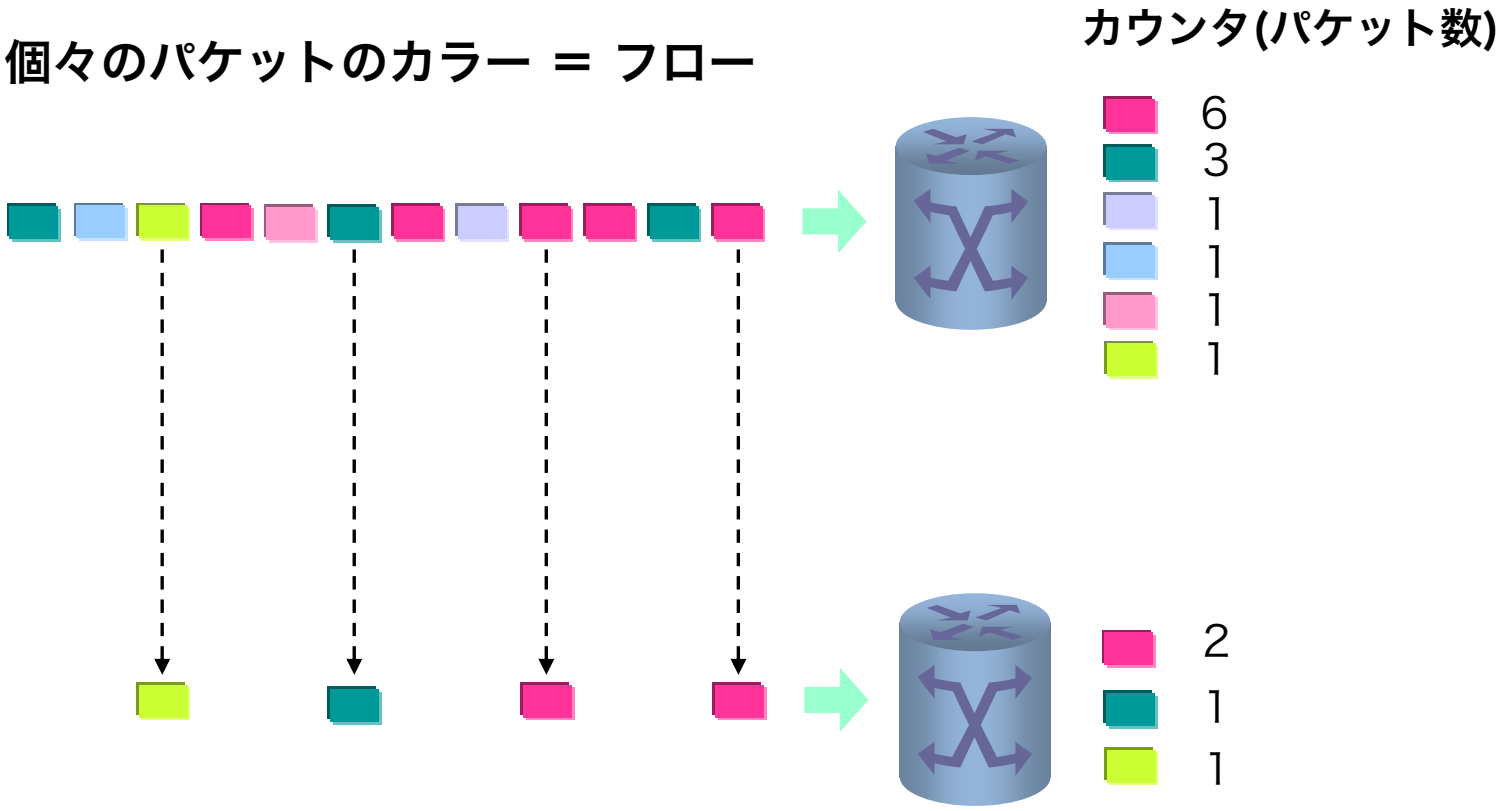
To Sample or Not to Sample?

JANOG19

2007.1.25

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



To Sample ...☺ スケーラビリティ

- **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

Not to Sample ...☹ 情報の喪失

- **サンプル情報から元の情報を統計的に推定することが必要**
- **Good news ☺**
  - **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**
    - 理論的な誤差を見積もることができる ※
- **Bad news ☹**
  - **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 ※
  - **特にセキュリティに関連する統計情報が激しく失われる**
    - OPF (One-Packet Flow) ← 本発表でとりあげる例

※ IRS 10 http://www.bugest.net/irs/docs_20060922/
Internet Week 2006 -- xFlow Operators' BoF http://www.flowops.jp/

実験方法

- パケットヘッダの計測データを使い、パケットサンプリング + NetFlow exporter + collectorをオフラインでエミュレート
- **計測データ**
  - pcap format (tcpdump)　w / gigapcap
  - 商用インターネットバックボーンリンク
- **ツール**
  - exporter: softflowd
    - pcap-cat *.pcap | sampling –n 1000 | softflowd –n collector_ip:port
  - collector: flow-tools
    - flow-capture – w /tmp/flow local_ip/exporter_ip/port

推定が容易なクラス：
パケットカウント，バイトカウント

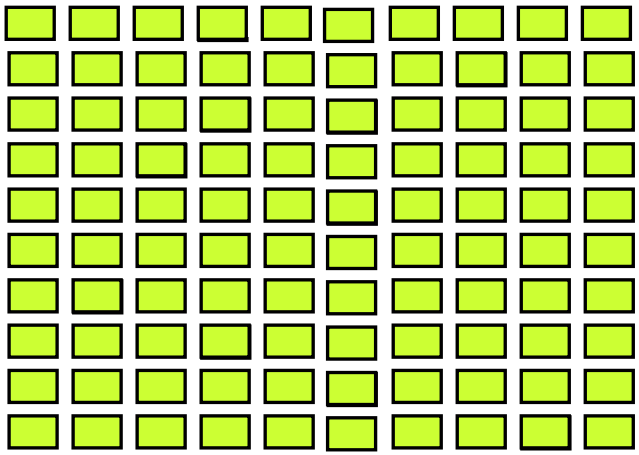
- ある IF を通過した UDP パケット数
- src / dst AS 毎のパケット数
- 上位 Top 10 フローのパケット数

