

インターネットコンファレンス'99



透過的クライアント監視による ネットワークシステム性能評価手法

齊藤 裕樹 中所 武司

明治大学大学院 理工学研究科 基礎理工学専攻 情報科学系

1999年12月15日



インターネットの普及と新たな要求

インターネット上での多様なサービスの展開

- ◆ WWWサイト数: 7,000,000 越
(1999年6月Network Wizards社調)
- ◆ エレクトロニックコマース
- ◆ 基幹業務がWWW技術に対応

インターネットアプリケーションの
QoS (Quality of Service) の提供

インターネットアプリケーションのQoS

■ サービスの品質確保：ユーザの利便性の観点

- ◆ 応答時間
- ◆ スループット
- ◆ エラー発生率
- ◆ 可用性



■ 管理者の性能評価の必要性

- ◆ ユーザの感じている性能をいかに知るか
 - ▶ 例：ユーザがWWWサイトにアクセスするのに何秒かかるのか？
- ◆ どのような性能指標をとらえシステムを設計するか
 - ▶ 例：ネットワークの帯域、サーバの処理能力

ユーザの利便性確保のための性能評価の難しさ

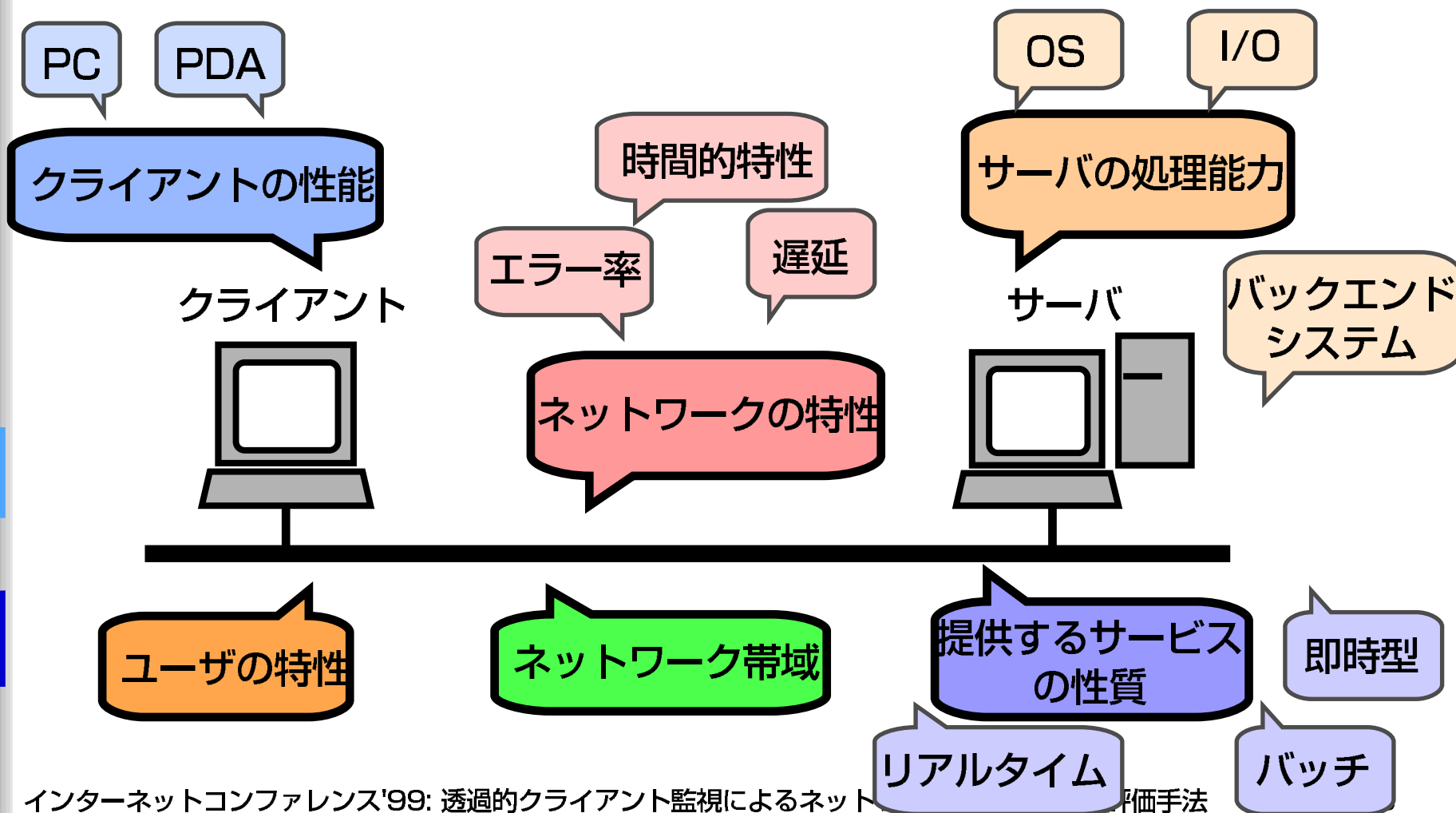
(1) 性能要因の多さ



ユーザの利便性確保のための性能評価の難しさ

(1) 性能要因の多さ

■ ネットワークアプリケーションの性能要因は多様





ユーザの利便性確保のための性能評価の難しさ

(2) 性能測定方法の欠如

■ 計算機上でのシミュレーション

- ◆ 性能を決定するパラメータが多い
- ◆ 全ての性能要因を計算機上でシミュレートするのは困難

■ ベンチマーク

- ◆ 稼働中のシステムは計測不可能
- ◆ 結果はベンチマーク条件でのみ有効
- ◆ 実際の利用状況と同様のベンチマーク条件の設定が困難



ユーザの利便性確保のための性能評価の難しさ (2) 性能測定方法の欠如 (続き)

