



# HTTP proxy サーバにおける 動的コネクション管理方式

大阪大学 大学院基礎工学研究科

岡本 卓也

[tak-okmt@ics.es.osaka-u.ac.jp](mailto:tak-okmt@ics.es.osaka-u.ac.jp)



# 背景

- エンドホストにおけるデータ転送処理の高速化
  - SSBT 方式の提案
    - Webサーバにおけるサーバ資源の管理
    - データ転送処理速度の向上
    - 応答時間の削減
- proxy サーバを介した HTTP アクセス
  - 全体の35%程度
  - proxy サーバの処理能力の不足によるスループットの低下



proxy サーバにおけるデータ転送処理の  
高速・高機能化の検討が必要



# エンドホストにおける問題点

- ソケットバッファの割り当て
  - 各 TCP コネクションの帯域, 遅延等は異なる
  - 固定サイズのソケットバッファの割り当て
    - 各 TCP コネクションの帯域の考慮
- コネクション管理
  - 資源の割り当て
    - mbuf, ファイルディスクリプタ, メモリ空間
  - 資源の不足
    - 新規にTCP コネクションの確立の拒否

特に多くの TCP コネクションを収容する proxy サーバでは  
これらのサーバ資源の効率的な割り当てが必要



# 提案方式

1. ソケットバッファ管理方式 (E<sup>2</sup>-ATBT 方式)  
E-ATBT 方式を改良したもの
  - ・ 受信側ソケットバッファの考慮
  - ・ TCP コネクション間の依存関係の考慮
2. コネクション管理方式
  - ・ サーバ資源の管理
  - ・ persistent TCP コネクションの管理



# ソケットバッファ管理方式

- E-ATBT (Equation-based Automatic TCP Buffer Tuning) 方式
  - 各 TCP コネクションのスループットの推測
  - そのスループットを基に各 TCP コネクションが必要としているソケットバッファサイズの決定
  - 決定した大きさのソケットバッファの割り当て
- proxy サーバの特性
  - TCP コネクション間の依存関係
  - 受信側ソケットバッファの制御

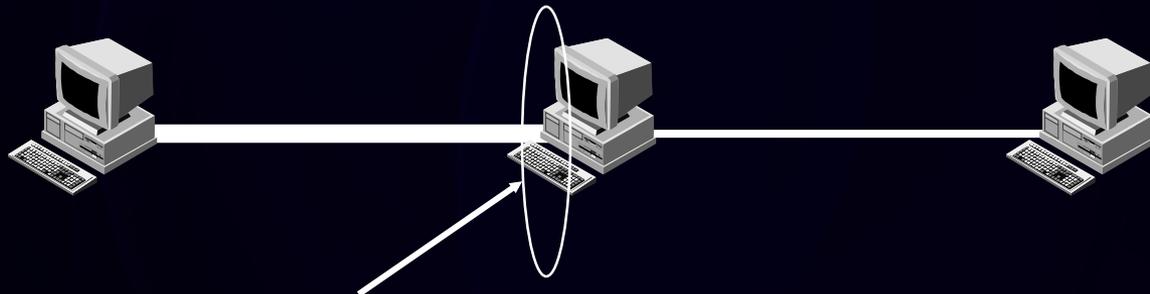


# E<sup>2</sup>-ATBT 方式

## - コネクション間の依存関係 -

- 依存関係とは  
TCP コネクション間のスループットの違い

client host                      proxy server                      web server



大きいサイズの  
ソケットバッファの割り当て

→ 無駄になる

↓  
クライアント間のTCPコネクションに  
割り当てるバッファサイズを減らす



# E<sup>2</sup>-ATBT 方式

## - 受信側ソケットバッファ -

- proxy サーバの特徴
  - 受信側ホストとして振る舞い
- 受信側ソケットバッファの割り当て
  - 受信側ソケットバッファの不足によるスループットの低下
  - 受信側ソケットバッファの動的な割り当てが必要



受信側ソケットバッファサイズを  
送信側ウィンドウサイズ以上にする



# コネクション管理方式

- persistent TCP コネクション
  - 転送終了後一定時間接続の保持
  - ウィンドウサイズなどのネットワーク情報の再利用
  - 3-way handshake を行わない
  - サーバ資源の割り当て
    - 割り当てられたサーバ資源が無駄になる可能性がある
    - サーバ資源の不足による新規 TCP コネクションの確立の拒否
- proxy サーバは多くの TCP コネクションを収容する
  - サーバ資源の無駄遣い
  - persistent TCP コネクションの切断