※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 10 = 100

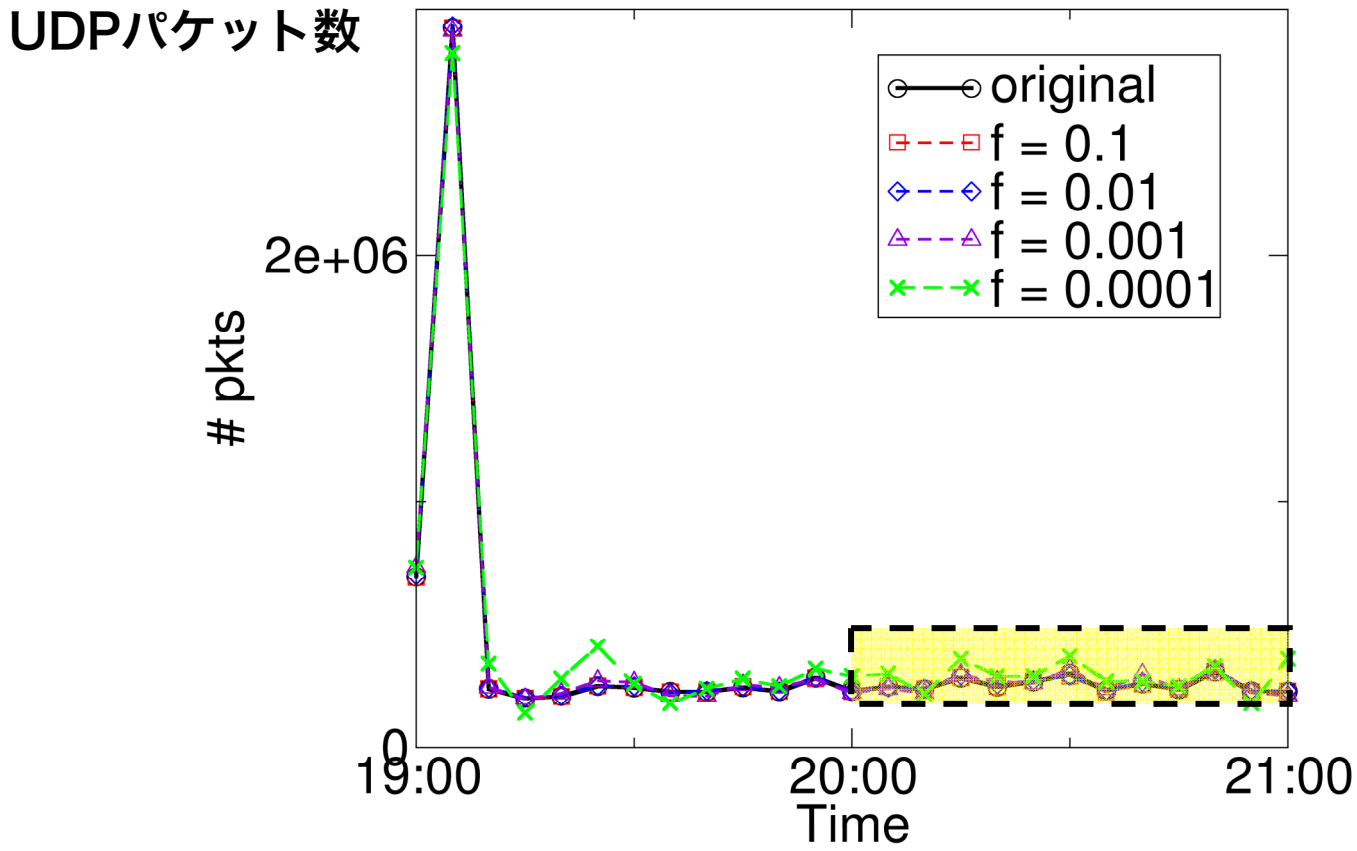
サンプリングレートの逆数をかければＯＫ！



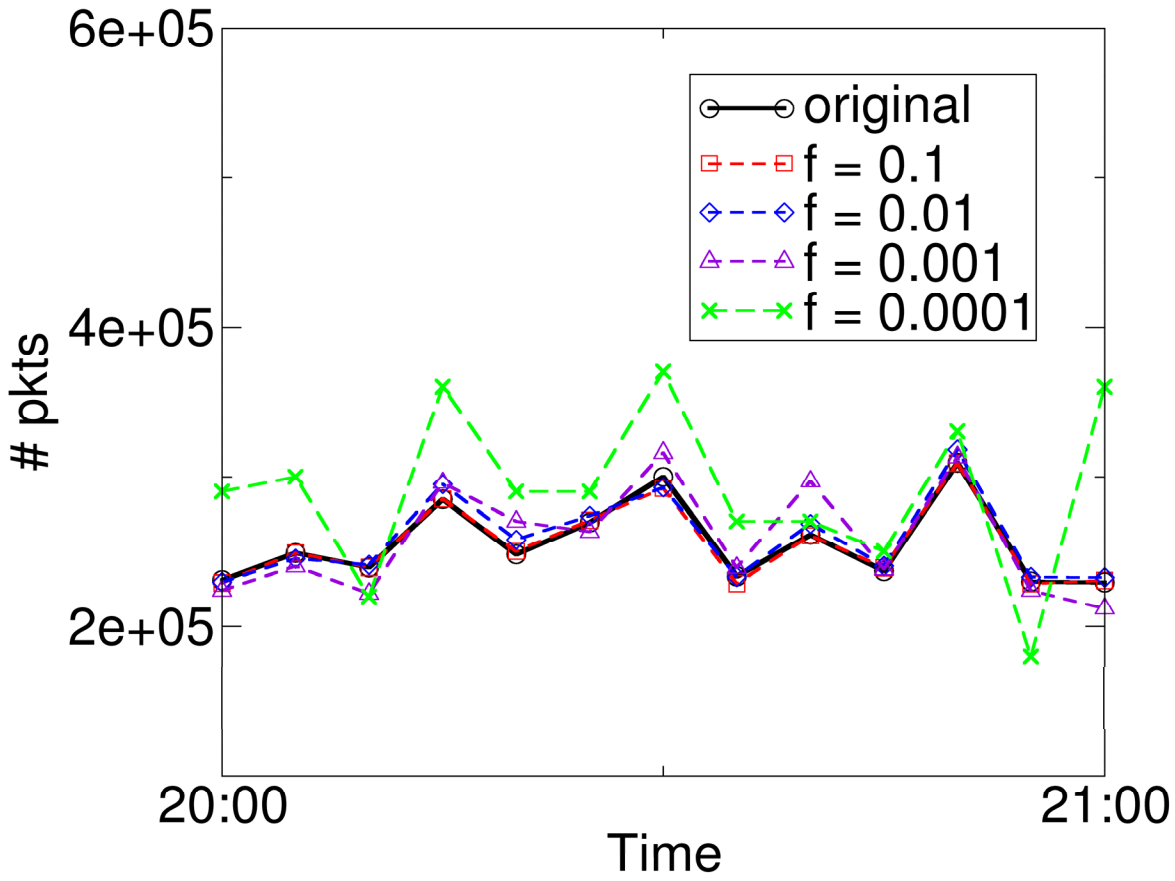
サンプリングレート = 1/10
のパケットサンプリング

うち１０パケットがサンプルされた
（実際に観測した値）

実例その１（真の値 vs. 推定値）



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

実例その２

- **Top 10 Dst AS (パケット数)**
  - サンプリング後もランキングはほぼ不変

ランキング	original	f=0.01	f=0.001	f=0.0001
1	4856531	48705	4987	484
2	2055260	21018	2035	215
3	1727424	17308	1710	161
4	1523169	15146	1537	150
5	1503497	15056	1444	145
6	1103281	11181	1117	106
7	863293	8513	870	93
8	651475	6500	654	54
9	539569	5323	539	49
10	436184	4446	459	43

実例その３

- **Top 10 Dst AS (バイト数)**
  - サンプリング後のランキングはややバラつくが
    ほぼ不変（パケットが可変長である影響はあり）