■ 実測に基づく方式

◆ ネットワークの通信状態を監視

- ▶ サーバ、クライアントの性能は計測不可能
- ▶ ネットワークの混雑や遅延だけがアプリケーション性能の悪化要因ではない

◆ サーバのアクセス記録を解析

- ▶ サーバでの応答時間、スループットを測定
- ▶ クライアント、ネットワークの性能を計測できない



性能測定ツールの目標

対象：企業や大学内のワークグループ

目的：クライアントシステムでのユーザの感じる
性能を評価

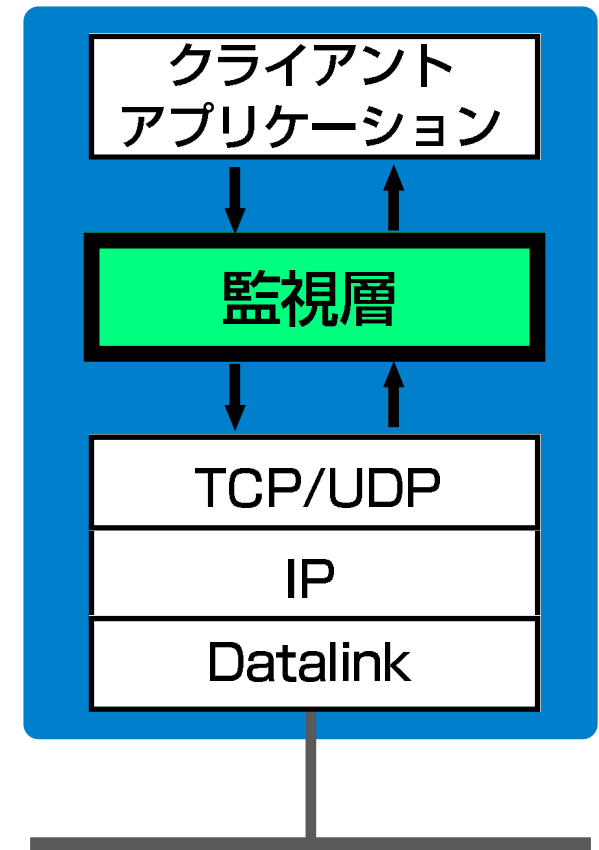
■ 要求機能

- ◆ 計測機能 所要時間、転送データ量
- ◆ 計測結果の分析 コネクション確立時間、データ転送時間、データ転送量、転送レート
- ◆ 統計処理 確率、頻度、分布などの分析
- ◆ 結果の可視化

監視層による性能測定

- クライアントアプリケーションを監視
 - ◆ アプリケーションとOS間のやりとりに注目
 - ◆ アプリケーションはネットワークへのアクセスの際システムコールを用いる

