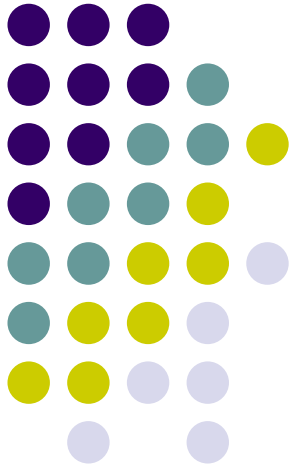
NetDelta: 大規模トラヒックデータの長時間・詳細分析方法

ネットワークシステム研究会

2006.9.15

森達哉(1)　石橋圭介(2)　上山憲昭(1)
川原亮一(1)　浅野正一郎(3)

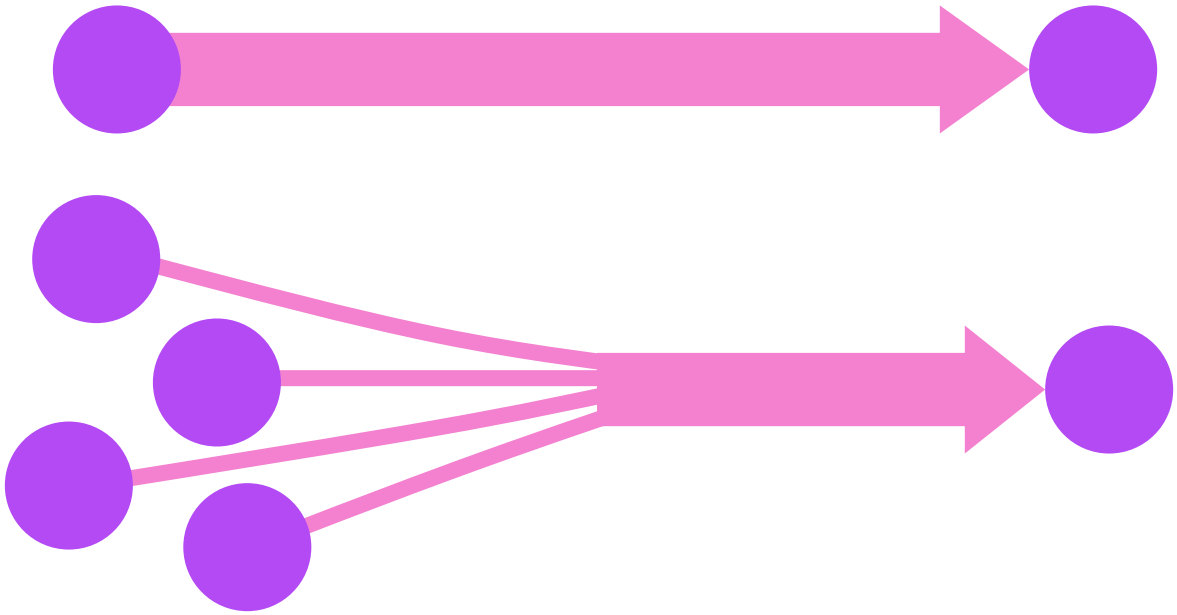
(1) NTTサービスインテグレーション基盤研究所
(2) NTT 情報流通プラットフォーム研究所
(3) 国立情報学研究所



