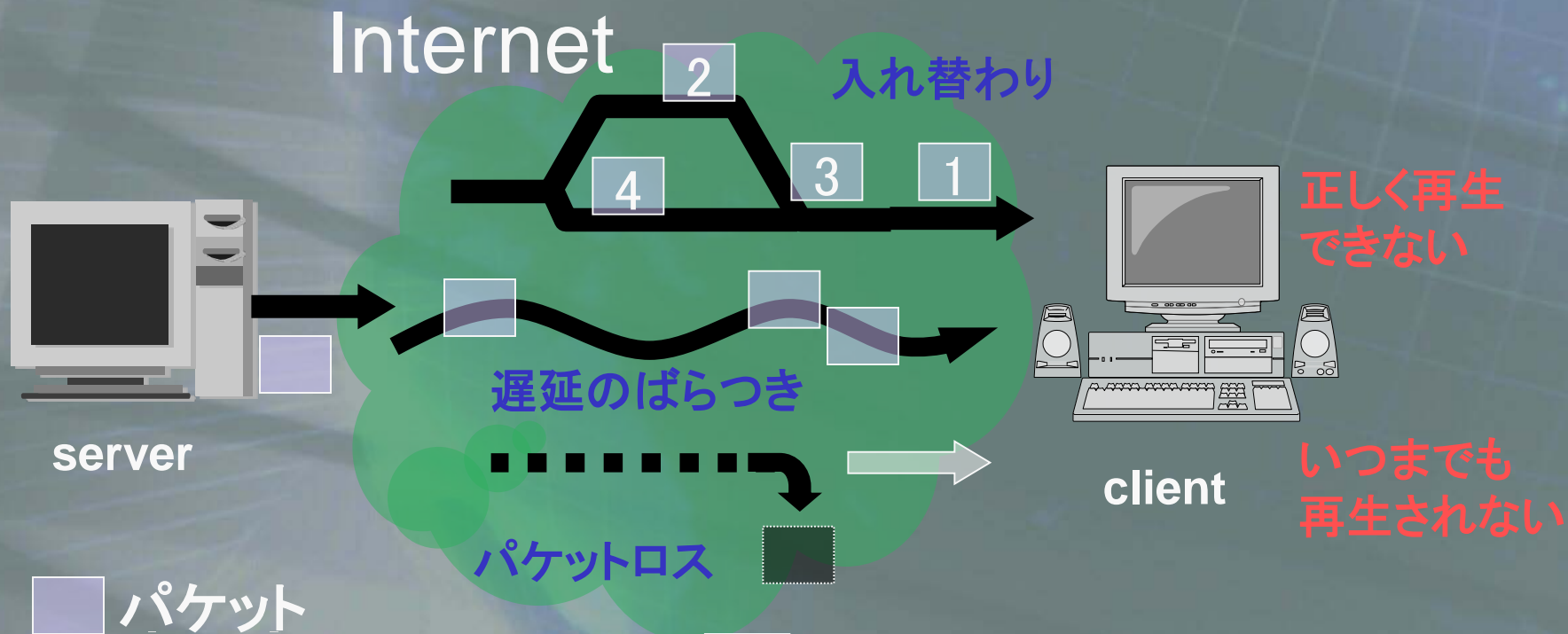
ストリーミングアプリケーション



- データをダウンロードしながら、到着したデータから直ちに再生
 - IPフォン,
音声会議
(vat, NeVoT..)
 - インターネットラジオ
(Winamp, RealPlayer..)
 - etc