ランキング	original	f=0.01	f=0.001	f=0.0001
1	2115423411	21312850	2192814	228742
2	2052572888	20877231	2035217	212906
3	949159018	9275234	964000	110400
4	837581375	8279363	804231	81297
5	422657237	4187658	406340	38790
6	401780873	4100083	397307	31826
7	315571301	3198051	314188	30207
8	302315468	3065809	294865	29798
9	302078385	2990121	291511	26999
10	296058033	2988871	279174	25364

OPF : One-Packet Flow(s) ※

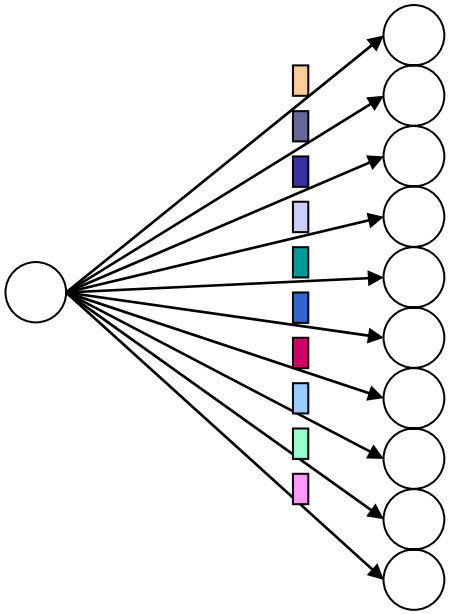
- **1パケットから成るフロー = OPF**
  - ホストスキャン、ポートスキャン
    - ワーム，bot
    - messenger spam
    - nmap 等のツール
  - SYN flooding
  - DNS, NTP, 制御系パケット等

- **OPFの比率が急激に増加→異常の兆候**

※一般的な用語ではありません

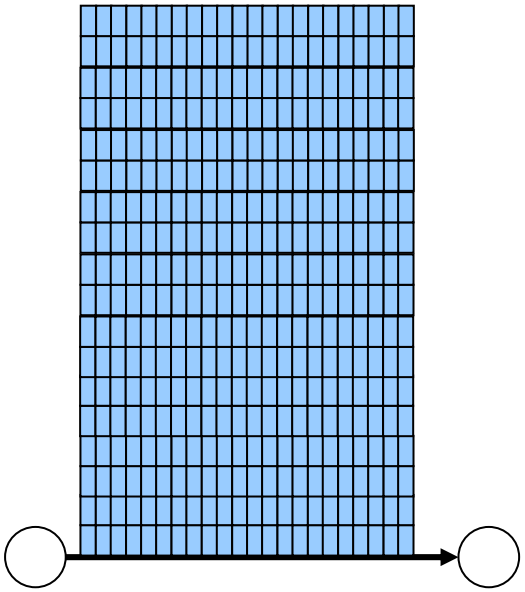
OPFの例

1フローあたりのパケット数=1



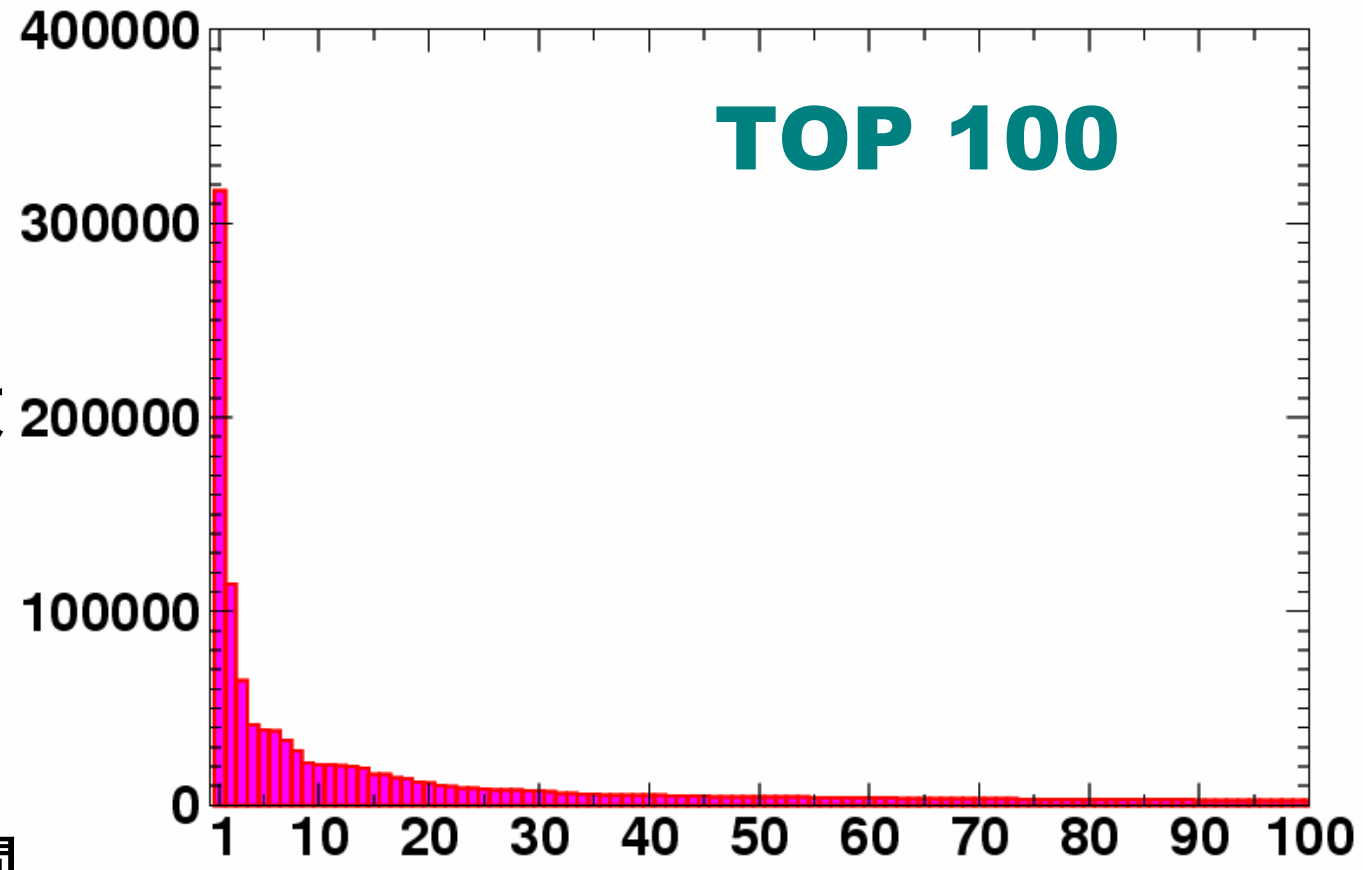
OPF(s)