NTT

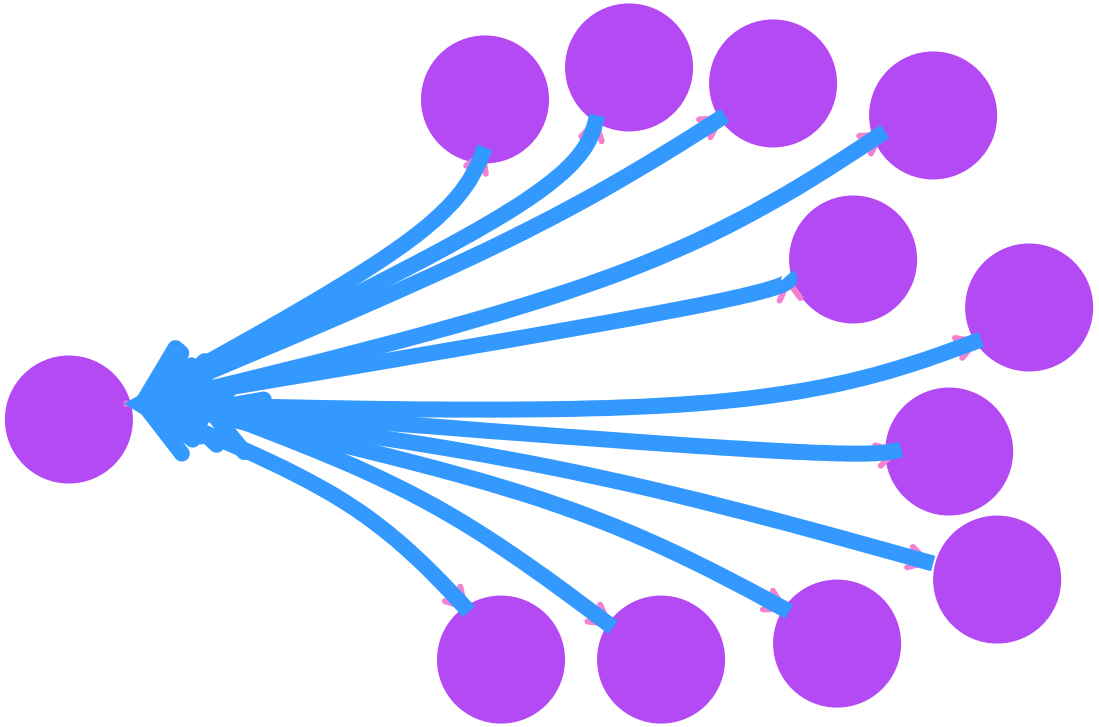
異常トラヒックとネットワーク計測

- Heavy-hitter, DDoS, 地域輻輳
  - 大容量転送・リクエスト、トラヒック集中
  - パケット数・バイト量 （累積値）



異常トラヒックとネットワーク計測

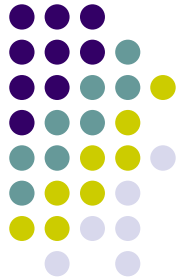
- ワーム拡散 (network scan), DDoS
  - Src/Dst IP などが分散
  - 異なる着IPアドレス数など （異なり数）



ネットワーク計測に求められる条件

- 詳細な分析
  - ホスト毎，ネットワーク毎(AS毎，dark space の union etc.) の異なり数など
- 長時間にわたる分析
  - 異常はいつおきるかわからない
  - 年単位の継続的な計測
- スケーラビリティ
  - データ量・処理速度
  - 例) サンプリングなしの生フローデータを年単位で持ち続けることはできない

本研究の目的



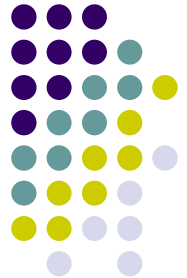
- ネットワークの長時間・詳細な計測を可能にする計測方法 ー NetDelta ー を提案する
  - Detailed and Long-Term Analysis

- 技術的ポイント:
  - NetFlow like な計測・コレクタ機能の分離

  - ビットマップを用いた異なり数のカウント
    → 既存技術の拡張として MACC (後述) を提案

  - ビットマップの論理演算による統計の集約

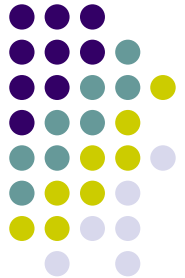
関連研究



- [Keys05] A Robust System for Accurate Real-Time Summaries of Internet Traffic, ACM SIGMETRICS 2005
  - 異なり数を含むホスト毎のトラヒックサマリを計測
  - 単一のグローバルBloom filter を用いることにより、異なり数を計測 (新規エントリ→カウントアップ)
- 本研究と異なる点
  - 計測した異なり数を集約することができない
  - 例) 5分毎に出力される異なり数 ➔ 60分では?
    ホストAとホストBに関する異なり数
    ➔ ホストA+B に関する異なり数は?
    ネットワークA, Bで観測した異なり数
    ➔ 双方で計測した重複のない異なり数は?