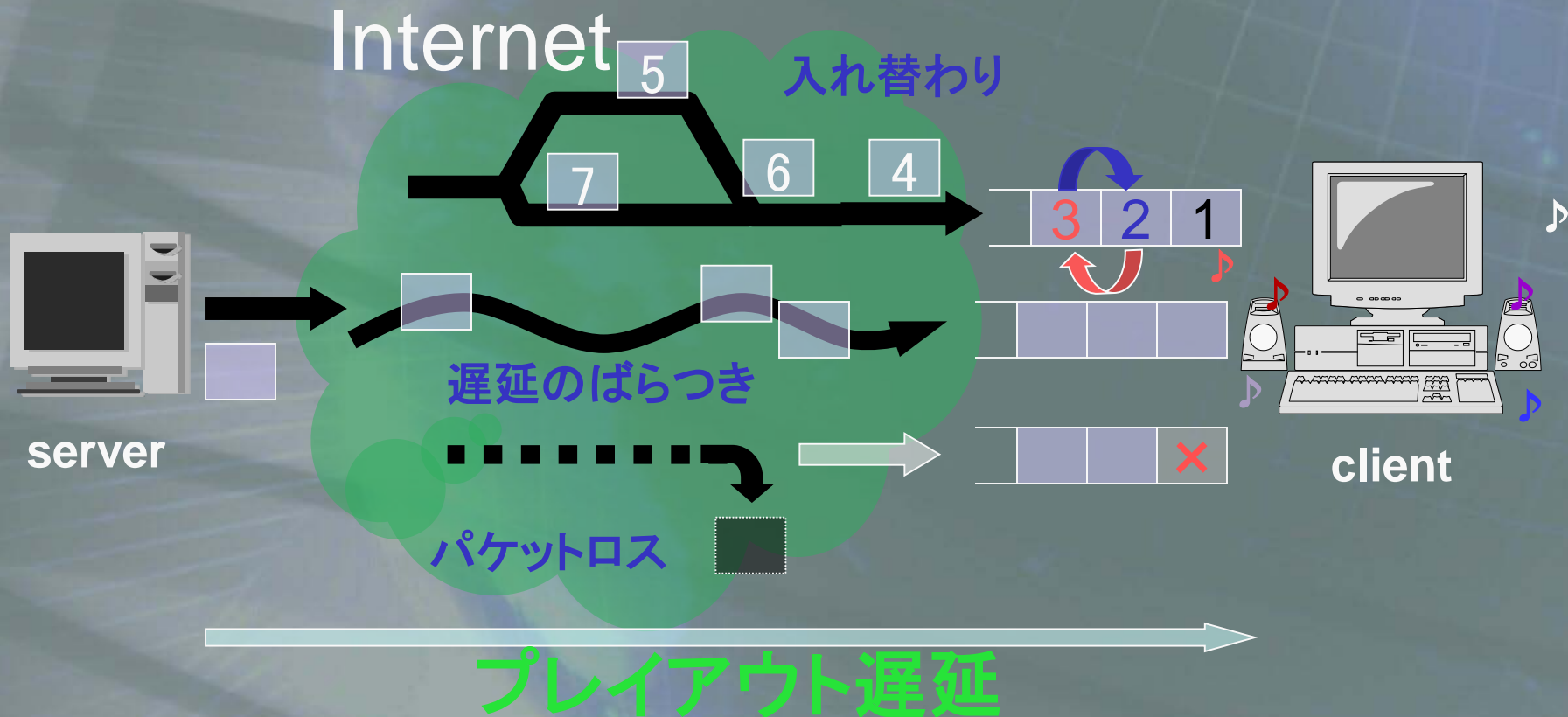
遅延に関する問題

データの転送遅延が保証されていない



プレイアウトバッファアルゴリズム

プレイアウトバッファアルゴリズム (PBA)