アプリケーションからの
システムコールを監視





監視層によって可能となる計測

- 監視層はクライアントアプリケーションの挙動を直接知ることができる
 - ◆ 監視層はアプリケーションにもっとも近いレイヤ
 - ◆ 下位層の影響を受けない

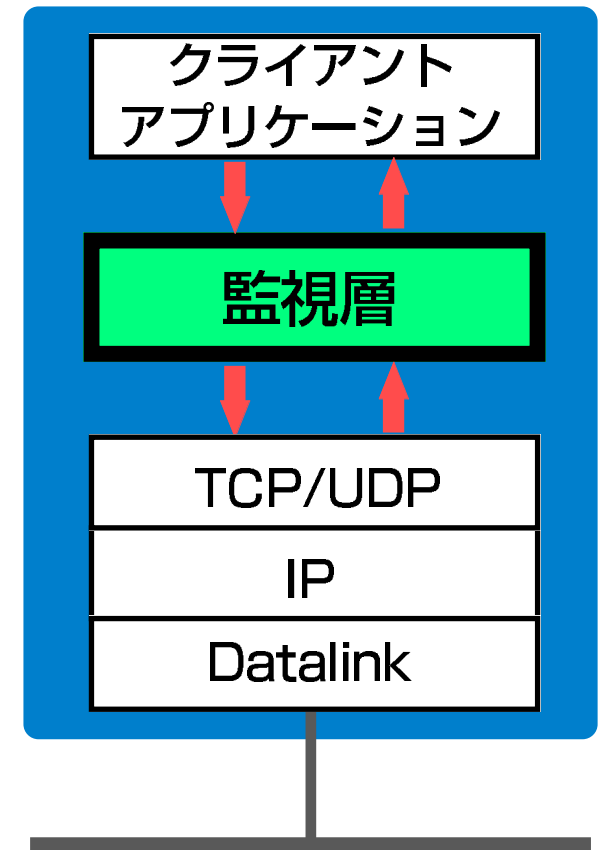
ユーザの感じる性能に最も近い計測が可能

- ◆ 応答性能
- ◆ スループット
- ◆ エラー率
- ◆ 可用性

監視層のインタフェース

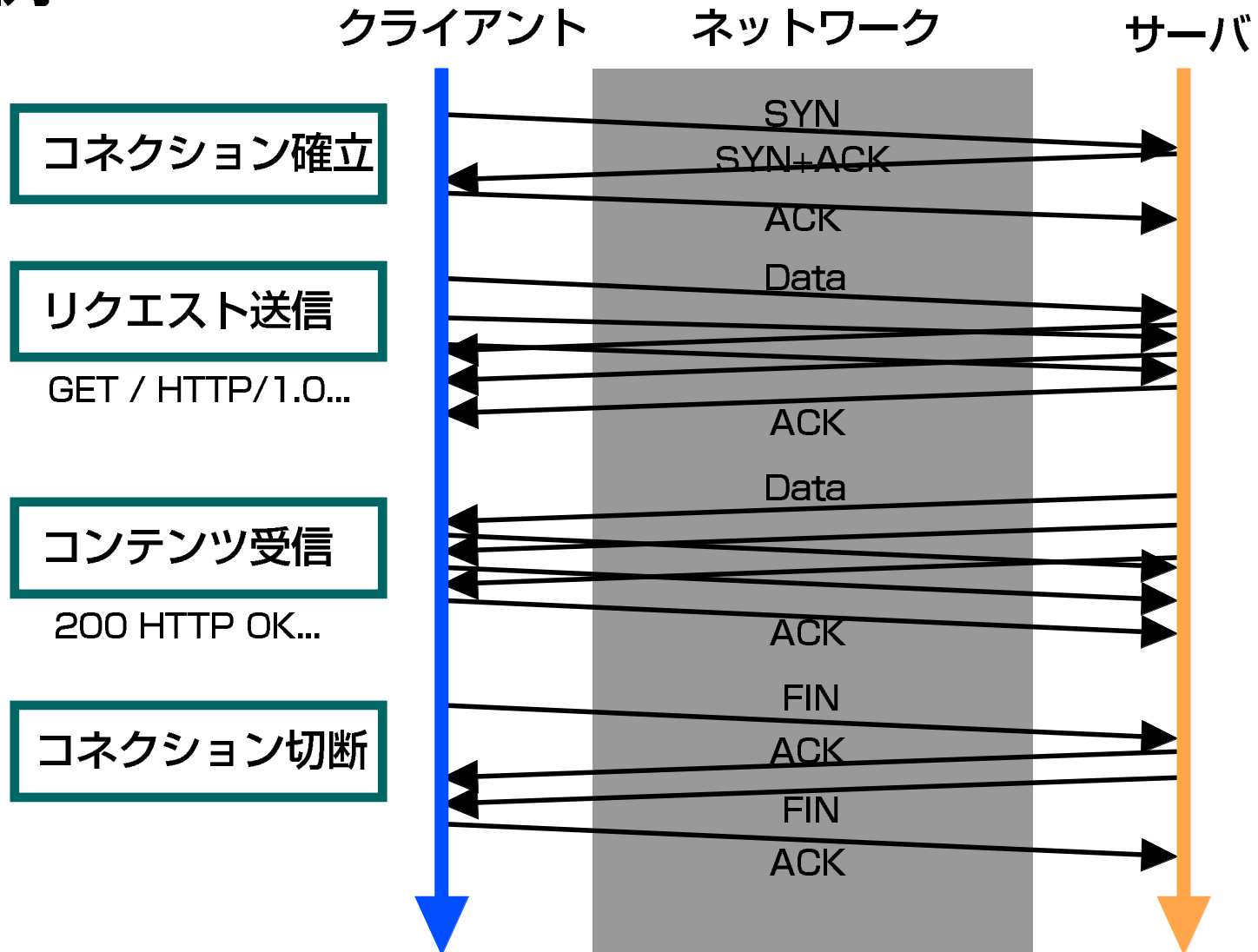
- OS標準のシステムコール、API と同一
 - ◆ BSD Socket, Transport Layer Interface(TLI), Winsock
- アプリケーションから見て透過な監視層