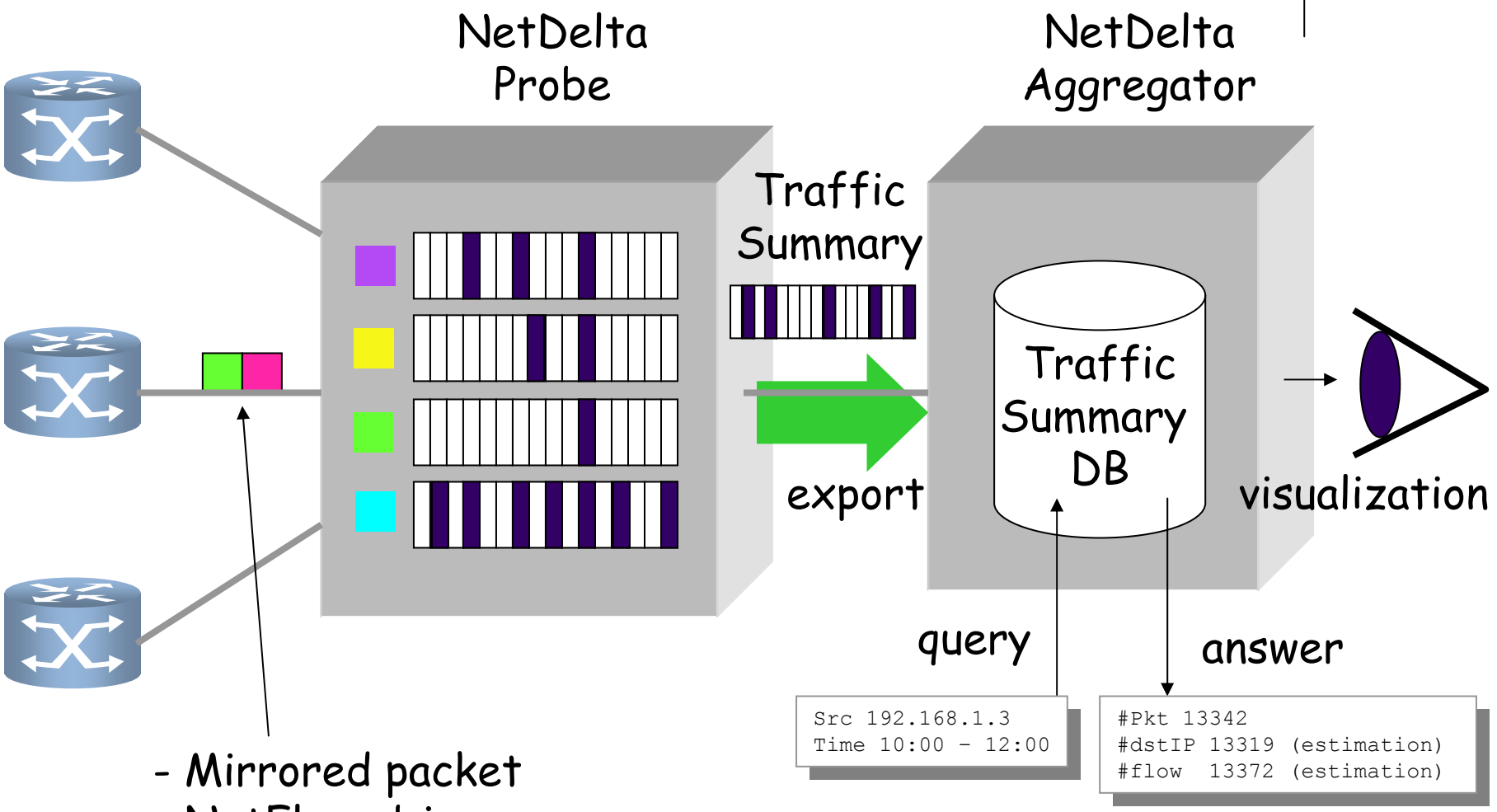
NTT

NetDelta の概要



- ホスト毎にトラヒックサマリを管理する
  - 累積値カウンタ
  - ビットマップ（異なり数）
    - トラヒックダイジェスト
- 定期的、あるいは使用メモリ量が閾値を超えたらトラヒックサマリを export
  - 固定長のメモリ量による管理を実現
  - あるホストのトラヒックサマリが分割する可能性がある