➡ オーディオデータの再生を安定させる

プレイアウト遅延

- PBA の重要なパラメータ

- プレイアウト遅延が短い場合

ロスしたとみなすパケットが増加するため音質は低くなるが到着したパケットは短い待ち時間で再生される

- プレイアウト遅延が長い場合

実時間性は下がるが、再生できるパケットは増える

プレイアウト遅延と再生される音質の間には
トレードオフの関係がある




トレードオフの関係を統計的に分析し、適切な値に設定することにより品質を保証

既存のPBA

- パケットロス率は 0% にする
- プレイアウト遅延は可能な限り小さくなるようにする


約 5% のロスが生じた音声でも
ユーザは我慢できる

 ユーザの要求するパケットロス率を
満たしつつ、プレイアウト遅延を最小化

アプリケーションの品質の考察

- 既存の PBA では、ネットワークパラメータを制御する

ネットワークパラメータとユーザの感じる品質の関係については考えていない

 ユーザの感じる品質を最適にする制御も必要

研究の目的

- パケット転送遅延の特性を明らかにし、パケットロス率(音質)とプレイアウト遅延(実時間性)の関係をモデル化する
- 遅延モデルを利用して、ユーザの期待するロス率を満たす PBA を提案する
- ユーザの感じる品質が最も高くなるようなプレイアウト遅延の制御を行う PBA を提案する (Enhanced MOS PBA)

Enhanced MOS PBA (E-MOS)