# コネクション管理方式

- proxy サーバの残存資源が十分ある時
  - 可能な限り persistent TCP コネクションを収容する
- proxy サーバの残存資源が少ない時
  - 使用されていない persistent TCP コネクションを切断し, 新規 TCP コネクションの確立を行う



この方式の実現には, 残存資源の監視  
および persistent TCP コネクションの管理が必要



# コネクション管理方式

## - 残存資源の監視 -

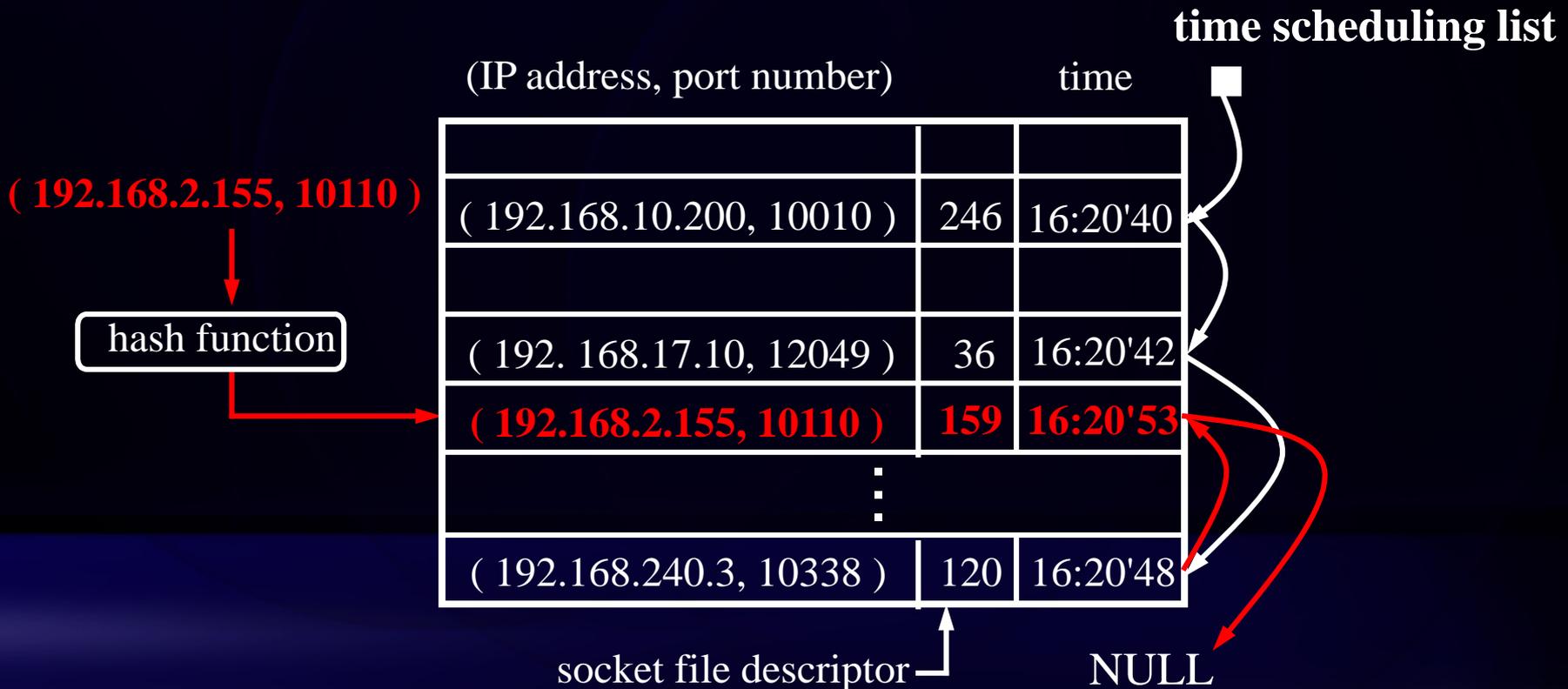
- システムコールによる、各サーバ資源の使用可能量、および現在使用している量の取得
- 各サーバ資源の閾値の設定
  - その閾値を越えた場合、proxy サーバの資源が少なくなったと判断