NetDelta のシステム構成の例



累積値のカウント

1

累積値のカウント

2

累積値のカウント

3

・インクリメンタルに加算できる

・必要なメモリ量は小さい

4 byte → 4,294,967,296

NTT

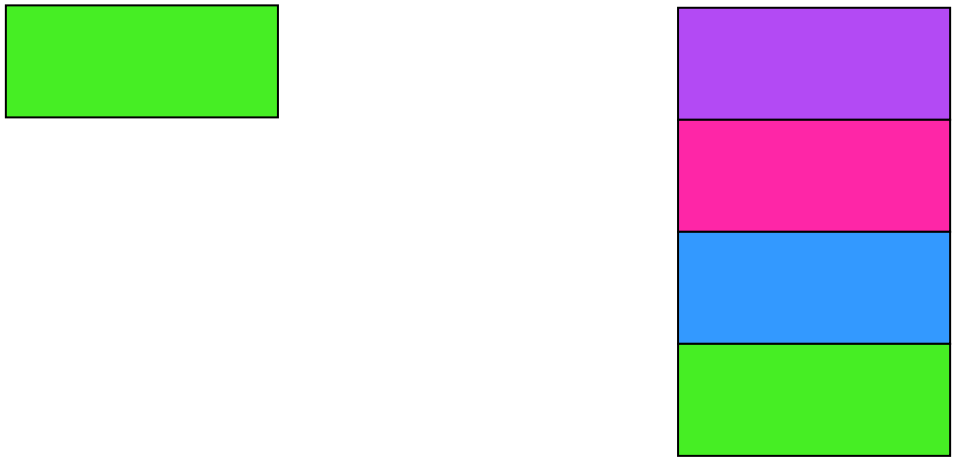
異なり数のカウント

異なり数のカウント

異なり数のカウント

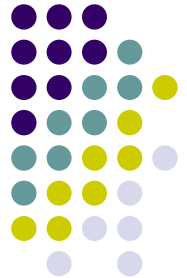
異なり数のカウント

異なり数のカウント



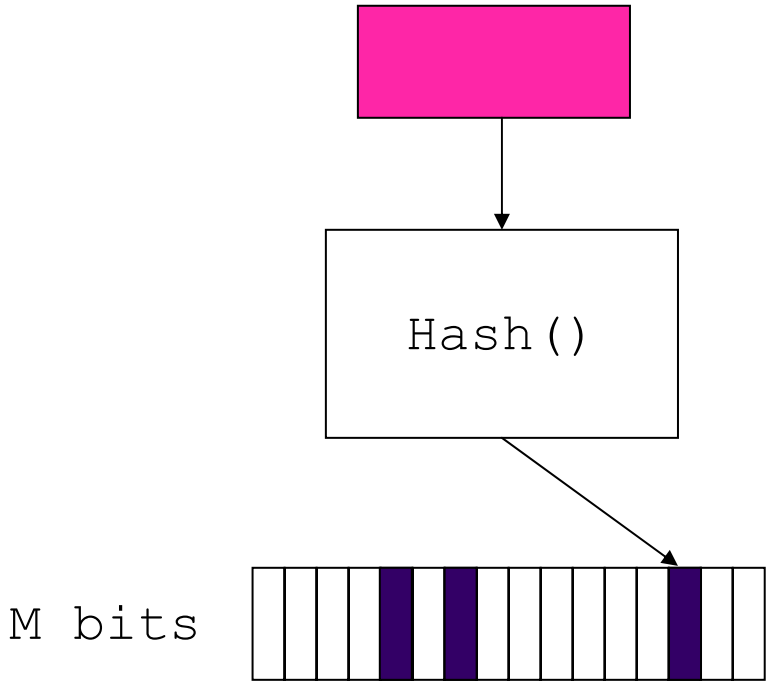
・ナイーブな方法ではすべてを記憶
➔ 必要なメモリ量が大きい
IPv4アドレスだとすると
4 byte × 異なり数

既存の異なり数推定方法



- Linear counting [Whang90]
  - Hash + Bitmap
  - 小さい異なり数も高精度に推定可能
  - 必要メモリ量は計測対象の異なり数Dに対しO(D)
- LogLog counting [Durand03]
  - Hash + Bitmap
  - 非常に大きな異なり数を推定可能
  - 必要メモリ量は計測対象の異なり数Dに対し O(log(log(D)))
- Adaptive cardinality counting [Cai05]
  - Linear Counting + LogLog Counting