ユーザの感じる品質が最も高くなるような
プレイアウト制御を行う

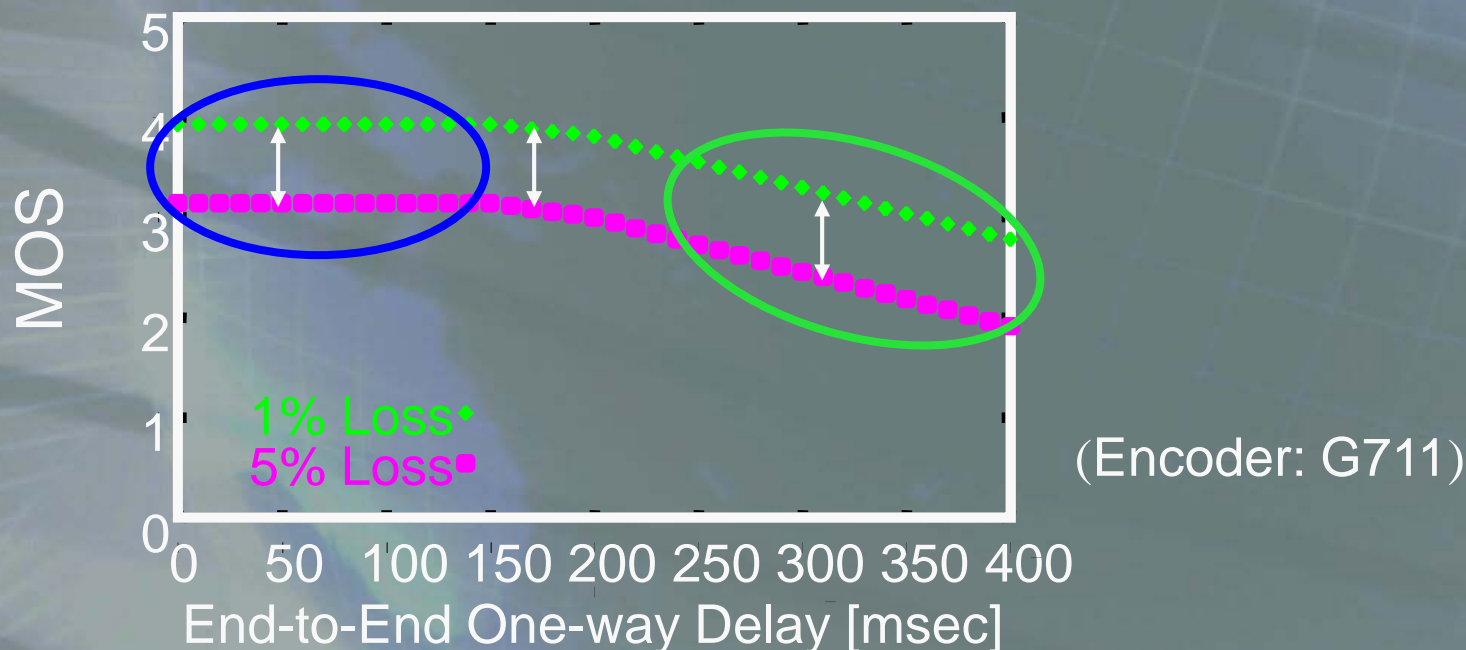
→ ネットワークパラメータと
MOS の関係を明らかにする

→ 現在のネットワークの状況において
実現可能な遅延とロス率の関係を
明らかにする

→ MOS を最高にするプレイアウト遅延の
算出方法を明らかにする

ネットワークパラメータと MOS の関係

- VoIP において、エンコーダ、遅延、ロス率に対する MOS の調査を行った文献を参照

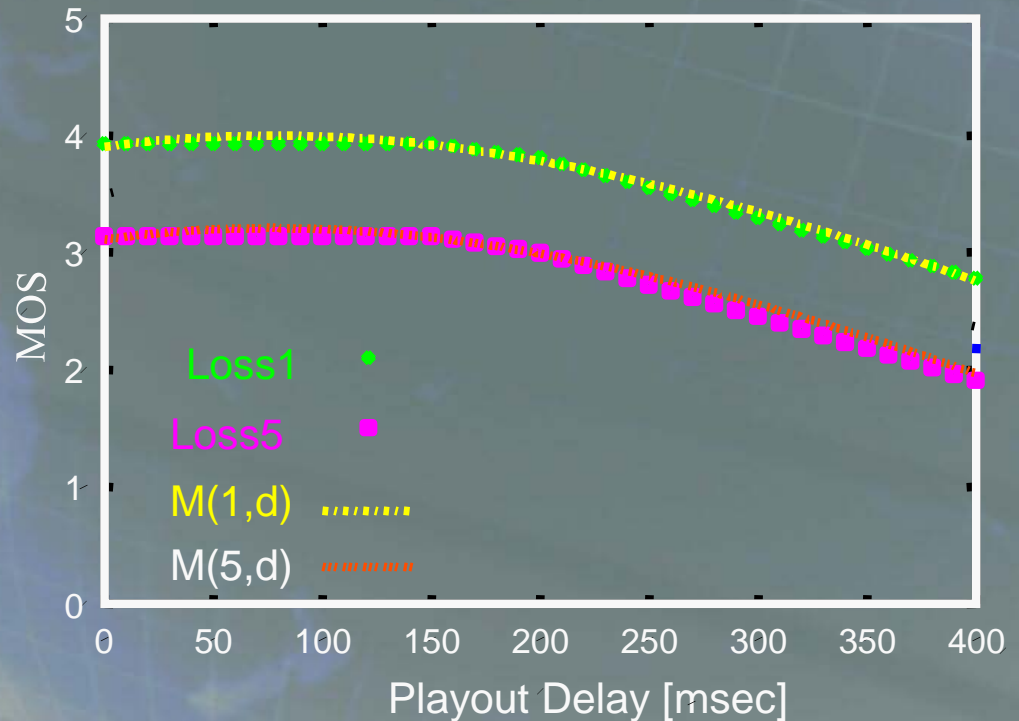


[11] C. Savolaine, "QoS/VoIP Overview," in IEEE Communications Quality & Reliability (CQR 2001) International Workshop, April, 2001

