





## データ系アプリケーションにおけるQoSとは？

- データ系は帯域を食い尽くすアプリケーション
  - アクセス回線、エンドホストの高速化
  - TCP (=エンドホスト) が輻輳制御を行う
  - バックボーンの高速化は解決策にならない
- データ系に適したQoS機構？
  - IntServによるQoS保証
    - パケット棄却率、パケット遅延を「保証」できるか？
    - トラフィック契約の考え方 (QoSパラメータ、トラフィック特性を事前に申告) とマッチしない
    - RSVPのスケーラビリティに対する限界
  - DiffServによるクラスに対するQoS差別化
    - 実現はHOL優先権制御で十分 (AFクラス)
    - 相対的なQoSはユーザのQoS要求とマッチするか？
  - QoS「保証」「差別化」なし
    - ただし、ネットワークプロビジョニングレベルでのQoS監視は重要
    - 帯域切り出し (VPN) は意味がある



## データ系アプリケーション QoSの3原則

1. Data applications try to use the bandwidth as much as possible.
2. Neither bandwidth nor delay guarantees should be expected. Only network provisioning can satisfy user's QoS requests.
3. Competed bandwidth should be fairly shared among active users.



## End-to-End Principle

- J. H. Saltzer, D. P. Reed, D. D. Clark, "End-To-End Arguments In System Design," ACM Transactions on Computer Systems, 1984.
- R. Bush and D. Meyer, "Some Internet Architectural Guidelines and Philosophy," RFC 3439, December 2002.
- "KISS: Keep it Simple, Stupid"
  - (1) ネットワークは特定のアプリケーションに基づいて、あるいは、特定のアプリケーションのサポートを目的として構築してはならない
  - (2) エンドノードで実現できる機能はそのノードに任せ、関係する状態情報はそのノードにおいてのみ維持すべきである
- 通信機能はできるだけエンドノードにおいて実現、ネットワークはビットを運ぶことに徹する



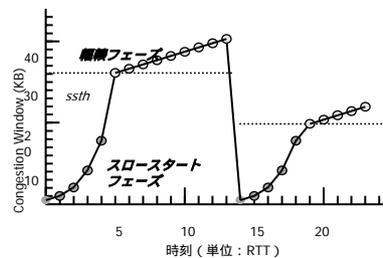
## TCPの特徴：成功の要因

- ネットワーク層以下に特定の技術を仮定しない
- Self-adaptability
  - エンドホストはACKを受け取ると、ウィンドウサイズを上げる
  - パケットロスがあると、ウィンドウサイズを下げる
- タイムアウト制御
  - ラウンドトリップ時間 (RTT) の計測  

$$RTT_{new} = RTT + (1 - \alpha) M$$
 M: 計測時間、 $\alpha$ : 重み(7/8)
  - 「ばらつき」の計測  

$$D = D + (1 - \alpha) |RTT - M|$$
  - タイムアウト時間の決定  

$$Timeout = RTT + 4 * D$$
- ネットワーク内輻輳制御を請け負う
  - ウィンドウサイズ可変型フロー制御
    - 単純な制御
      - ACKを受け取る ネットワークは空いている 転送速度を上げる
      - パケットロス ネットワークが混んでいる 転送速度を下げる





## 不公平性の要因: TCP Calculus

- TCPのウィンドウ制御
  - Self-clocking: RTTごとにウィンドウサイズを増やす
  - 帯域、距離の異なるコネクション間の長期的な不公平性
    - TCP Renoの場合

$$S = \frac{1.3}{RTT \times \sqrt{\rho}}$$

- TCP Reno + REDの場合: RTTにのみ反比例
- TCPのバージョン間の違い
- 実装の違い
  - タイマ処理
  - ウィンドウサイズの初期値
  - スループットに大きく影響する
- トランスポートプロトコルの違い: TCPとUDP間
- バグ、プログラムの改変によるもの: エンドツーエンド制御



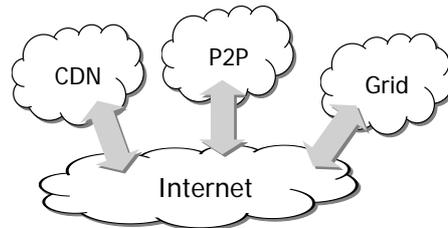
## 新しいアプリケーションの登場(1)

- 大量のデータ転送
  - 広域分散SAN (Storage Area Network)
    - 大量データのバックアップ
  - グリッドネットワーク (特にデータグリッド)
    - QoS要求「瞬時に損失なくペタバイト級のデータを送ることができる」
  - CDN
- 結局、QoSは帯域と回線容量との比 (多重度) で決まる
  - 多重度が十分にあれば、パケット交換 (+ 輻輳制御) で十分
  - 個々のエンドユーザのQoSを保証したければ、回線を渡してしまう (回線を使い切るだけの能力がなければムダ)
- アプローチ?
  - データ転送プロトコル (TCP) の高速化 HighSpeed TCP
  - エンドユーザ間回線の提供 フォトニックネットワーク (GMPLS)



