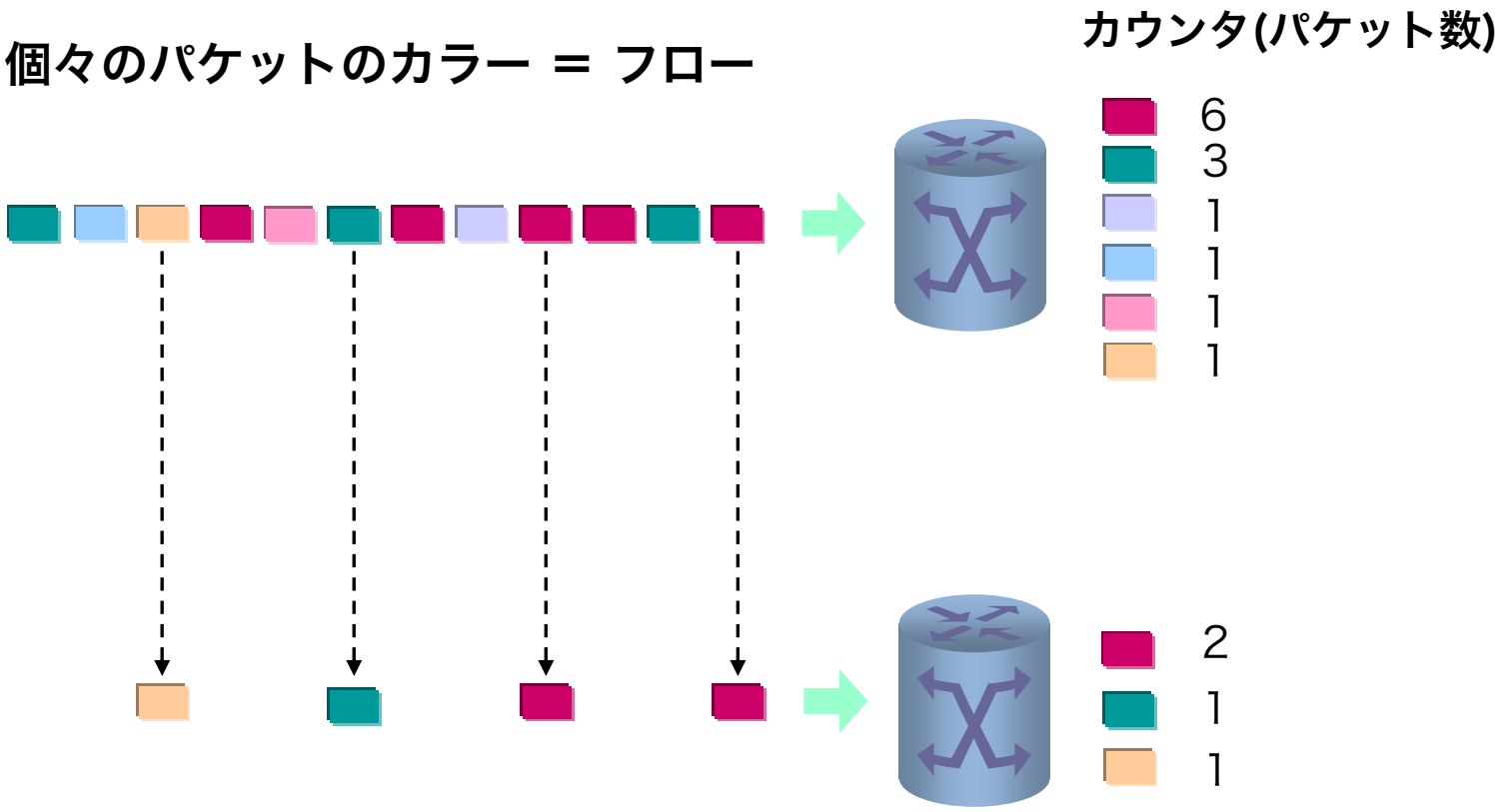
パケットサンプリングで失われる情報は何か？

Internet Week 2006
xFlow Operators' BoF

2006.12.7

NTT サービスインテグレーション基盤研究所
森　達哉
tatsuya@nttlabs.com

フロー計測とパケットサンプリング



パケットサンプリングのメリット☺

- **ルータの負荷軽減**
  - 処理速度：OC-192 → 1 pkt / 8ns
  - メモリ量：同時管理フロー数 (フローキャッシュ)
- **コレクタの負荷軽減**
  - ディスク消費量：raw flow records / SQL DB
  - 計算量：統計量の計算