# コネクション管理方式

## - persistent TCP コネクションの管理 -

- persistent TCP コネクションの管理

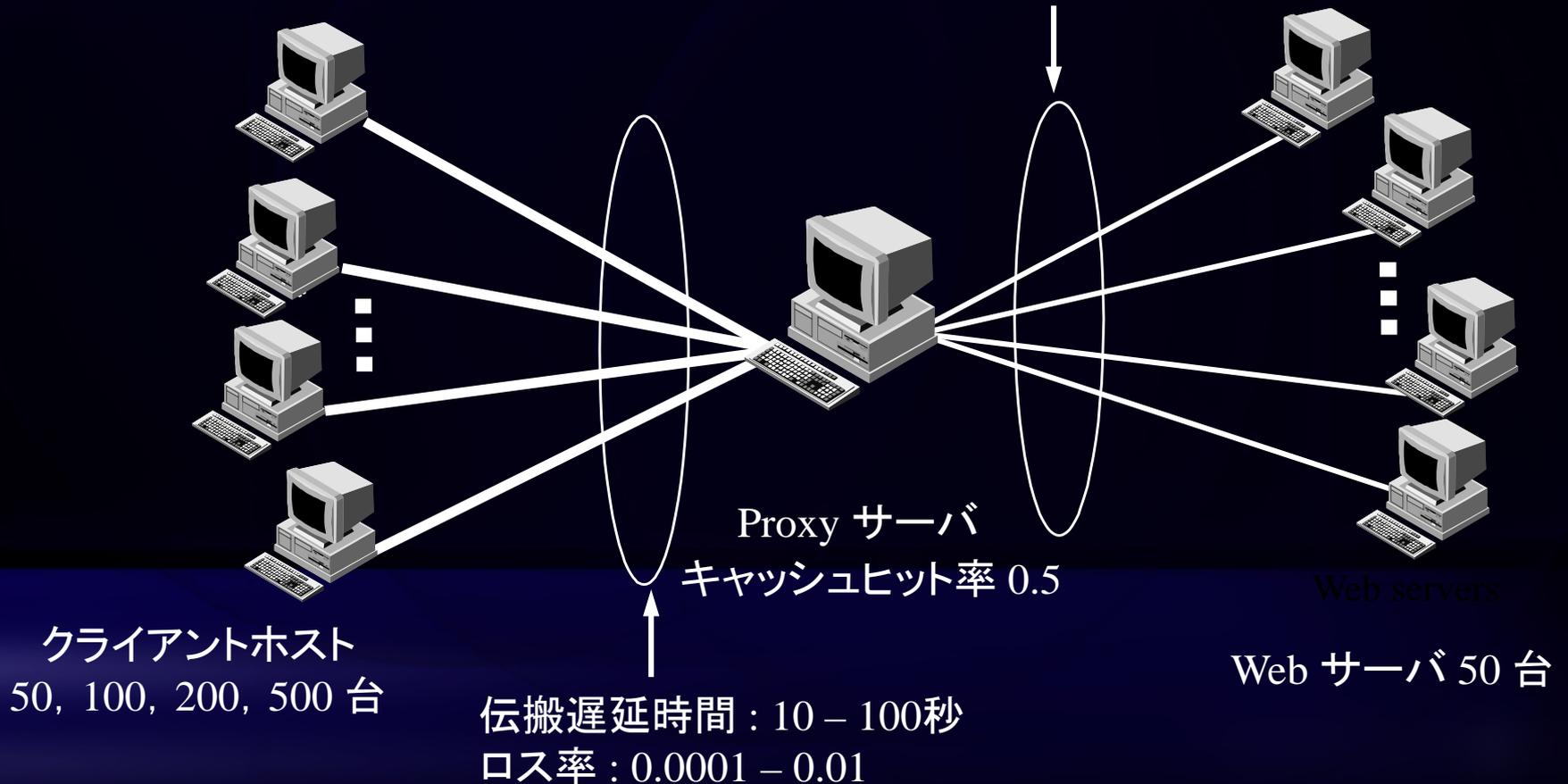




# シミュレーションによる評価

- シミュレーションモデル

伝搬遅延時間 : 10 – 200秒  
ロス率 : 0.0001 – 0.01