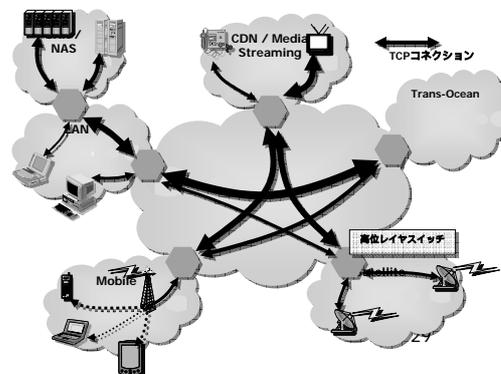
## 新しいアプリケーションの登場(2)

- オーバーレイネットワークの発展
  - P2P、CDN、Grid
- 特徴
  - GIDに基づく独自の資源発見機構
- 研究課題
  - 論理ネットワークと実ネットワークのインタラクション
  - 多重構造ネットワーク



## TCP オーバーレイネットワークによる高品質化

- TCPプロキシによるTCPコネクションの分割
  - 帯域遅延積相当のウィンドウサイズの確保
  - パケットロスへの早急な対処
- Deployment可能
- 分割する・しないによる品質差別化





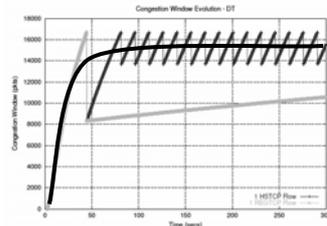
## インラインネットワーク計測

- 新しいトラフィック計測
  - パッシブ計測：ポイント計測
  - アクティブ計測：帯域浪費の問題
- TCPのデータ転送中にネットワーク状態を計測
  - CDNにおけるバックグラウンド転送
  - P2Pにおける適切なピア発見
  - TCPの適切なウィンドウサイズ設定；利用帯域の近傍まではアグレッシブにサイズを上げ、そのあとはゆっくり
  - エンドホスト自身がネットワーク適応性を持つ



1. ウィンドウサイズ分のパケットのバースト転送
2. ACKをそれぞれのパケットに対して返送
3. 返送されたACKの時間間隔を見て利用帯域の予測

M. Murata



## 新しいアプリケーションの登場(3)

- (モバイル) アドホックネットワーク、(モバイル) センサネットワーク
- 特徴
  - 独自のピア (通信相手) 発見機構：経路制御
    - AODV, DSR, OLSR, ...
  - システムの大規模化
    - ユーザ数、ノード (ホスト=ルータ) 数
    - オンデマンド資源発見 フラッドニング前提 スケーラビリティの問題
- 研究課題
  - 自律分散制御の必要性
    - Robustness, Resiliencyの確保
    - 非常に不安定な通信環境
    - IPはルータ間の制御メッセージ交換を前提にしている
  - 新しい品質メトリック：電力

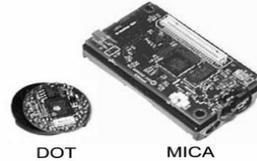
M. Murata

33



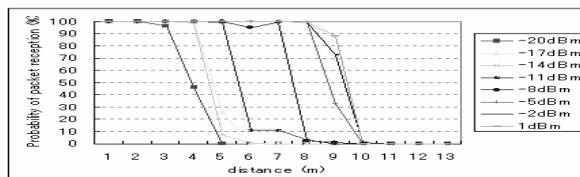
# センサネットワークの 実通信環境の測定

- クロスボー社のMOTE-2を用いた、実通信環境の測定
- 通信可能距離の環境による変化
  - パケット損失率が5%以下になる距離の実測
- 非対称リンクの影響
  - 片方向しか通信できない非対称リンクの発生状況の確認
- 実消費電力の測定
  - 送信、受信で実際に消費される電力とトラフィック量の関係
- 大阪大学情報科学科棟屋上での計測