→ スケーラビリティの向上

パケットサンプリングのデメリット☹

- **情報の喪失**
  - 統計的推定・誤差の評価が必要
    - パケットカウント・バイトカウントは取り扱いが比較的容易
  - 統計的手法によっても推定が困難なクラスがある
    - 種類の数：出現フロー数、出現アドレス数など

評価方法

- パケットヘッダのキャプチャデータを使い、パケットサンプリング + NetFlow をオフラインでエミュレート
- キャプチャデータ
  - pcap format (tcpdump)
  - DAG format (ENDACE DAG シリーズ) 10G 対応
- サンプリング + フローツール
  - 自作 (sampling + NetFlow-like なフロー計測機能を実装)

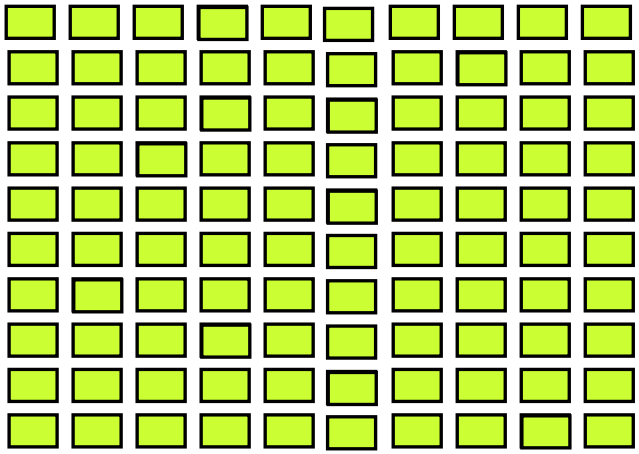
推定が容易なクラス：
パケットカウント，バイトカウント

- ある IF を通過した UDP パケット数
- src / dst AS 毎のパケット数
- 上位 Top 10 フローのパケット数

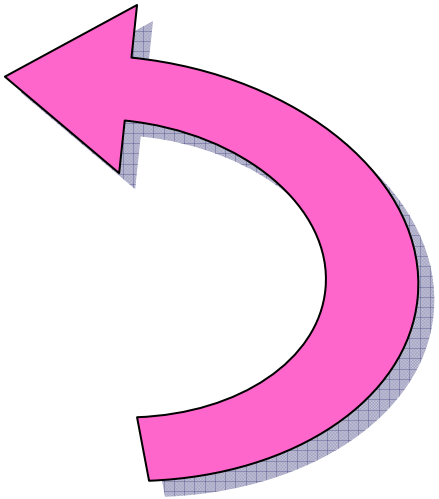
※理論的な誤差の詳細についてはIRS 10の資料を参照
http://www.bugest.net/irs/docs_20060922/irs10-mori-20060922.pdf