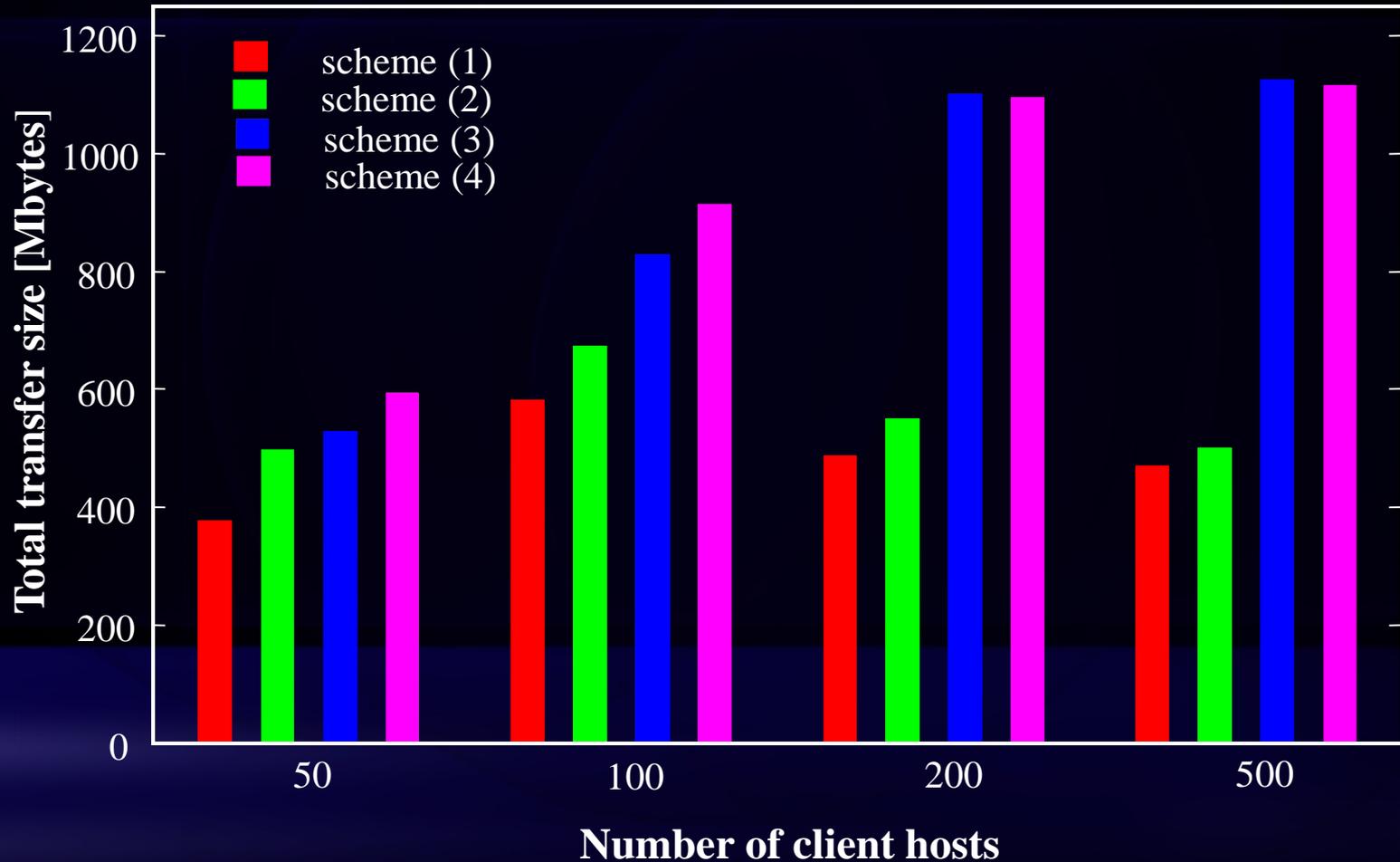


# 性能評価

- 性能評価を行った方式
  - scheme 1: 従来方式
  - scheme 2: ソケットバッファ管理方式
  - scheme 3: ソケットバッファ管理方式と  
コネクション管理方式
  - scheme 4: scheme 3 にさらにソケットバッファを  
徐々に減らす方式



