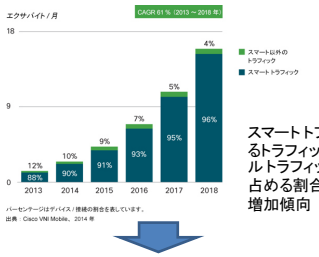
Androidアプリケーションからの通信特徴抽出手法

日本電信電話株式会社
NTTネットワーク基盤技術研究所
中野雄介，上山憲昭，塩本公平

大阪大学
サイバーメディアセンター
長谷川剛
大学院情報科学研究科
村田正幸

背景

- スマートフォンの普及による，スマートフォンのトラフィックの増加



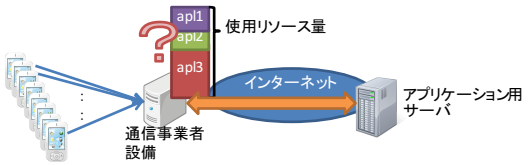
モバイルネットワークに対する影響の増大

http://www.cisco.com/web/JP/solution/isp/ipngn/literature/white_paper_c11-520862.html

課題

- 通信事業者以外の事業者やユーザによるアプリケーション作成
  - ネットワークへの負荷を意識しないアプリケーション

特定アプリケーションによるリソースの専有



リソースを専有しているアプリケーションの特定が困難

目的

Androidアプリケーションの通信特徴抽出

目的達成でできること

アプリケーション固有の通信の特徴を用いて，ネットワーク内で収集された，複数アプリケーションのパケットが混在するキャプチャデータからアプリケーションごとのパケットに分類

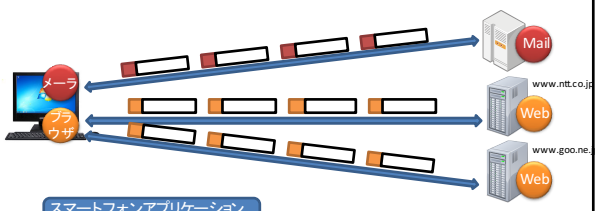
アプリケーションごとのネットワークにおける振る舞いを抽出

アプリケーションごとのインストール数を特定

リソースを専有しているアプリケーションを特定

既存技術：5-tupleによる分類

- TCPヘッダ，IPヘッダの情報でアプリケーションごとのパケットに分類



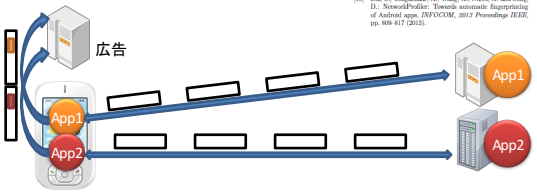
スマートフォンアプリケーションへの適用困難

複数種類のアプリケーションが共通のサーバと通信する場合，アプリケーション毎の分類が困難
例) TwitterのAPIが実装されたアプリケーションは複数存在

既存技術：NetworkProfiler

- Androidアプリケーションを解析することで通信のFingerprintを抽出
  - Fingerprint：広告取得時のHTTPに付加するアプリケーションごとのIDなど
  - ネットワークに接続された端末における特定アプリケーションの起動を特定

[10] Dai, S., Tongaonkar, A., Wang, X., Nucci, A. and Song, D.: NetworkProfiler: Towards automatic fingerprinting of Android apps, *INFOCOM, 2013 Proceedings IEEE*, pp. 809–817 (2013).



パケットの分類には適用困難

アプリケーションの広告取得以外のパケットの分類は対象外

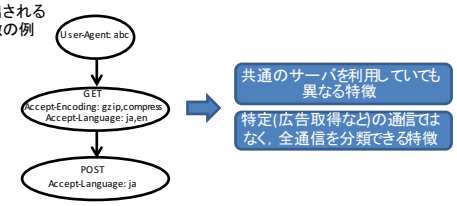
提案手法

- 各アプリケーションのソースコードを解析し，

通信の順序 と 各通信に含まれる文字列 (通信の内容)

を通信の特徴として抽出する

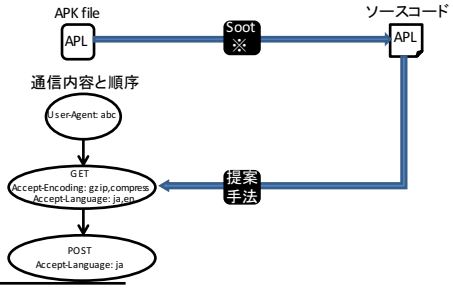
抽出される特徴の例



 7

0. ソースコードの抽出

1. Sootを利用してAPKファイルからソースコード抽出
2. 提案手法により，ソースコードから通信内容とその順序を抽出



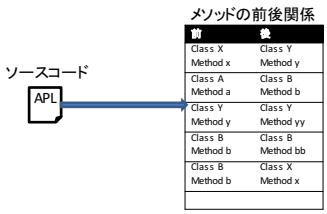
※ Soot http://sable.github.io/soot/

 8

1. メソッドの前後関係抽出

前後関係があると判断されるソースコードの箇所

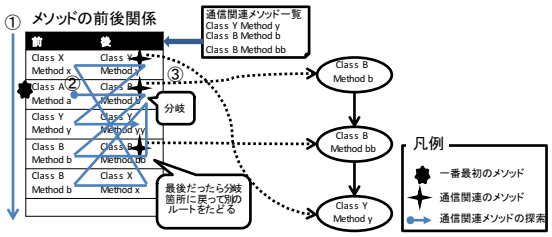
- メソッドの呼出
  メソッドaからメソッドbを呼出し
- Intentの使用
  画面Aから画面Bに遷移
- イベントリスナによる実行
  setOnXXXListener →OnXXXListener.onXXX
- ActivityやAsyncTaskクラスなどのライフサイクル