パケットカウント

サンプル前のパケット数の推定値＝ 10 x 1 / 0.1 = 100

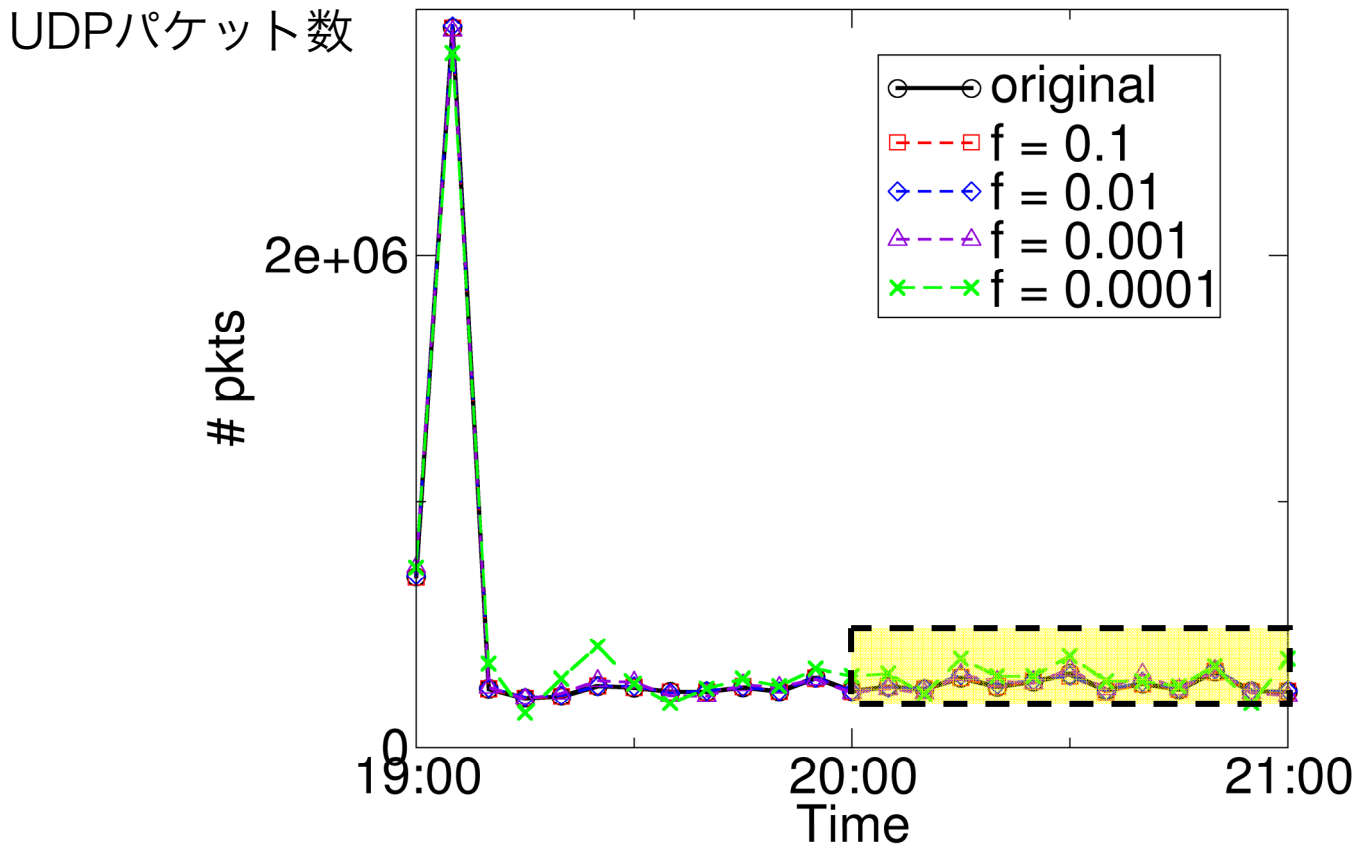


レート = 1/10 (=0.1) の
パケットサンプリング

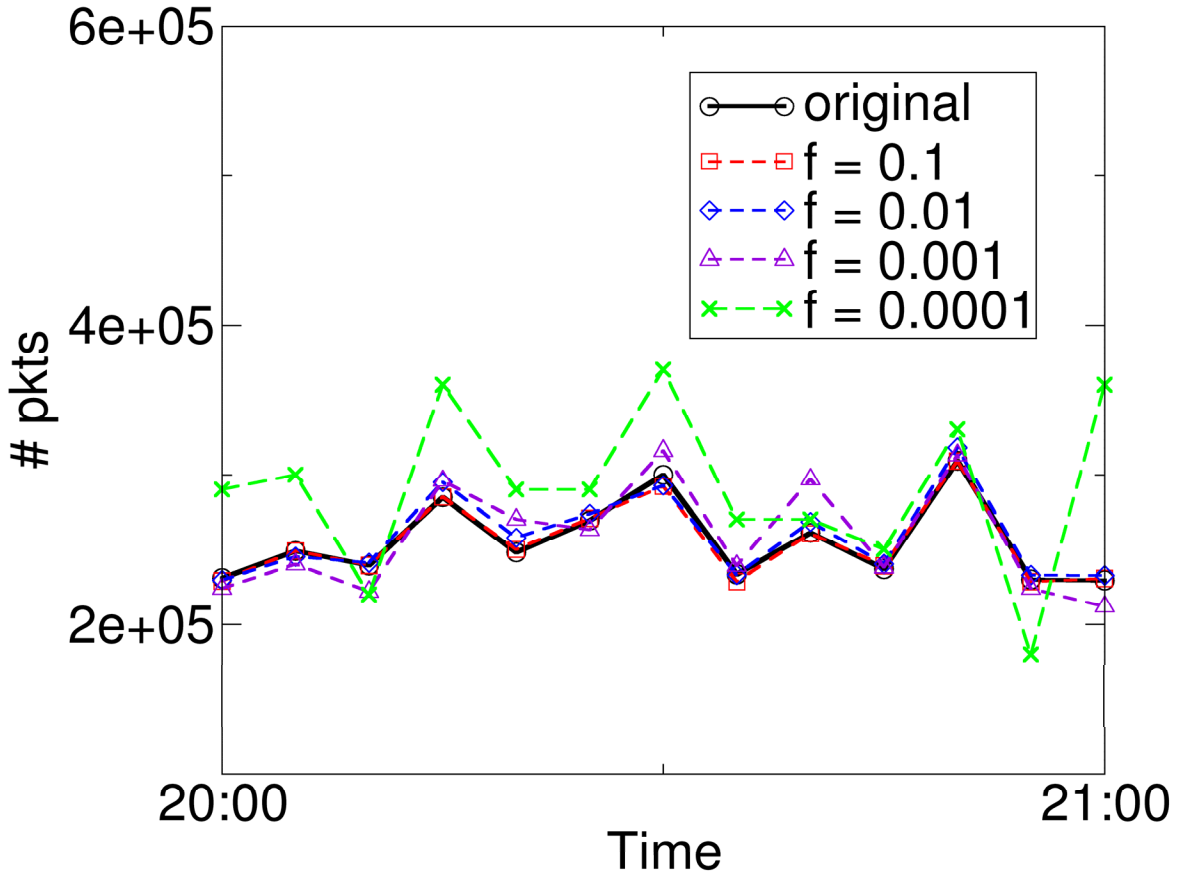


うち１０パケットがサンプルされた
（実際に観測した値）

実例（真の値 vs. 推定値）


UDPパケット数
original
f = 0.1
f = 0.01
f = 0.001
f = 0.0001
2e+06
0
pkts
19:00
20:00
21:00
Time

拡大図


6e+05
4e+05
2e+05
pkts
20:00
21:00
Time
original
f = 0.1
f = 0.01
f = 0.001
f = 0.0001

実際の観測値と推定値

- 19:00 – 19:05 のUDPパケット数

	サンプルUDP パケット数 Y	推定UDP パケット数 X*
Original	691,693	
f = 0.1	68,888	688,880
f = 0.01	6,973	697,300
f = 0.001	732	732,000
f = 0.0001	73	730,000

推定が困難なクラス：出現フロー数、アドレス数など

- **IF毎に出現したフロー数**
  - フロー数の急激な増加：
    → worm outbreak, SYN Flooding, network/port scanning
  - 大域的な異常検出に有用
    - ポイントだけを監視するIDSとは異なるアプローチ
- **パケット・バイトカウントのように単純ではない**
  - サンプリングレートの逆数をかけても×
  - **種類の数（異なり数** = cardinality）
    - 出現フロー数、アドレス数、ポート数など