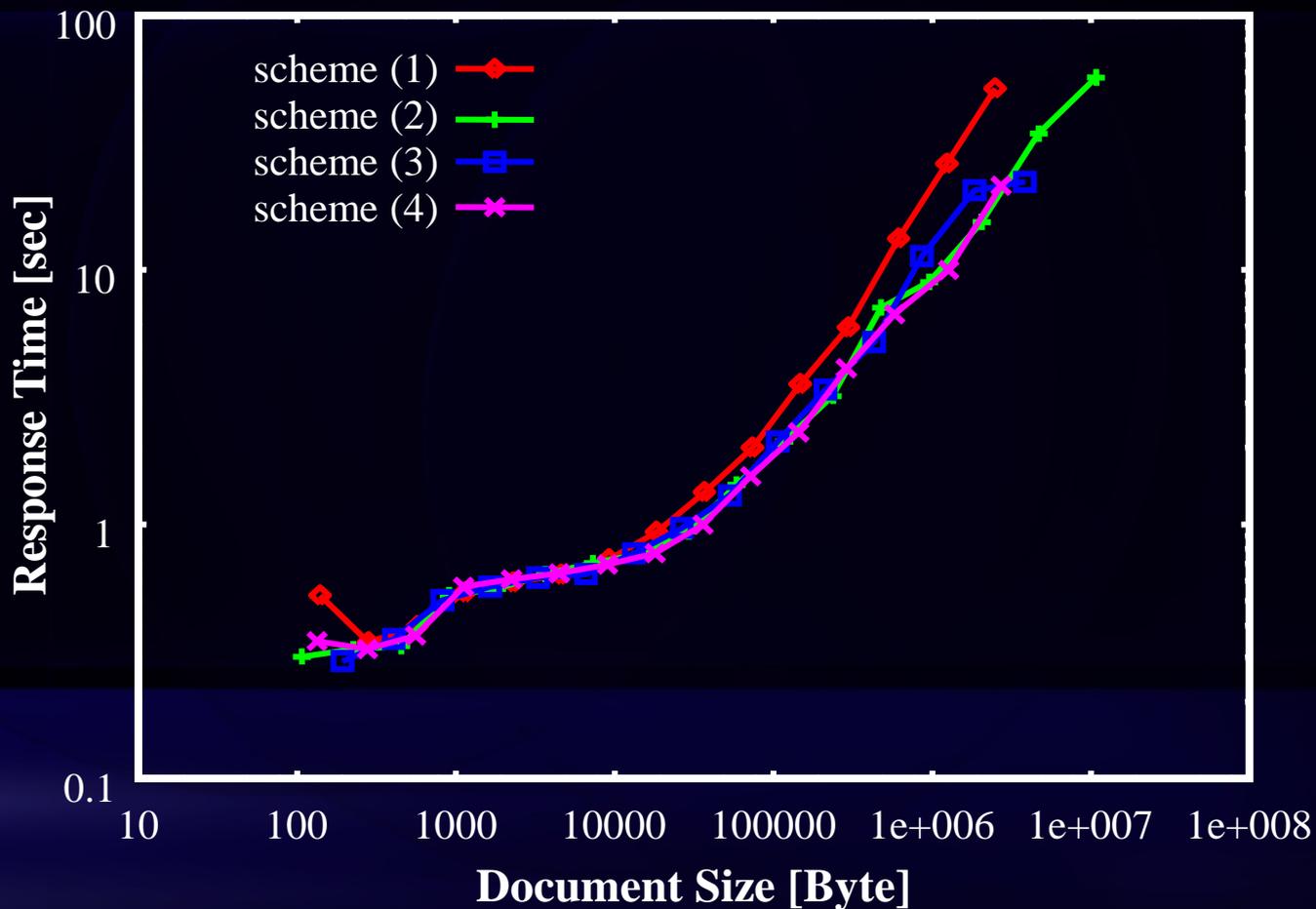
# proxy サーバでの性能評価





# クライアントの応答時間

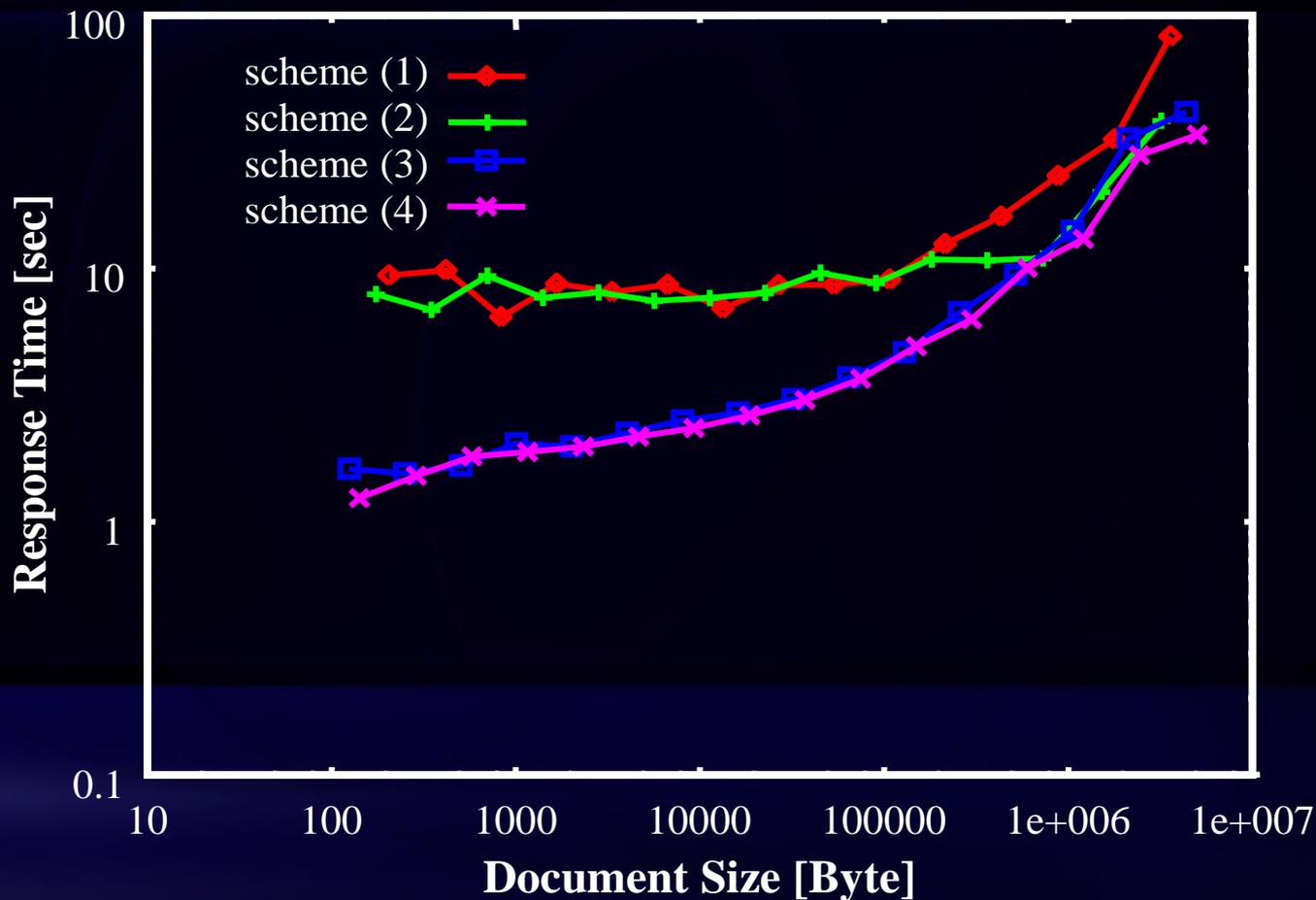
## - クライアント数 50 -





# クライアントの応答時間

## - クライアント数 200 -





## まとめと今後の課題

- proxy サーバにおけるサーバ資源管理方式の提案
- シミュレーションによる有効性の確認
  - ソケットバッファ管理方式
    - Proxy サーバの性能改善
  - コネクション管理方式
    - proxy サーバの性能改善
    - 応答時間の改善
- 今後の課題
  - 提案方式を実際の proxy サーバへの実装
  - その他のエンドホスト資源管理方式の検討