

小口研究室 研究紹介 (2011年度)

(お茶の水女子大学理学部情報科学科)

Smart GridのEV活用に向けたシミュレーションによる性能評価(研究担当:唐石 景子)

世界的な環境問題

解決策? 電力需要の増加

停電などの電力トラブル

自然エネルギー発電の導入

大量導入には様々な問題が...

自然エネルギー発電は出力変動が大きい → 発電量がよめず調整が困難

研究背景

電力の流れ ITによる制御

原子力発電所 水力発電所 風力発電所 太陽光発電所

工場 ビル 住宅

EV (Electric Vehicle) 電気自動車

- 環境問題に貢献
- 今後広く普及していく見込み

スマートグリッドとは

「賢い電力系統」を意味する電力エネルギーのネットワークを効率的に運用することが広義

特徴: 自然エネルギーの大量導入が可能 エネルギーが双方向に流れる 「エネルギー」+「情報」のやり取り

研究目的

EVのバッテリーと充電ステーション間の効率的な電力エネルギーのやり取りをシミュレーションによって評価

◆エネルギー需給設備としてのEVに注目

- 蓄電池として利用が期待されている
- 充電インフラを通じ、施設に電力エネルギー供給

◆ITを用いた安定化制御が不可欠

- ただし、EVは接続が固定されていない
- グリッド上の電力量がタイムリクに変動
- 固定の需電量を扱った場合と比べて制御が複雑化

スマートグリッドを利用!!!

実験概要

評価用モデル: 8500-Node Test Feeder

各EVの出力: 5[kW]

太陽光発電所の出力規模: 1200[kW]

各EVの出力: 2[kW]

太陽光発電所の出力規模: 2400[kW]

実験: 太陽光発電所の出力量が変動した状況と安定(100%→0%)

評価用モデルに太陽光発電所とEVを追加

EVからの出力でグリッド全体の電圧変動をどの程度カバーできるか

設置箇所: 120/240[V]地点の各々種2354か所

各EVの出力は5[kW]・2[kW]の2パターンで実験

それぞれのパターンを組み合わせて検討

EVの放電による電圧変動を観察

グリッド上の1177ヶ所

目標となる規定範囲を設定

通常状態の電圧値とどの程度差があるかを調査

実験結果

◆総合的な電力システムシミュレータを利用 OpenDSS (Open Distribution System Simulator)

各条件下でEV全体の総出力量を6パターンに変化させ検証

各EVの出力: 5[kW] 太陽光発電所の出力規模: 2400[kW]

各EVの出力: 2[kW] 太陽光発電所の出力規模: 2400[kW]

規定範囲内(0.95~1.05)に収まった地点の割合(%)

電圧変動を平均化した標準偏差

※発電所の出力規模が小さい程グリッド全体の電圧変動が抑えやすい

※EVの放電は多くの数値を用いた方がグリッド全体の電圧安定化が容易

まとめと今後の課題

太陽光発電所とEVを追加したスマートグリッドシミュレーションシステムを構築

環境の要因による電圧変動状況の比較、検証

今後の課題

グリッド状況から自動的に電力調整を行う任意の制御アルゴリズムの実装、評価

環境情報を利用した上位層における通信制御(研究担当:飯尾 明日香)

通常の無線通信

- 各端末は周囲の状況に応じた通信設定を行っていない

車

- 搭載された無数のセンサから周囲の環境情報(Context)を取得可能

TCPの輻輳制御

- 周囲の環境情報を活かした通信設定を行えば通信性能の向上が見込める

環境情報を用いて適切に通信パラメータ設定を行い、より効率的な通信を実現

シミュレーション内容

使用するContext: 周辺端末台数

利用先: TCPの輻輳制御

複数のクライアントがAPとルータを経由しサーバへパケットを送信

各クライアントはサーバへ2MBずつ送信

輻輳制御の有無で通信性能を比較

- TCP Tahoe.cc (一般的なTCP)
- TCP Reno.cc (一般的なTCP)
- TCP NewReno.cc (一般的なTCP)
- NoCongestionControl.cc (cwndサイズを大きな値に保つTCP)

IEEE802.11

ビットレート 6Mbps

広告ウィンドウサイズ 8,356kByte

最大セグメントサイズ 65,280Byte

1. クライアント数に伴うトータルスループット

2. 往復遅延時間(RTT)に伴うトータルスループット

1. クライアント数に伴うトータルスループット

スループット(Mbps)