[cf] 1フローでパケット多数



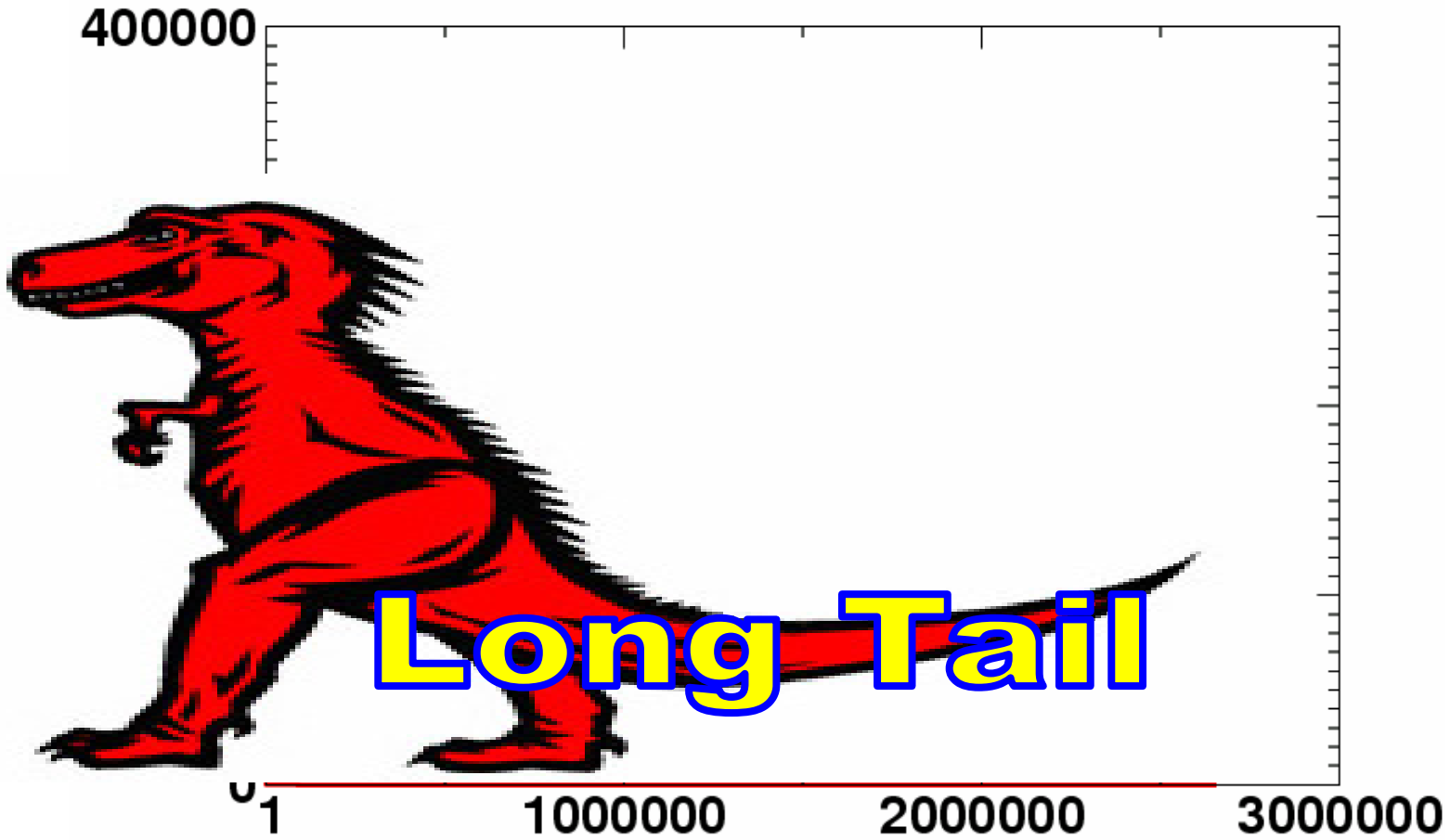
Heavy hitter

パケット数が多い順にフローをソート

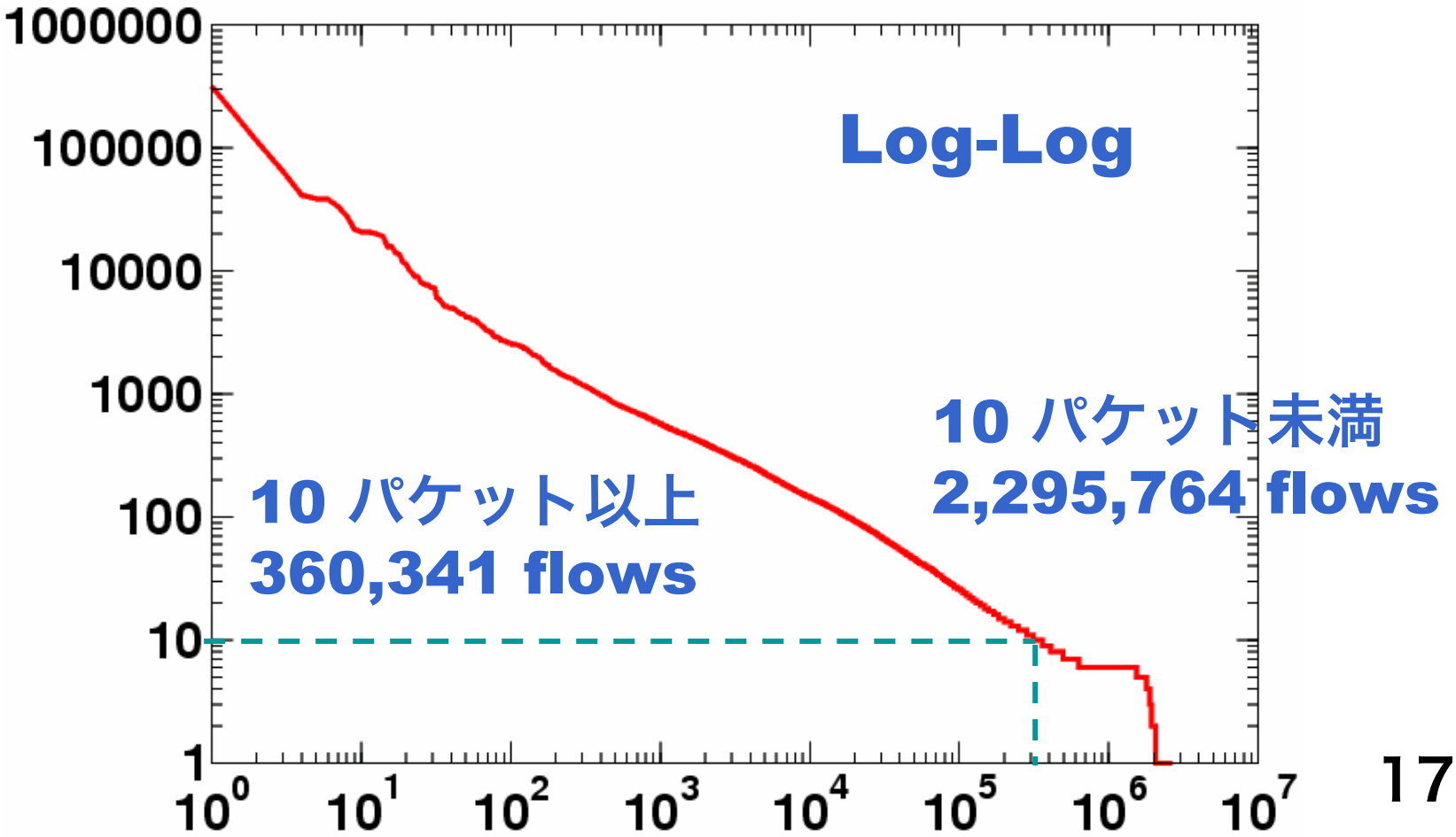


※昼間5分間
のデータ

パケット数が多い順にフローをソート

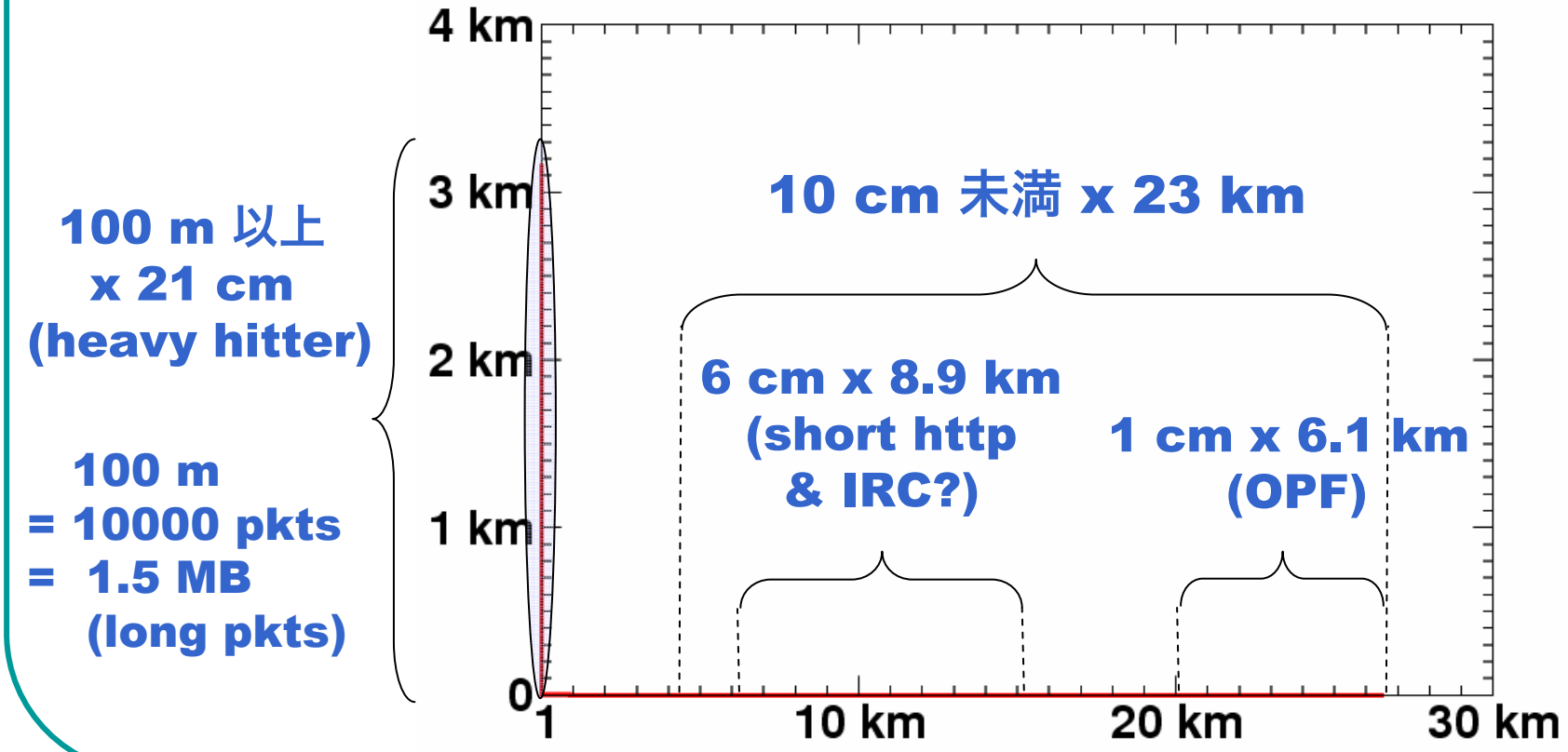


パケット数が多い順にフローをソート



フローとロングテール