アプリケーション非依存
既存システムに改造が不要



監視層の対象とする通信モデル

■例：HTTP





要求応答単位の通信モデル

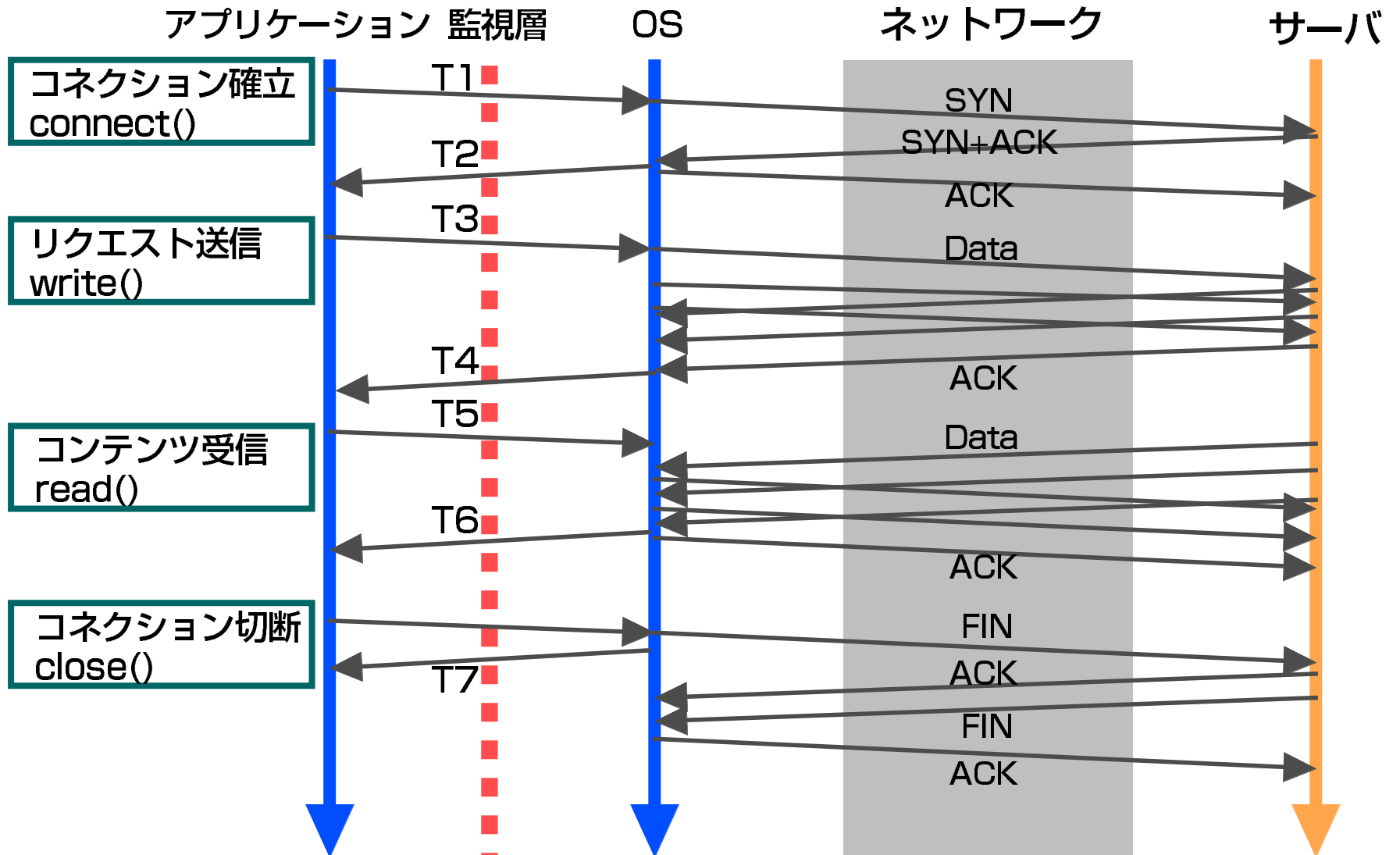
■ 「要求-応答」単位の通信モデル

1. コネクションの確立
2. リクエストの送信
3. 処理結果の受信
4. ……2,3のくり返し
5. コネクションの切断

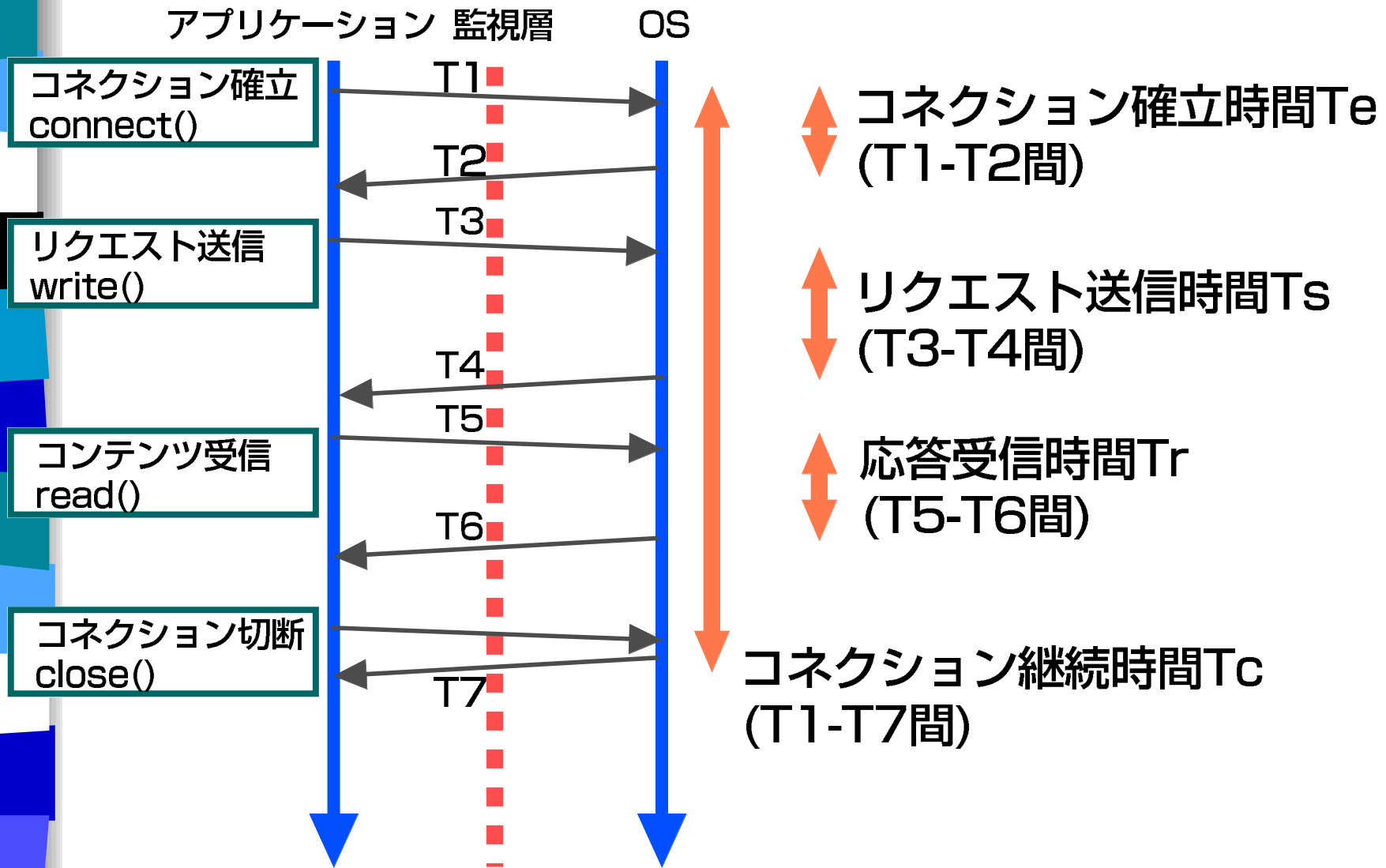
■ インターネット主要プロトコルに適用可能

- ◆ HTTP (WWW)
- ◆ POP3, SMTP (電子メール), etc.

要求-応答単位の通信モデルでの性能指標



性能指標（1）経過時間に関する指標





性能指標（2）データ転送量に関する指標

- リクエスト送信量(D_s)
- 応答受信量(D_r)
- リクエストデータ送信レート(R_s)
 - ◆ ($\text{リクエスト送信量 } D_s / \text{リクエスト送信時間 } T_s$)
- 応答受信レート(R_r)
 - ◆ ($\text{応答受信量 } D_r / \text{応答受信時間 } T_r$)