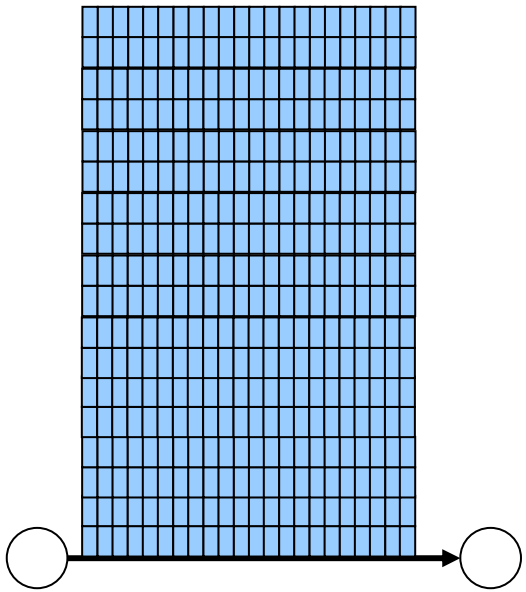
種類の数（異なり数）

- **種類の数、ユニーク○○数**
  - cat hoge.dat | sort –u | wc –l の結果

- **他の「種類の数」の例**
  - ある本に含まれる単語の種類の数
  - 出現アドレス数
    - ある時間帯で観測した src or dst IP アドレスの数
  - ポート別アクセスアドレス数
    - dst port が tcp.port 135 にアクセスしている src IP が急激に増加 → ワームの発生
  - src IP 毎の dst IP数
    - Network scan をしているホストを検出など