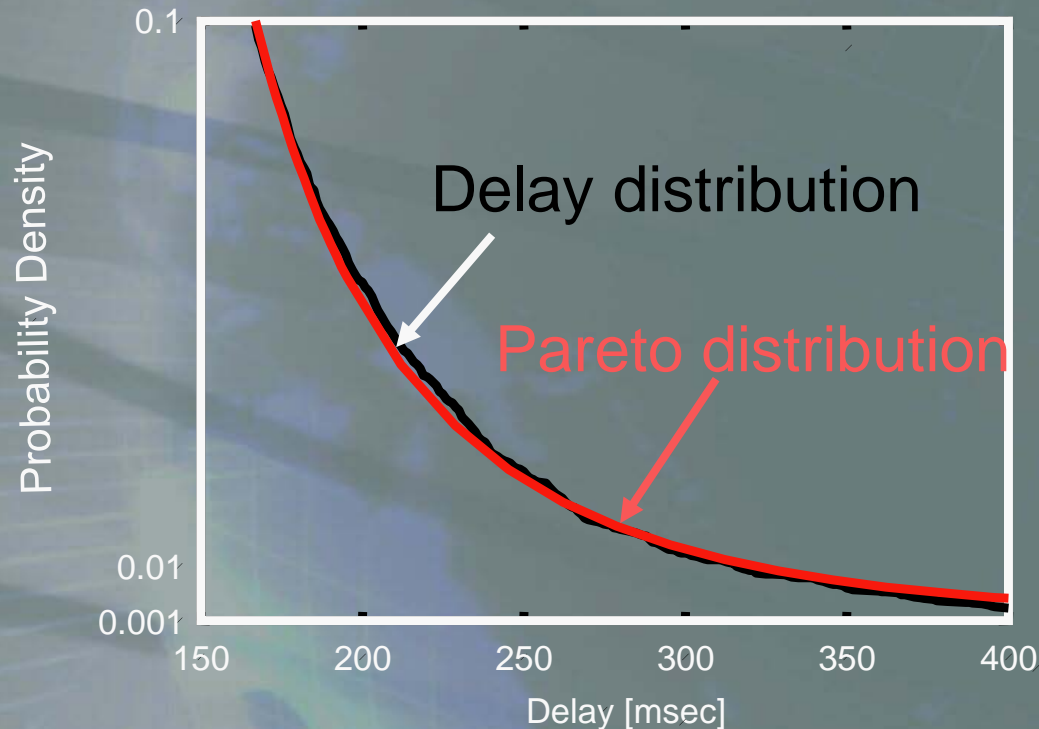
MOS 関数

- MOS の分布より近似曲線を算出し、MOS 関数を作成する

$$M(p, d) = 4.10 - 0.195p + 2.64 \times 10^{-3}d - 1.86 \times 10^{-5}d^2 + 1.22 \times 10^{-8}d^3$$



遅延分布のモデル化



遅延の確率分布をパレート分布でモデル化
プレイヤー遅延を x msec に設定すると、
→ $y \times 100$ % のパケットの再生が推定できる

MOS 関数のパラメータ

- 遅延モデルと MOS 関数より、MOS が最高となる遅延を推定

- 到着したパケットの遅延モデル (パレート分布)
- 実際のパケットロス率 (p_n : ネットワークにおけるロス)

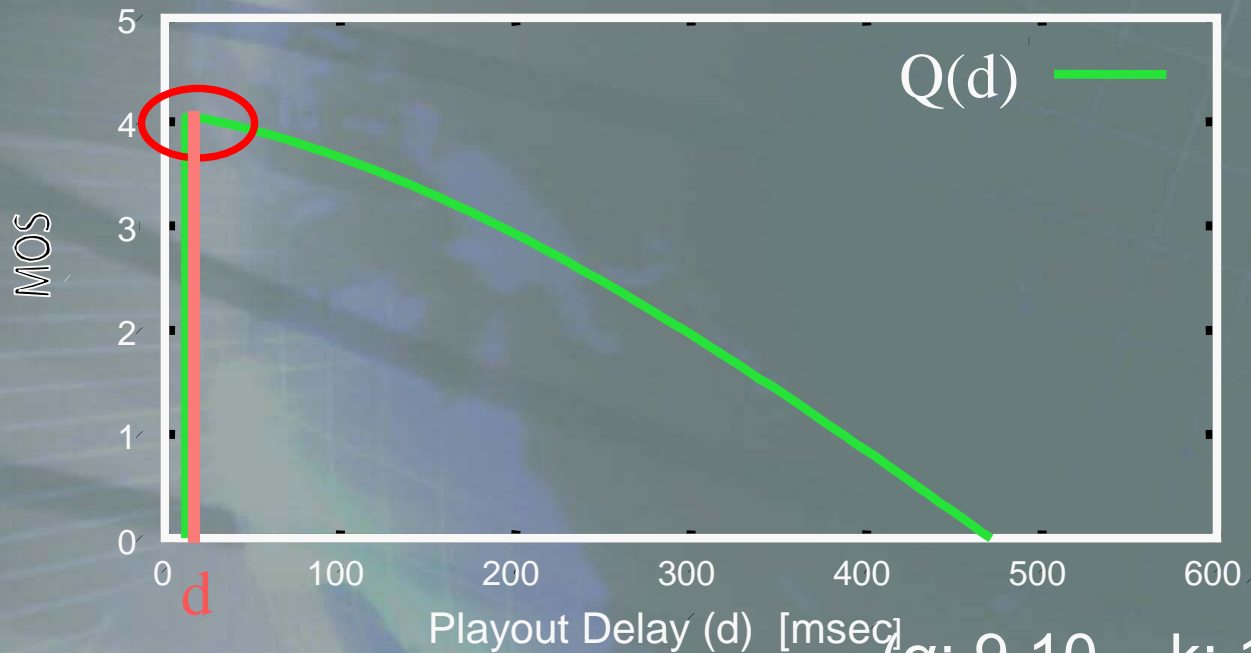
$$p_d = 100 \left(1 - \frac{k}{d} \right)^\alpha \quad (1) \quad p = p_n + p_d \quad (2)$$