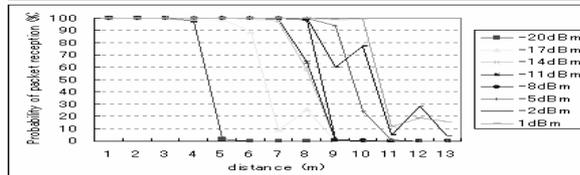


# 測定結果例 (1)

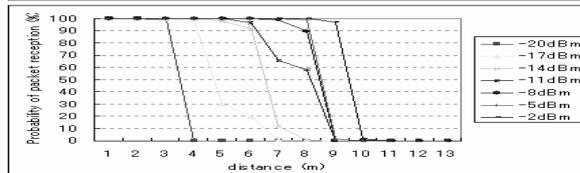
- 各距離で1000パケットを送信した時のパケット損失率



(A) 壁に平行



(B) 壁に垂直



(C) Aと逆



## ネットワーク制御に 求められる3要素

- 拡張性 (スケーラビリティ)
  - ルータ数やエンドホスト数、ユーザ数、情報機器端末数の増大への対応
- 多様性
  - 情報機器デバイスの多様性、ネットワーク技術の多様性、ネットワークサービスの多様性、トラフィックの多様性への対応
    - 単一のネットワークアーキテクチャによる統合ネットワークは存在しない
    - 不安定な通信環境
- 移動性
  - 利用者自身、ネットワーク資源 (ルータ、回線、サーバ) の移動
  - それらの生成・消滅が頻繁に発生



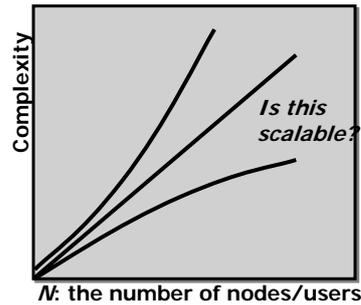
## 適応性

- 基本原則
  - エンドホストの適応性(adaptability)を根幹とし、ネットワークはそのような適応性をサポートするための機構を提供する
  - インターネットの分散処理志向をさらに推し進め、それによって損なわれる資源利用の効率性については、エンドホストの適応性によって補償する
  - 今後も開発されていく多様な通信技術に対応しながら、スケーラブルでかつ耐故障性に富んだネットワークを構築しつつ、ユーザの多様な要求に対するサービスを提供する
- エンドホストの自律性がますます要求されるようになり、それを前提として、ネットワーク全体の調和的な秩序を保つ



## スケーラビリティとは

- Metcal's law
  - "The value of a network increases exponentially with the number of nodes."
  - すべてのユーザが直接的に通信できる場合、ユーザ数 $N$ に対して、ネットワークの価値は  $V(N) \sim N^2$
  - Webシステムのクライアント/サーバモデルにより崩れつつあったMetcal's LawはP2Pネットワークの登場により、復活しつつある
    - P2Pネットワーク：パワー則の観察
    - Preferential Attachment
- $O(N)$ オーダのネットワーク制御はスケーラブルか？
  - $N$ がノード数であればおそらく答えはYES
  - しかし、ユーザ数（ホスト数）の場合はNO
  - エンド間通信を考えれば $O(N)$ が必須



## Complex Internet

- 複雑系としてのインターネットにおける、頑強性、安定性の確保
  - (フィードバック)制御：TCPそのもの
  - 冗長性：フォトニックネットワークのSelf-Healing
  - モジュール化：プロトコルの階層化
  - 構造安定：AS単位の階層化
- 研究課題
  - 現状の技術でインターネットはほんとうに安定しているのか、頑強なのか
  - 頑強な、また、安定な制御、ネットワーク構造？
  - 冗長性はどの程度、必要なのか？
- インターネットは設計を変更できる：巨大な実験場！



## 生物界に学ぶネットワーク制御

- 背景
    - 生物を複雑系として見た場合の、頑強性、安定性は次第に明らかになりつつある
      - 外乱に対する適応能力は高い
      - ただし、その適応速度は遅い
    - 生物の自己組織的、自立的な制御をネットワークに持ち込み、その適応性 (adaptability)、頑強性 (robustness)、安定性 (self-stability)、耐故障性 (resiliency) を利用できないか？
  - 過去の例
    - GA (Genetic Algorithm) : 遺伝子をモデル化し、それを最適問題の解法に適用
    - ACO (Ant Colony Optimization) : ありの生態を模した最適問題の解法
  - 文部科学省21世紀COEプログラム「ネットワーク共生環境を築く情報技術の創出」
- M. Murata 生物界の共生関係に学ぶ情報技術 (ネットワーク制御) の創出 45



## まとめ

- キーワード
  - 大規模化 スケーラビリティ
  - 多様性 複雑化 自律性、耐故障性
  - 移動性 自己組織化、自律性
  - End-to-End Principle
  - エンドホストの適応性
- ネットワークは (やはり) コネクティビティを第一のQoS目標にしつつ、自己組織化による自律制御、スケーラビリティを提供する
- エンドホストは「現時点」でのネットワーク状態に適応した資源発見、輻輳制御、フロー制御 (含ストリーミング制御) を実現する