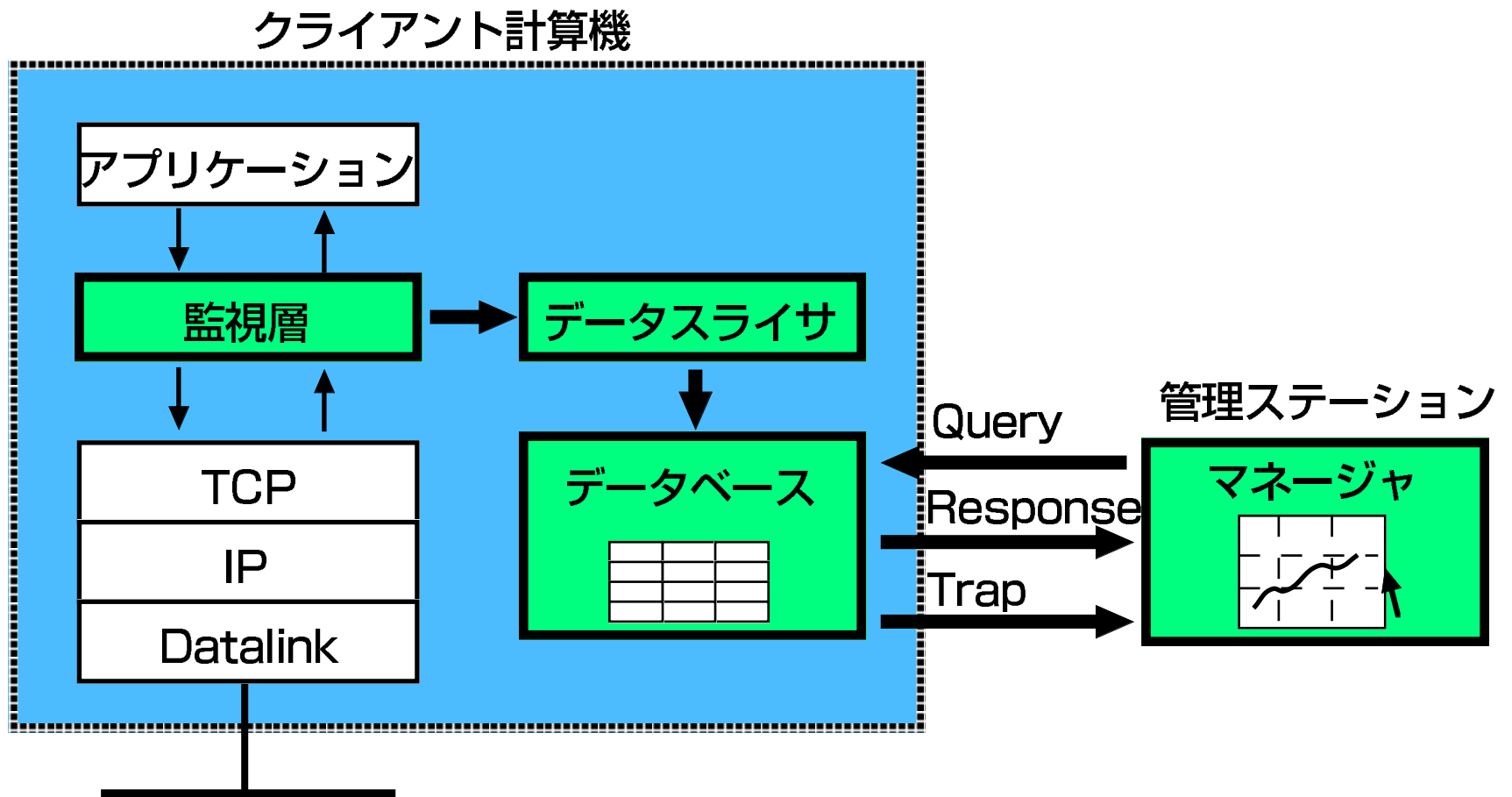
性能指標（3）転送率に関する指標

■ 全データ処理レート(Ra)

- ◆ $(\text{送信量}D_s + \text{受信量}D_r) / \text{コネクション継続時間}T_c$
- ◆ 通信に要した全所要時間に対する送受信したデータ量の割合
- ◆ 扱うデータの大きさに応じたユーザの感じる性能の指標

監視層による性能評価システム

■ システム構成



性能評価システムの構成

■ 監視層

- ◆ BSD Socket インタフェースを監視

■ データスライサ

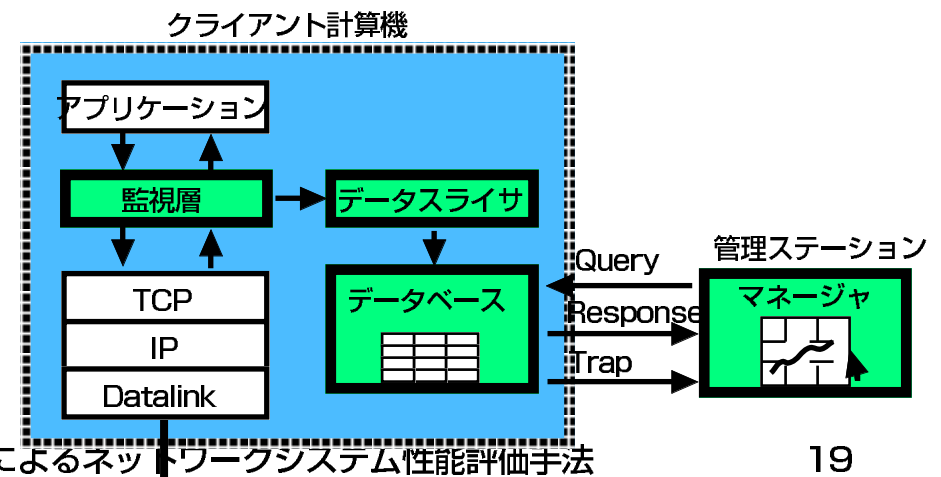
- ◆ IPアドレス、ポート番号などの条件に基づき測定データのフィルタリング

■ データベース

- ◆ (通信先, ポート, *T*, *D*, *R*) を時系列に並べる
- ◆ 2次元のマトリクス

■ マネージャ

- ◆ 遠隔管理ツール





実験ネットワークへの適用評価

実験 1: スケーラビリティの評価

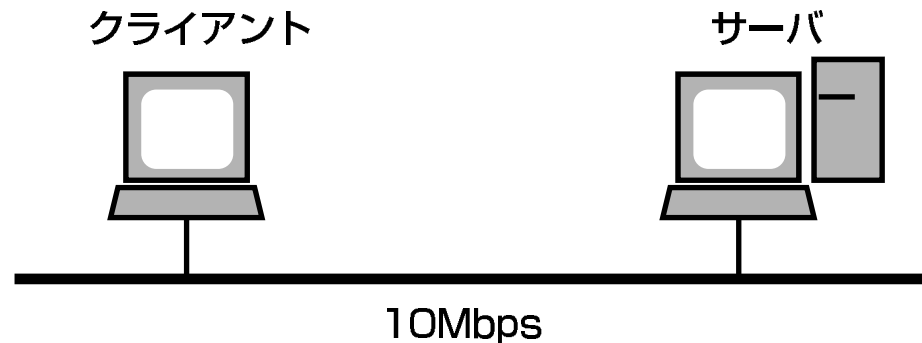
- 性能評価システムの適用範囲を検証
 - ◆ 監視を行うことによる性能劣化の有無
 - ◆ 計測時の時間計測、データ量のカウント処理が測定結果に与える影響を検証
- 監視層有無によるスループットの比較
 - ◆ 監視層における測定処理のオーバーヘッドを検証