- 1フロー × 1パケットを 1 cm × 1 cm で表現すると・・・ © 梅田望夫, ウェブ進化論



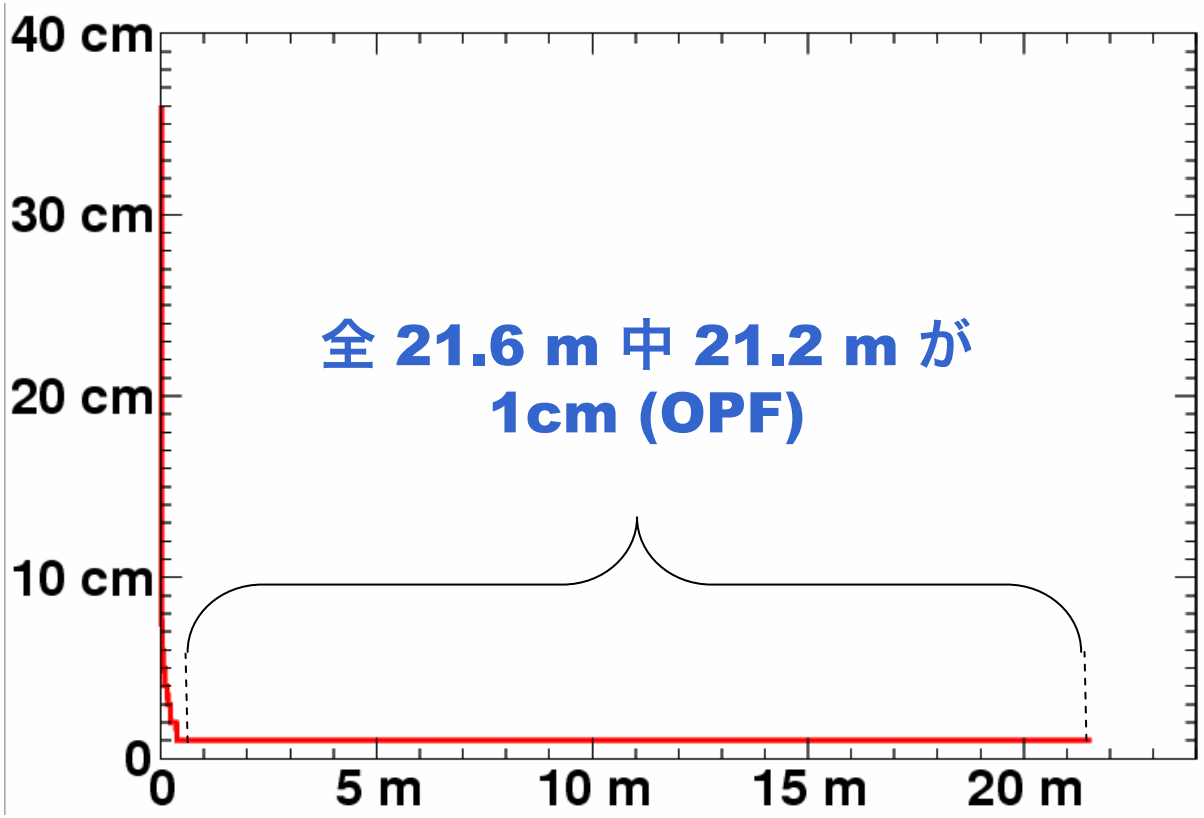
ロングテールの意味するところ

- **ヘッド**
  - 超巨大なフローが存在する
  - トップ 1 % のフローで全パケットの約 26 % を占める
- **テール**
  - ごく小さなフロー(OPF)が無数に存在する
  - 全 2,656,105 フロー中 611,023 フローが OPF (23 %)
  - 大多数が小さいフロー (86 % が 10 パケット未満)