Linear Counting の概要



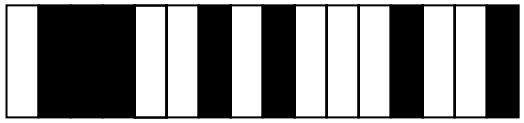
ビットが立っていない数: U

推定異なり数
$= -M \log(U/M)$

推定量の標準誤差
$\lesssim O(1/\sqrt{M})$

入力異なり数と必要なメモリ数が線形比例
➔ 大きな異なり数には向かない

Loglog Counting の概要



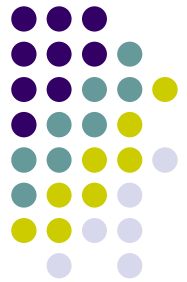
PFOB (最初に1が立つ位置)= x である確率 = 1 / 2^x
Max(PFOB) = m であるとき、異なり数の推定値 = 2^m

Max (PFOB) = 2 ➔ 推定異なり数 ≒ 4

※Loglog counting では複数Max (PFOB)の平均をとり精度をあげる

入力異なり数Dと必要なメモリ量MはM = O(log(log(D)))
の関係 → スケーラビリティに優れる
しかし小さな異なり数の計測には向かない

MACC (modified ACC)の概要



- Linear counting と Loglog counting を並列に運用し、両者の推定値のいいとこどりをする (ACCと同様)
  - 異なり数が小さいエリアは Linear Counting
  - 異なり数が大きいエリアは Loglog counting

- 両者を切り替えるポイント
  - ACCでは切り替えポイントで精度が劣化
  - 精度が劣化しないようなメモリサイズ、切り替えポイントの設計方法を提案した → MACC

MACCの管理するデータ

- Linear Counting 用に L ビットのビットマップ
- Loglog counting 用に各々が 5 ビットからなる M 個のバケット
  - M個の Max(PFOB)を管理する

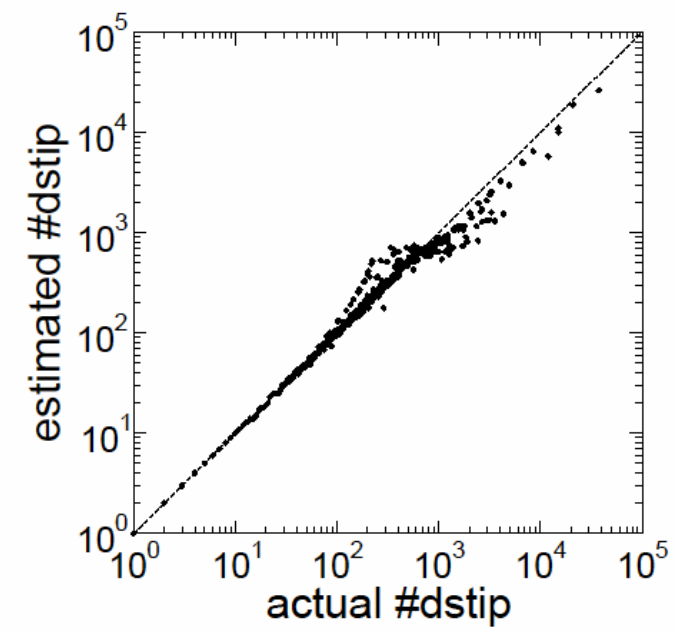
MACCの設計方法

- 下記の2条件で各々の計数法に必要となるメモリサイズ・切り替えポイントを決定
  - 許容誤差を満たすようにメモリサイズを決定
    - 理論式より、メモリ量と誤差の関係が求まる
  - 各々の推定手法を適切な範囲で用いる
    - Loglog counting は入力数が十分に大きいときのみ(複数バケットがすべて埋まったときのみ)
    - Linear counting はビットマップが埋まらないときのみ
    - 上記以外は各々の推定値の重み付け平均を用いることによってさらに精度の向上を図る

