

小口研究室 研究紹介 (2012年度)

(お茶の水女子大学理学部情報科学科)

モバイルネットワークにおける周辺端末からの情報に基づく協調制御ミドルウェアの提案と実装 (研究担当: 平井 弘実)

研究背景

端末が集中する場所では輻輳による通信性能の低下
 ・スマートフォン端末の急速な普及
 ・大容量高速通信に対する需要が増大

モバイルネットワークの特性
 ・低帯域無線アクセス
 ・APに繋がるノード数が増加しやすい
 ・ハードウェアリソースが限定

輻輳制御アルゴリズム
 ・輻輳領域を避けるよう送出するセグメント数を調節
 ・輻輳を避けながら、可用帯域を使い切れると望ましい

TCP Cubic

$$W_{cubic} = C \left(T - \sqrt{\frac{W_{max}P}{c}} \right)^3 + W_{max}$$

・時間的な変化が特徴
 ・輻輳ウィンドウは三次関数状増加

モバイル端末・サーバ間のTCP通信の高速化・公平化

周辺端末の情報を利用し、連携した輻輳制御
 ・APを共有する端末間で、通信状況に関する情報を通知
 ・周辺端末からの通知に基づき、自端末が使える帯域幅を計算

輻輳制御の補正
 ・輻輳ウィンドウの理想値を最大値を設定
 ネットワークの許容量を上回ることを抑制し可用帯域を各端末に均等に割り当てる
 有線区間でパケットが溢れた時は、柔軟に輻輳ウィンドウをスケール
 ・帯域幅遅延種
 ・理想的な輻輳ウィンドウ = セグメントあたりのデータ量 × 端末数

協調制御ミドルウェア

情報共有モジュール
 ・カーネル内部の処理を監視
 ・通信処理があれば通信状況に関する情報を取得し、ブロードキャストで通知

輻輳制御補正モジュール
 ・周辺端末からの情報通知パケットを受信し、状況に応じて輻輳制御を補正

評価実験

実験目的
 APを共有するアクティブな送信端末数の変化に応じて、輻輳制御を補正し、オーバーヘッドを含めた通信性能を測定し、ミドルウェアを評価

実験内容
 複数端末が、1台のAPを共有している際の通信スループットを測定
 周辺端末数に応じて、ミドルウェアが輻輳制御を補正
 5~8台で端末数を変化させた時のスループットを測定
 人工遅延時間: 25ms、通信方式: 802.11g

実験結果

ミドルウェアを導入している場合と導入していない場合で通信スループットを比較

実験1: 送信端末数が徐々に増える
 実験2: 送信端末数が徐々に減る

各端末の通信時間の時系列変化
 各端末の通信スループット

通信スループットが最大64%向上!

輻輳ウィンドウの時系列変化解析結果 (実験1)

ミドルウェアを導入しない各ホストの輻輳制御
 ・オーバーフローなどの通信エラーが、多数発生している

ミドルウェアを導入した各ホストの輻輳制御
 ・端末数が増えると最大値が下がる
 ・各端末の輻輳制御が協調
 輻輳ウィンドウが比較的安定
 通信エラー・再送が少ない

まとめ
 ・ミドルウェアの導入により、効率的な制御が可能となり、通信速度・公平性の向上を確認
 ・ミドルウェアが、アクティブな送信端末数の動的変化に対して、適切に対応することを確認

ライフログ解析アプリケーションにおけるデータ品質評価フレームワークを用いた入力データ品質評価 (研究担当: 山下 暁香)

研究背景

ライフログ解析アプリケーション
 ・利用されるデータの品質の考慮
 ・入力データの品質がライフログ解析アプリケーションの性質に及ぼす影響を定量的な指標で評価する

研究目的

入力
 出力

データ品質評価フレームワーク

言語化アプリケーション
 言語化
 言語化

データ品質
 A: データ取得コマ数 (単位時間当たりのデータ量)
 B: 動画データの画質
 C: 動画データ転送時の無線通信品質

C: 無線データ品質実験環境

言語化対象
 ・椅子に座る
 ・冷蔵庫の扉をあける
 ・本棚を開ける
 ・部屋のドアを開ける

評価結果2: 動画データ送信端末と背景端末のスループット特性

評価結果1: 動画データ送信端末のパケットロス率

評価結果3: 言語化アプリケーションの正答率

今後の課題: クラウド上における言語化アプリケーションの実行

無線LAN環境における周辺端末の情報に基づく輻輳ウィンドウサイズ設定手法の検討 (研究担当: 飯尾 明日香)

研究背景

車など無数の高性能センサが搭載された端末は、位置情報や加速度などの多様な周囲の環境情報を取得可能
 ・これらの端末はセンサで取得した環境情報からContextを生成し、車両制御やさまざまなアプリケーションで利用

Contextの生成
 sensor → COLLECTION → 抽象化 → Context → Storage → Use

Contextの利用
 ・従来、上位層のTCPでは、輻輳を防ぐため、輻輳ウィンドウサイズ(cwnd)をアルゴリズムによりエンドツーエンドで動的に制御している
 ・そこで、各端末がContextを利用し、周辺端末数に応じて最適なcwndに定めて通信を行うことで、性能を向上させる

提案するTCP
 帯域幅遅延種を、通信を行っている周辺の端末間で均等に分けることにより、公平かつ効率の良い通信を実現

固定するcwnd値の計算式
 帯域幅遅延種(bit) = 帯域幅(bit/s) × RTT(s)
 cwnd = 帯域幅遅延種 ÷ 周辺端末数

通信を行っているすべての端末が、上式で求めたcwnd値に固定して通信するものと想定

シミュレーションモデル

ネットワークシミュレータOMNeT++ version 4.1
 全ての端末がAPとルータ2台を経由し、サーバへパケットを送信し続けるモデルを構築

シミュレーションパラメータ

シミュレーション時間	20sec
無線規格	IEEE802.11g
RTT(APからサーバの有線部分)	40msec
データレート	54Mbps
周波数帯域	2.4GHz
最大セグメントサイズ	65,280Byte
広告ウィンドウサイズ	33,423,360Byte
端末数 (距離I = 距離II)	2~50台
距離I, 距離II	5m, 50m

通信性能比較
 ・従来のTCP(TCP New Reno)
 ・独自のTCP

シミュレーション結果

トータルスループットおよびRTTを比較

独自のTCPでは、従来のTCP(ほど)端末数の増加に伴うスループットの低下は見られなかった

また、RTTを比較しても、従来のTCPは大きく増加しているのに対し、独自のTCPでは安定していた

独自のTCPでは、従来のTCPよりも公平で効率の良い通信が行われている