※**大多数のフローはパケットサンプリングで**
ひっかからない

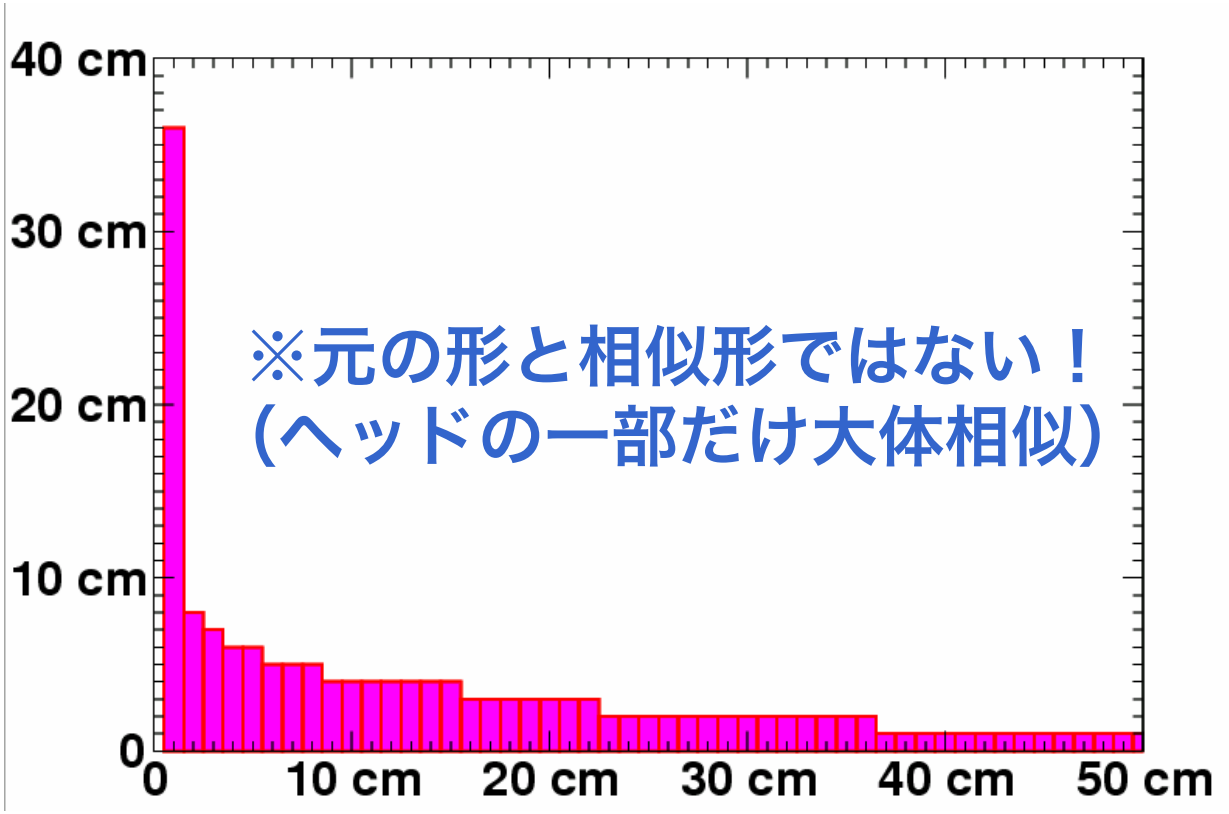
パケットサンプリングとロングテール

サンプリング後のランキング (1/10000 サンプリング)



パケットサンプリングとロングテール

サンプリング後のランキング (1/10000 サンプリング)



サンプリング後のフロー統計

- **ヘッド**
  - 上位フローはサンプリング後も把握可能
    - ここでの上位フロー == サンプルパケット数が十分であるフロー
    - 単にサンプリングレートをかけてスケールダウンしただけ
- **テール**
  - ほとんどがOPFとなる
    - 全サンプルフロー 2175 フロー中 OPF は 2121 (98%)
  - が、これらはサンプル前から OPF なのではなく、ただしくは OPSF (One-Packet-Sampled Flow)

サンプル前のテールの情報（分布の構造）は失われてしまう

実例その２

- Top 10 OPF/OPSF 発生ホストランキング
  - OPF/OPSF を最も多く発生した source host

ランキング	original	f=0.01	f=0.001	f=0.0001
1	82463	695	269	27
2	72064	332	225	26
3	34867	331	178	21
4	28852	324	163	17
5	27074	321	123	16
6	17829	307	121	16
7	14993	302	119	13
8	10095	298	119	12
9	8601	296	104	12
10	5035	290	103	12

実例その３

flow-dscan ※で “host scan” として検出された
IPアドレスの数 (パラメタはデフォルト)

サンプリングレート	検出ホスト数
1.0 (サンプリング無)	584
0.1	10