- MOS関数への代入 (p_n は外部要因であるため一定とする)

$$M(p, d) =$$

$$Q(d) = 4.10 - 0.195 \left(p_n + 100 \left(\frac{k}{d} \right)^\alpha \right) + 2.64 \times 10^{-3} d - 1.86 \times 10^{-5} d^2 + 1.22 \times 10^{-8} d^3 \quad (3)$$

MOS 関数の最大値の算出



- $Q'(d)$ よりにはさみうち法を用いて、最大値をとる d を算出

E-MOS アルゴリズム

ユーザの感じる品質 (MOS) が最適になるよう
プレイアウト遅延を制御する

1. 到着したパケットの遅延を記録
2. 記録した過去 n 個の遅延値より、パレート分布に必要なパラメータ (α, k) を算出
3. $\alpha, k, Q(d)$ より、MOS 値が最高になるプレイアウト遅延の値を算出

性能評価1

アルゴリズム	目標再生率 [%]	パケットロス率 [%]	平均プレイアウト遅延[msec]	MOS値
Loss-Control	95	5.7	227.92	2.22
	99	0.94	387.12	2.41
	99.9	0.12	770.44	0.59
Exp-Avg	-	4.54	247.91	2.38
F-Exp-Avg	-	0.1	970.34	0.10
SPD	-	5.44	198.74	2.33
Window	99	1.34	362.57	2.47
E-MOS	-	2.95	294.75	2.49

性能評価2

アルゴリズム	目標再生率 [%]	パケットロス率 [%]	平均プレイアウト遅延[msec]	MOS値
Loss-Control	95	6.02	40.61	2.99
	99	1.77	58.45	3.83
	99.9	0.60	375.28	3.61
Exp-Avg	-	4.93	39.79	3.21
F-Exp-Avg	-	0.04	102.26	4.13
SPD	-	3.08	39.74	3.57
Window	99	2.33	48.60	3.72
E-MOS	-	0.10	77,71	4.17

実装実験

- E-MOS の計算量、実現性を調べる



実装結果

アルゴリズム	パケット ロス率 [%]	平均プレイ アウト遅延 [msec]	MOS値
Exp-Avg	4.52	35.23	3.45 
E-MOS	0.50	63.14	4.01 

E-MOS は高い品質をユーザに提供できる

まとめ

- パレート分布による遅延分布のモデル化
- 再生率を制御可能なプレイアウトバッファアルゴリズムを提案
- ユーザの感じる品質を最適化するプレイアウトバッファアルゴリズムの提案
- シミュレーション、実装結果による提案アルゴリズムの有効性の確認