 9

2. 通信関連メソッドの呼出順序抽出

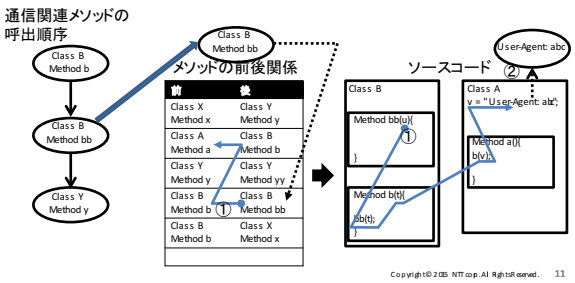
① 一番最初のメソッドを発見(「後」の行に書かれていないメソッド)
② 通信関連メソッドを探索
③ 発見された順序で，通信関連メソッドの呼び出し順序を抽出



 10

3. 通信内容の順序抽出

① 対象のメソッドから，メソッドの前後関係を逆にソースコードを探索し，対象メソッドの引数を発見
② 発見された引数を通信内容として抽出

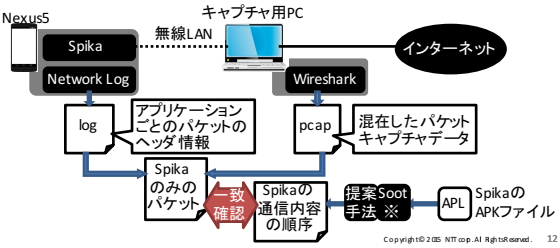


 11

評価手法

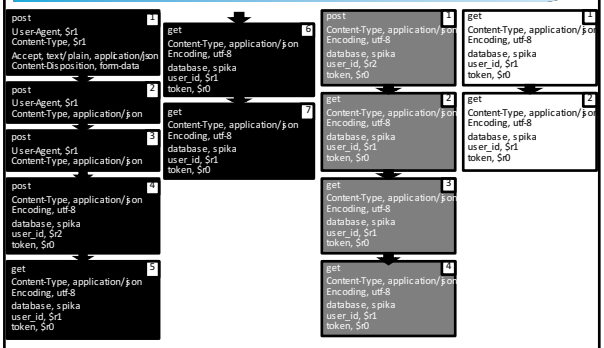
- 評価対象アプリケーション：Spika
  - オープンソースのインスタントメッセンジャーアプリケーション

Spikaのみのパケット ⇔ 一致確認 ⇔ Spikaの通信内容の順序 (提案手法で抽出)



 12

抽出された通信内容の順序

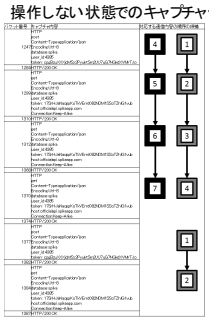


 13

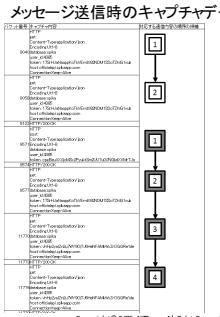
一致確認の結果

- キャプチャデータと，抽出された通信内容の順序との一致を確認

操作しない状態でのキャプチャデータ



メッセージ送信時のキャプチャデータ



 14

まとめと今後の課題

- 複数のアプリケーションのパケットが混在するキャプチャデータから，アプリケーションごとのパケットに分類するための，アプリケーションごとの通信の特徴を抽出する手法を提案
- 通信の特徴として

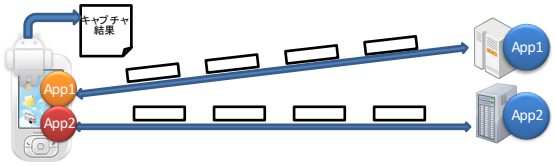
  通信の順序 と 各通信に含まれる文字列（通信の内容） を抽出

- 今後
  - Spika以外の多様なアプリケーションで，提案手法の有効性を確認
    - 複数アプリケーションが特定のサーバと通信していても，異なる特徴が抽出されること
  - 抽出された特徴を用いて，混在するキャプチャデータからアプリケーションごとのパケットに分類する手法の検討

 15

既存技術：Network Log

- スマートフォン内でアプリケーションごとに分類してパケットをキャプチャ



ネットワーク内でのキャプチャ結果の分類に適用困難

- ネットワークリソースを専有しているアプリケーション発見のためには，ネットワーク内でキャプチャされた複数端末からのパケットを分類する必要がある．

 16

既存技術：AndroidLeaks

 17

既存技術：TaintDroid

 18