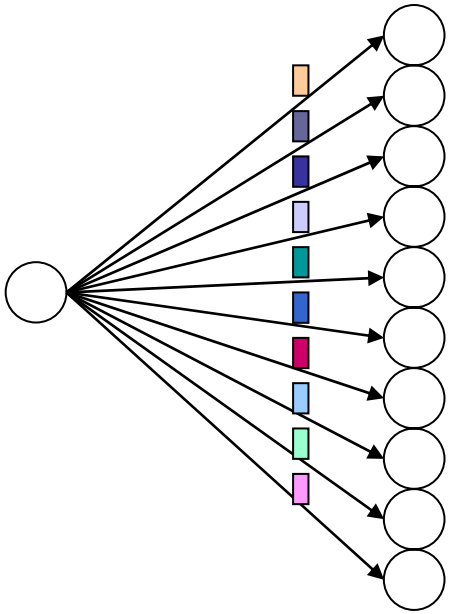
種類の数の例

種類の数=フロー数=１



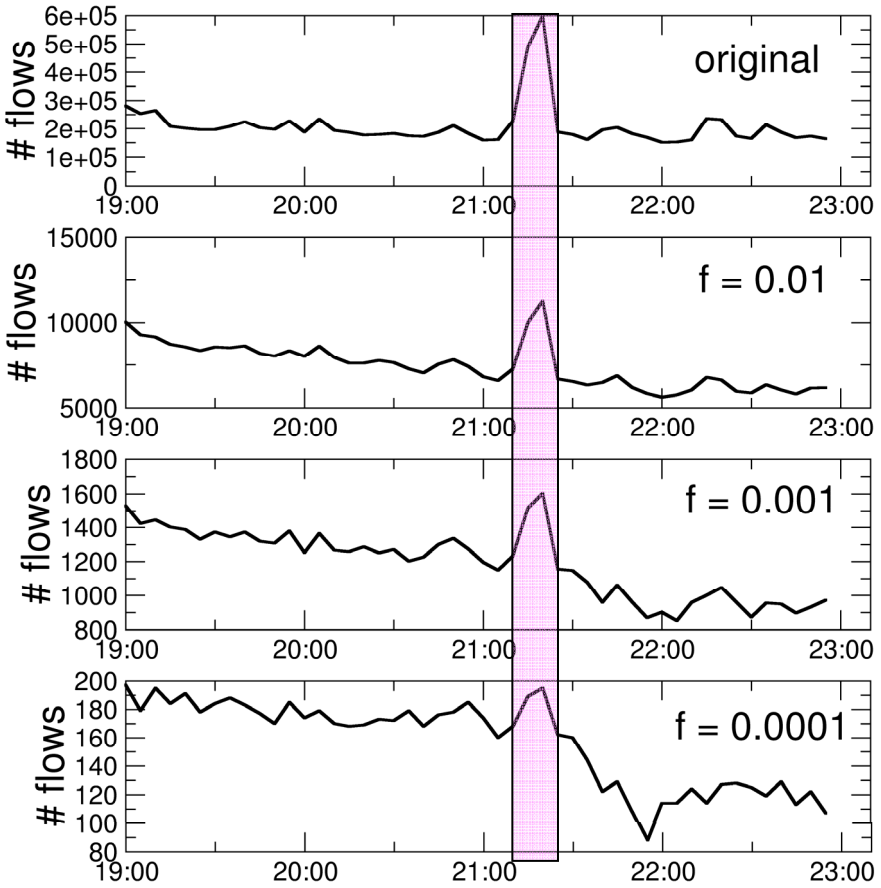
Heavy hitter

種類の数=フロー数=たくさん



Network scan

実例（5分あたりの出現フロー数）



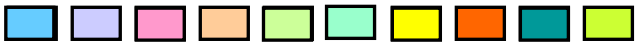
実際の観測値

通常時、異常時のフロー数

	(サンプル)フロー数 (19:00 – 19:05)	(サンプル)フロー数 (21:25 – 21:30)
Original	281,318	598,775
f = 0.01	10,033	11,252
f = 0.001	1,529	1,603
f = 0.0001	197	195

種類の数とサンプリング

パケットのカラーの総数：10



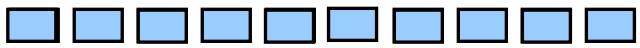
レート = 1/2 (=0.5) の
パケットサンプリング

サンプル後のカラーの総数：5

サンプル前のカラーの総数 = 5/ 0.5 = 10?

種類の数とサンプリング

パケットのカラーの総数：1



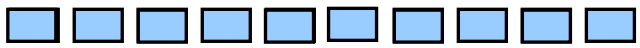
レート = 1/2 (=0.5) の
パケットサンプリング

サンプル後のカラーの総数：1

サンプル前のカラーの総数 = 1/ 0.5 = 2?

種類の数とサンプリング

パケットのカラーの総数：1



レート = 1/10 (=0.1) の
パケットサンプリング

サンプル後のカラーの総数：1

サンプル前のカラーの総数 = 1/ 0.1 = 10?

種類の数とサンプリング

- 10,000パケットに対して 1/10,000サンプリングをする場合を考える

- Heavy-hitter: 10,000 pkts x 1フロー
- Network scan: 1 pkts x 10,000 フロー

- サンプルされたパケット数(=フロー数)の期待値 = 1

  → **その1パケットから上記2つのケースを区別することはできない！**

種類の数とサンプリング

- **（前スライドのつづき）実際には 1～10000フローのどこかに正解がある**
- **それを知るためには、フロー毎のパケットカウントの分布がどうなっていたかを正確に知る必要がある**
  - が、それは統計的手法によっても難しい（詳細は下記文献参照）
  - ので、フロー数を正確に推定することは難しい

  → **別のアプローチが必要**

T. Mori, R. Kawahara, N. Kamiyama, and S. Harada, "*Inferring original traffic pattern from sampled flow statistics*," IEEE/IPSJ SAINT 2007 Workshop on Internet Measurement Technology and its Application to Building Next Generation Internet, Jan. 2007.

まとめ

- **パケットカウント（バイトカウントも同様）は統計的推定が容易**
- **種類の数（フロー出現数など）はパケットサンプリングによって元の統計を正しく復元できないので要注意**
  - これらの統計はボリュームベースではなく、パターンに着目した異常検出などにきわめて有効な尺度である