実験1: スケーラビリティの評価

実験環境

■ 実験ネットワーク

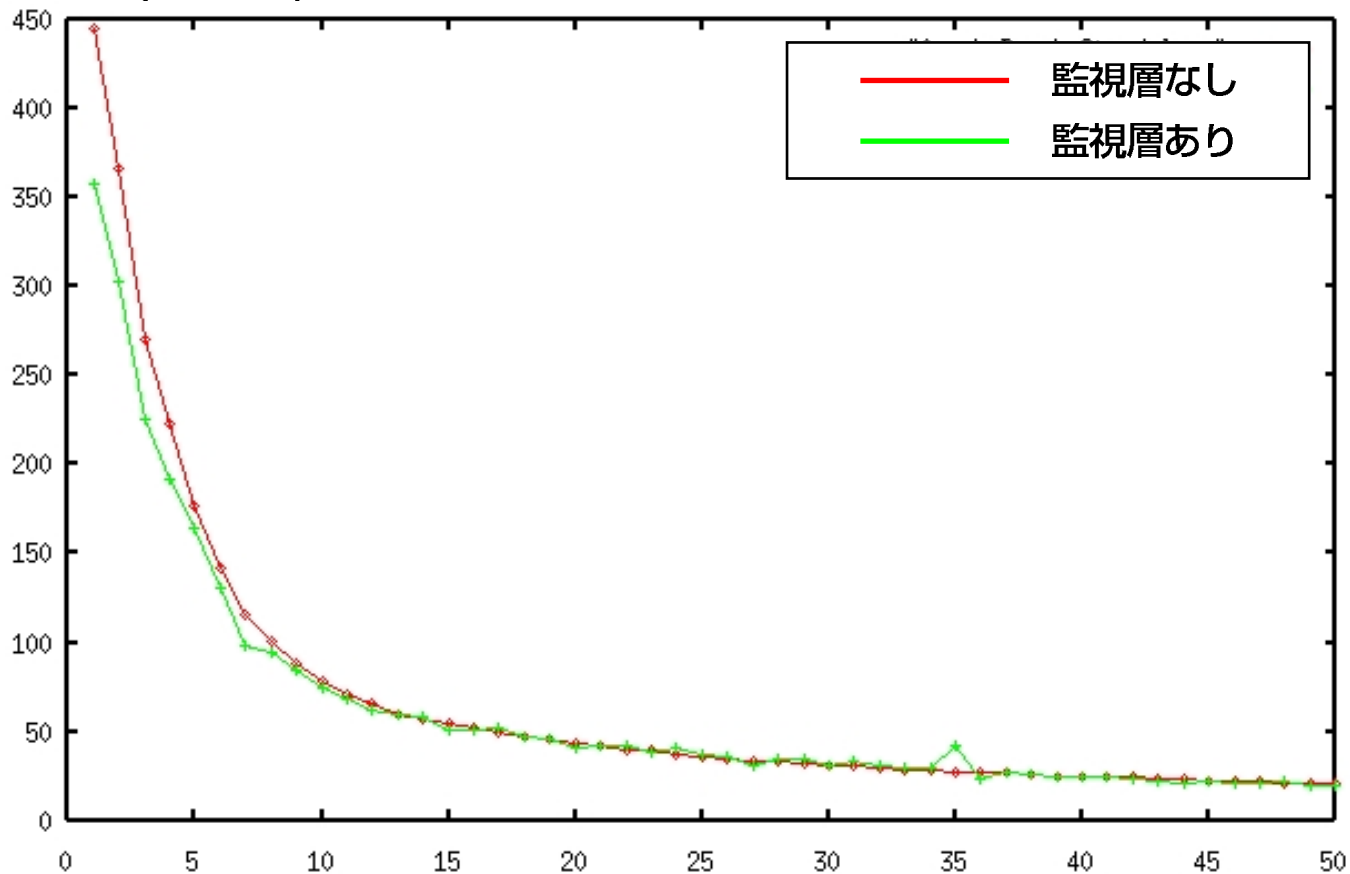
- ◆ 10Mbpsネットワーク上のサーバとクライアント
- ◆ アプリケーション: HTTP
- ◆ 100kbyteのデータ転送を反復



実験 1: スケーラビリティの評価 計測結果

データ転送量を変化させたときのスループットの比較

スループット (KB/sec)



インター

同時接続数 (回)



実験1: スケーラビリティの評価 結果の考察

■両者の差は最大10%程度

- ◆データ転送量に伴い悪化

■考察

- ◆大量のデータ転送を伴わない一般的なインターネットアプリケーションではオーバーヘッドは問題とならない
- ◆データ転送量の増大に対する計測機構の改良
 - ▶計測データのバッチ処理
 - ▶データスライサ、データベースの改良
 - ▶マルチスレッド化によるデータ転送と計測データ処理の並行処理

実験ネットワークへの適用評価

実験 2: 従来手法との比較

- クライアント側の計測値とサーバ側計測値の比較
 - ◆ 監視層でのコネクション継続時間 T_c
 - ◆ サーバ上アクセス記録の処理時間
- 実験ネットワーク
 - ◆ 1.5Mbpsネットワーク上のサーバとクライアント
 - ◆ アプリケーション: HTTP
 - ◆ 100kbyteのデータ転送を反復