0.01	2
0.001	1
0.0001	0

※flow-tools に同梱 http://www.splintered.net/sw/flow-tools/docs/flow-dscan.html
flow-cat ft-v05.* | flow-dscan –b –m –p -W

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**
- **パケットサンプリングによってOPFの情報などが大幅に失われる**
  - このような情報が重要であれば、新たな計測方法が必要
  - ひとつの可能性 = non-sampling 処理
    - Exporter の拡張
    - Collector での処理能力
    - 新たな分析（データマイニング）

Acknowledgements

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.

表紙の写真は下記のものを引用させていただいた
no sampling, please (by brantastic)
http://www.flickr.com/photos/bran/sets/1797147/detail/

mozilla の写真は下記のものを引用させていただいた
Mozilla: Im Zeichen des Sauriers
http://phlow.net/mag/web_design_technik/mozilla_im_zeichen_des_sauriers.php

Appendix

Future Work

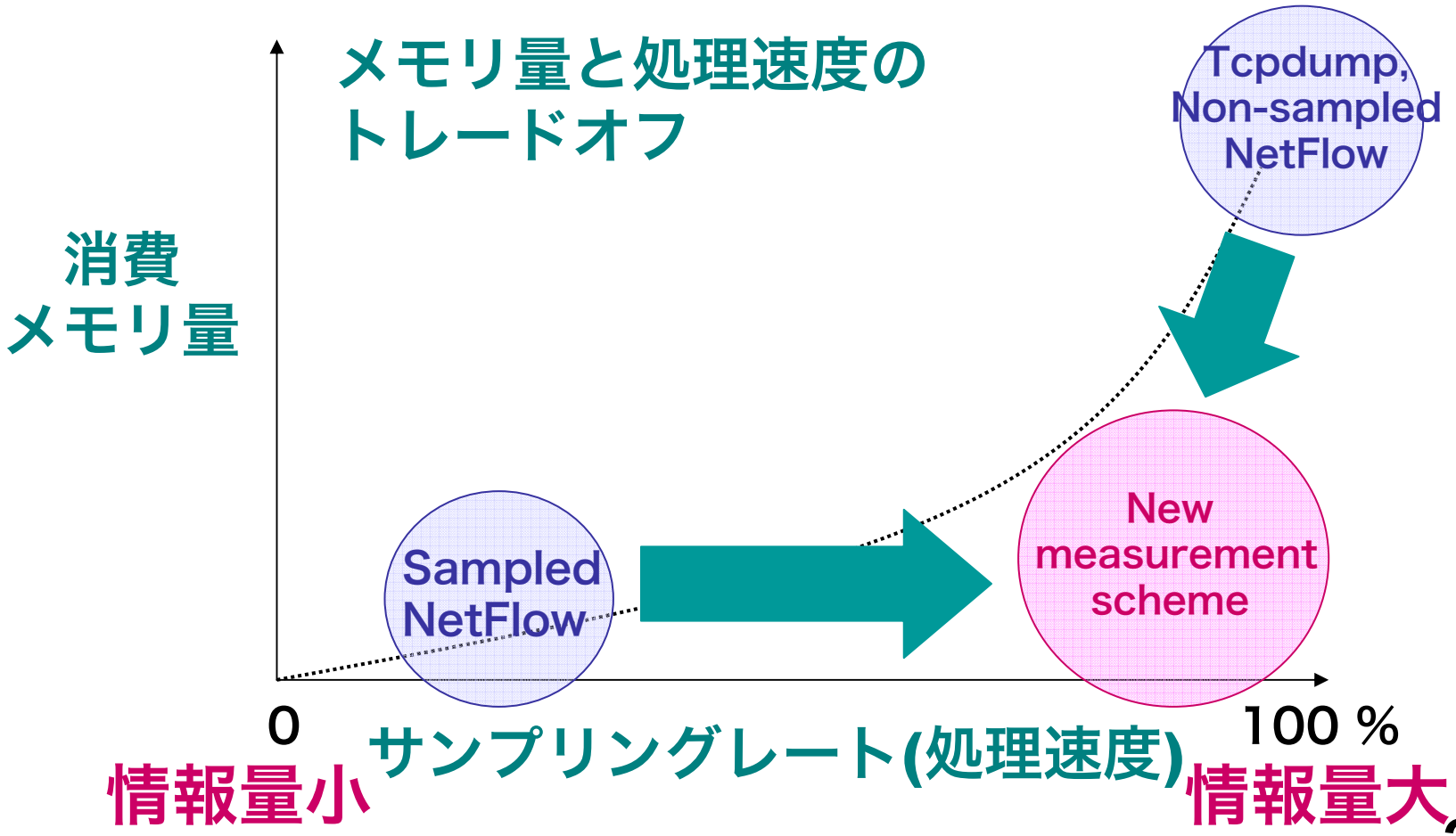
- **大域的な異常検出を目的としたフロー計測・分析は有効であると考えられる**
  - IDSとは異なるアプローチ
    - ネットワークワイド、分散性、スケーラビリティ
  - しかしパケットサンプリングとは相性が悪い