クライアント数

TCP Tahoe

TCP Reno

TCP NewReno

NoCongestionControl

TCP NewReno

輻輳制御のあるTCPでは、シミュレーション時間が経過しても各端末のcwndサイズが上手く成長していない

- クライアント数が大きい場合、TCP TahoeやTCP Reno、TCP NewRenoで最適な通信が行われているとは言えない
- 周辺の端末数がわかれば、より適した設定が可能

2. RTTに伴うトータルスループットの比較

スループット(Mbps)

RTT(ms)

TCP NewReno

NoCongestionControl

2Nodes

5Nodes

10Nodes

cwndサイズを大きく保たれたNoCongestionControlにおいて、そのcwndサイズは低遅延では十分な大きさだが、高遅延では不十分で、性能を制限してしまっている

cwndサイズを上手く設定できれば

高遅延環境において従来のTCPより良い性能での通信が可能

Android端末におけるアプリケーションに依存する通信特性に関する一検討(研究担当:熊谷 菜津美)

研究背景

- 従来の携帯電話からスマートフォンへ
- 手軽にインターネット機能を利用可能
- リアルタイムに情報を取得・発信

Android OS

オープンソース 様々な会社で製作

トップシェアを占めるスマートフォンOS

※注目度が高い

Google社開発のソフトウェアプラットフォーム

ソースコードをオープンソースとして無償で提供

誰でも自由にアプリケーション開発可能

Androidアーキテクチャ

アプリケーション

アプリケーションフレームワーク

ライブラリ

Linux Kernel

カーネルモニタ

モニタ関数を導入

輻輳制御の解析

研究目的

アプリケーション通信特性の違い

通信方向の視点

- Upload
- Download
- 双方向通信

スマートフォンの利用目的

データのUploadが簡単、Uploadアプリケーションの増加

各アプリケーションの特性にも応じて通信制御を行い、全端末で公平な通信を目指す

実験対象アプリケーション

- リアルタイム通信: Ustream
- ファイル転送: ES file explorer, Picasa Tool

実験環境

サーバ

- OS: Fedora 15
- CPU: Intel Celeron (R) 2.30GHz
- メモリ: 2GB

Android

- モデル番号: Nexus S
- ファームウェアバージョン: 2.3.4
- ベースバンドバージョン: I9023XXXB1
- カーネルバージョン: 2.6.35.7-kaori1198-g382d80-dirty
- ビルド番号: GRJ22

実験(1)

1台のAndroid端末の通信

輻輳制御の検証

- Ustream: 40程度まで上昇、安定を保つ
- ES file explorer: 60程度まで上昇、安定を保つ
- Picasa Tool: 25程度まで上昇、不安定

複数台のAndroid端末の通信(端末数4台のとき)

- 異なるアプリケーションを実行

Android1: Ustream

Android2~4: ES file explorer

【実験結果】

リアルタイム通信増大

↑CWNDが上がり続ける場合、上がりにくい場合の2パターン

ファイル転送端末の上限值抑制

【実験結果】

リアルタイム通信増大

↑常に安定してCWNDを上げられた

ファイル転送増大

↑上限値を抑えたことで、転送時間が増加

リアルタイム通信増大の上限值抑制

【実験結果】

ファイル転送増大

↑転送にかかる時間が短縮

リアルタイム通信増大

↑配信動画の見え方、PC上には流れるまでの時間に影響ない

実験(2)

通信状況の分からない端末とAndroidの同時通信

- Nexus Sのカーネルを切り替え、パケット送出手の積極性を弱める

【測定環境】

- default TCPのNexus S+PC2台
- HT-03Aのカーネルを移植したNexus S+PC2台
- HT-03Aのカーネルを移植+CWND上限値20のNexus S+PC2台

Androidなし

ubuntu11.04

ubuntu11.04

【2台PCの性能比較】

Throughput(Mbps)

- default(TCP)Nexus S
- Nexus S
- HT-03Aを移植したNexus S
- HT-03Aを移植した上限値20のNexus S
- Androidなし

積極性の弱いHT-03Aのカーネルを用いることで性能向上

Androidなしの場合のスループットに近い値

Ustreamの動画品質に影響はない