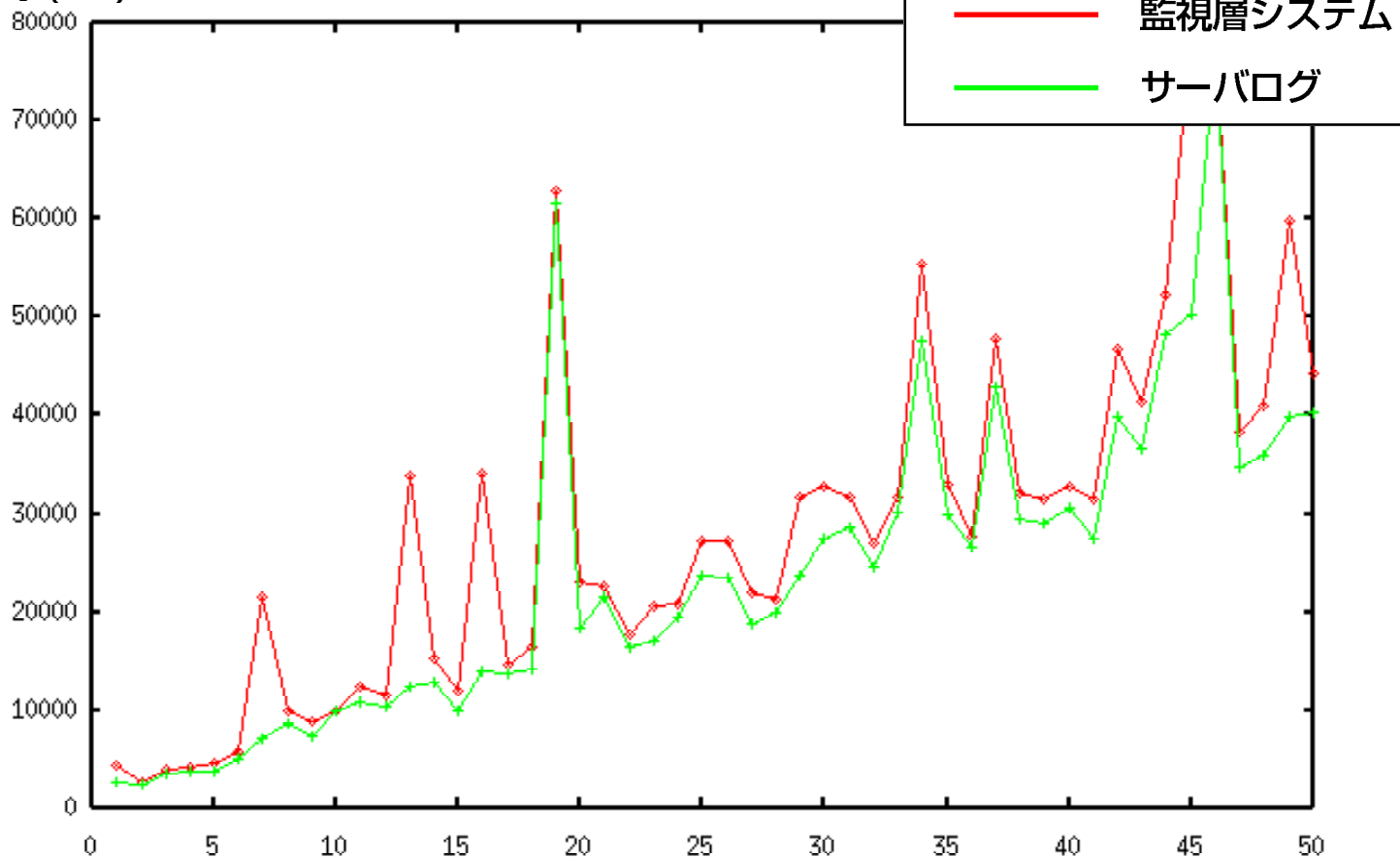


実験 2 : 従来手法との比較

計測結果

負荷を徐々に上げたときに計測された処理時間の比較

処理時間 (ms)



インター

同時接続数 (回)



実験2：従来手法との比較 結果の考察

- 両者の差は最大約28秒、平均約4.5秒
- 原因
 - ◆ 監視層：コネクション確立時間、終了時間が含まれる
 - ◆ サーバ側：コネクション制御に関する時間は計測不可能
 - ▶ コネクション確立・終了には少なくとも3RTTが必要
 - ◆ サーバ側：OSのバッファリングのため実際より短時間として計測される

サーバのアクセス記録では
性能評価に不十分



むすび

- クライアントアプリケーション監視によってネットワークシステムの性能を測定・評価する手法の提案
- 監視層によるネットワーク性能評価システムの実装
- 実験による評価
- 今後の予定
 - ◆ 実用アプリケーションの開発
 - ◆ システムを公開、インターネット上で広く評価を求める
 - ◆ 負荷に対するシステムの挙動の分析
 - ◆ 計測結果の精密な分析技法の検討
 - ◆ クライアント集中管理の枠組みの検討