- **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

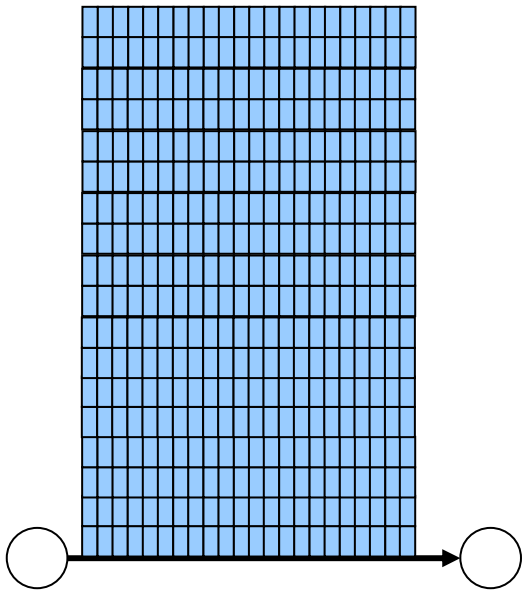
    →しかし情報量は集約・圧縮することでスケーラビリティは確保する i.e., 単純な tcpdump ではない
    確率的計測手法（ハッシュの利用）

Future Work (cont.)



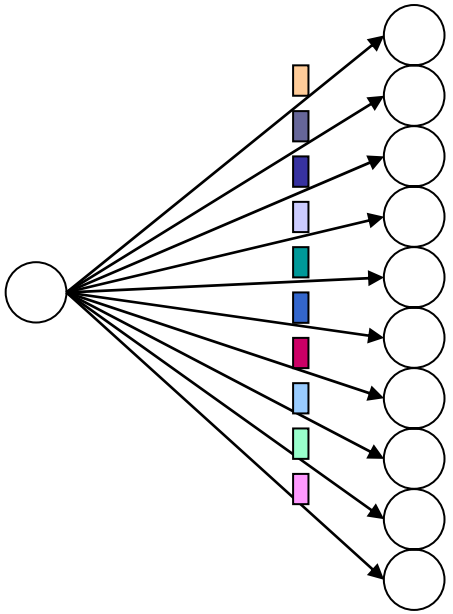
種類の数の例

種類の数=フロー数=１



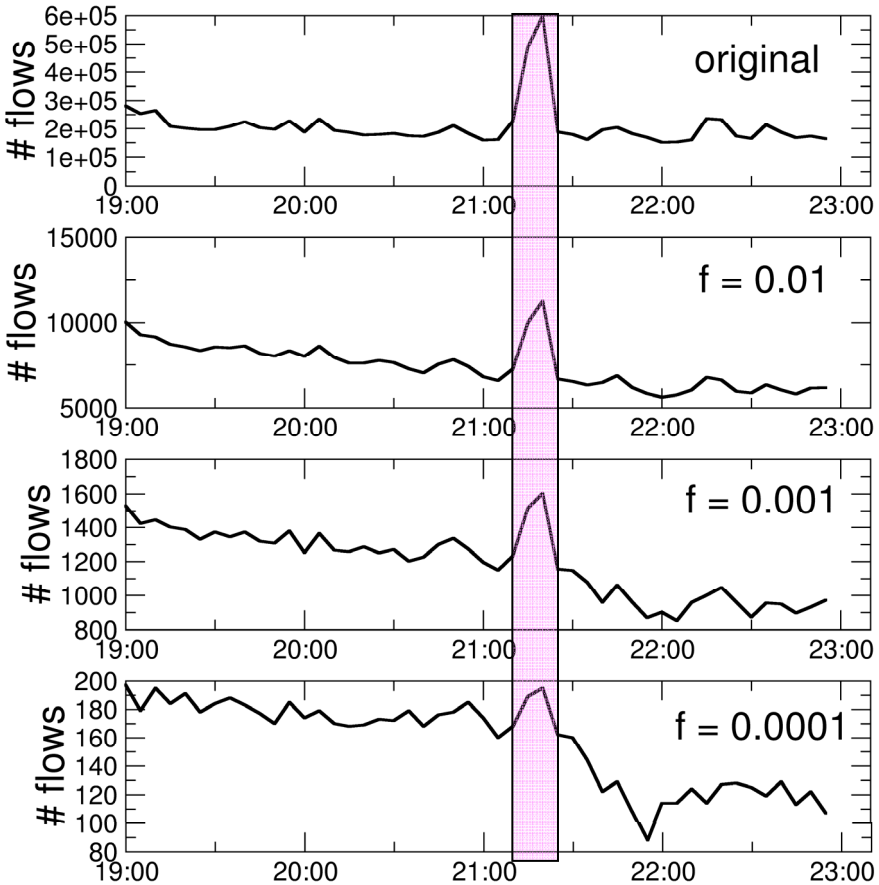
Heavy hitter

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）



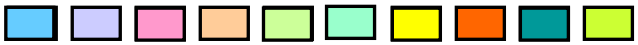
実際の観測値

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



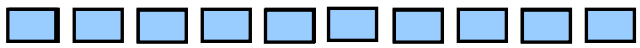
レート = 1/2 (=0.5) の
パケットサンプリング

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



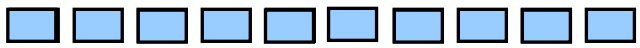
レート = 1/2 (=0.5) の
パケットサンプリング

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.