Future Work

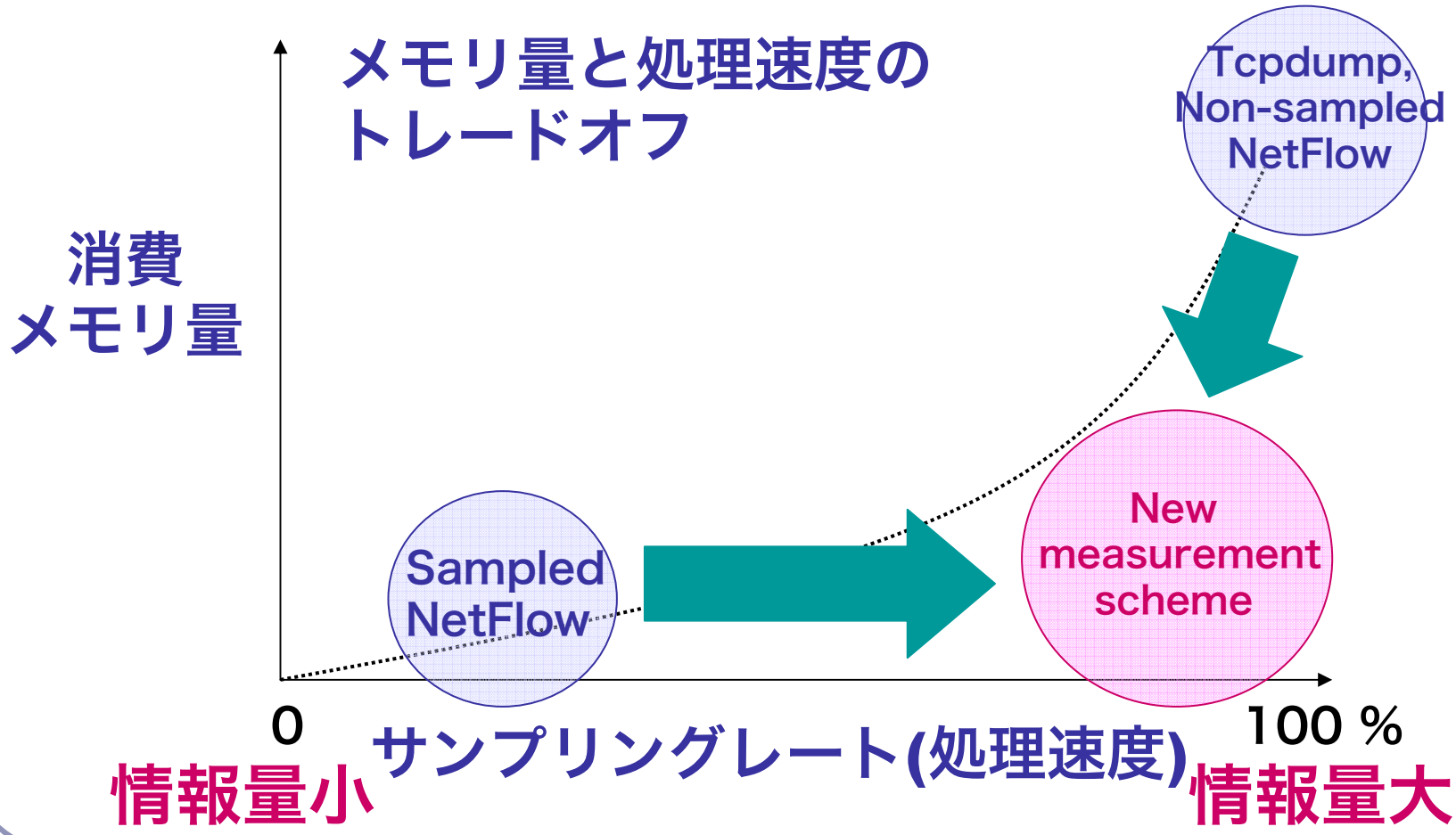
- **大域的な異常検出を目的としたフロー計測・分析は有効であると考えられる**
  - IDSとは異なるアプローチ
    - ネットワークワイド、分散性、スケーラビリティ
  - しかしパケットサンプリングとは相性が悪い

- **パケットサンプリングとは異なるアプローチ**
  - 処理速度コストをある程度犠牲にし、per-packet で計測

    →フロー数のような情報(種類の数)を取得できるように

    →しかし情報量は集約・圧縮することでスケーラビリティは確保する i.e., 単純な tcpdump ではない
    確率的計測手法（ハッシュの利用）

Future Work (cont.)



Acknowledgement

本研究の一部は総務省委託研究課題「次世代バックボーンに関する研究開発」の成果である.

貴重な意見・コメントを頂いたFlow Inspection Project のメンバーに感謝する.