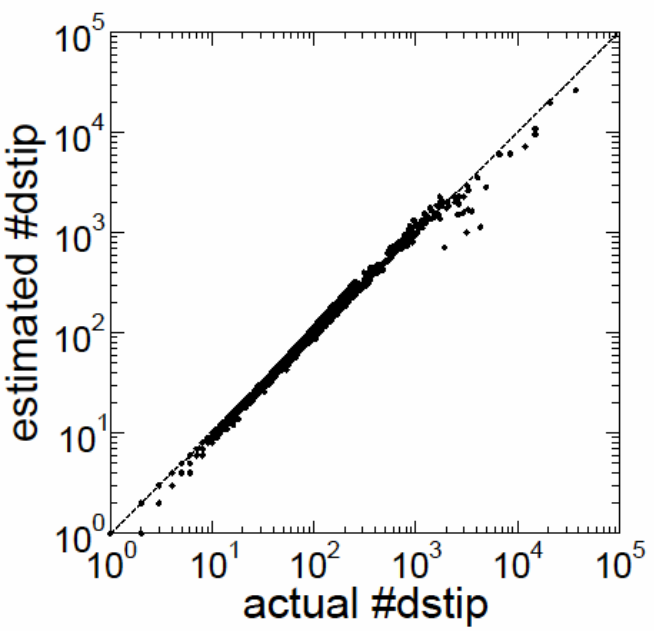
MACCとACCの比較結果

- SINET で計測した3分間のデータを利用
- srcIP あたりの異なる dstIP を評価
- ACCの方が多くメモリを使用する条件で比較
- ACC は切り替えポイントで精度が劣化

ACC



MACC

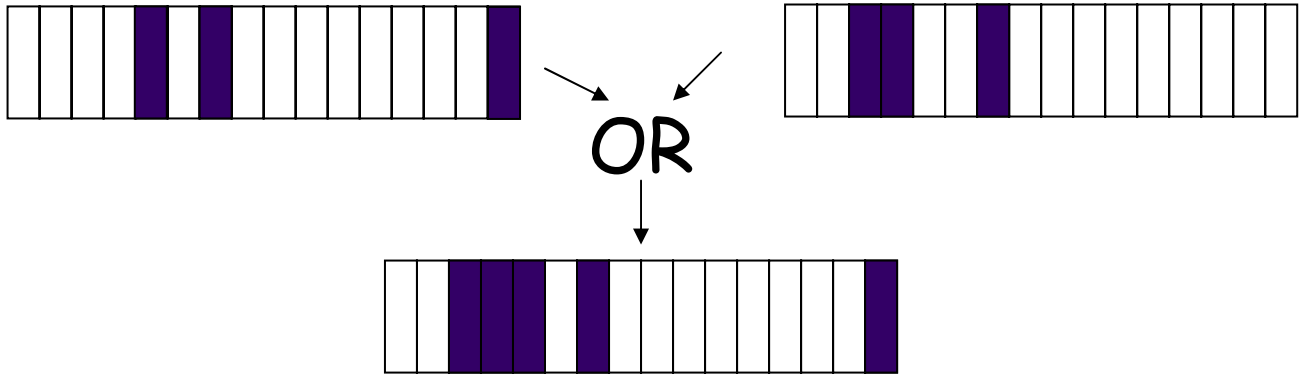


NTT

和集合のサイズの性質

- 複数のビットマップに記録した情報をビット演算によって集約できる
- 例) あるホストについて連続するインターバルA,Bで計測したビットマップ (linear counting)

- インターバルA+B での異なり数



和集合のサイズの性質

- 複数のビットマップに記録した情報を簡単な演算によって集約できる
- 例) あるホストについて連続するインターバルA,Bで計測したMPFOB (Loglog counting)
  - MPFOB(A), MPFOB(B)

- インターバルA+B での MPFOB
  - MPFOB(A+B) = max (MPFOB(A), MPFOB(B))

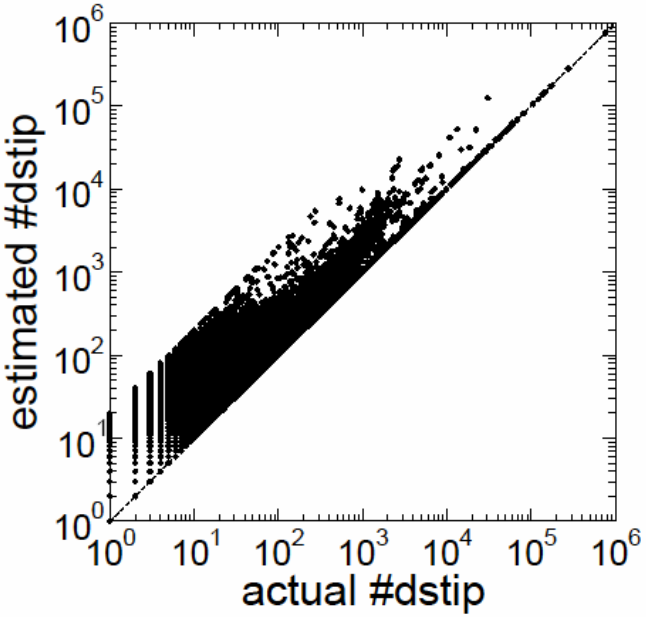
複数ビットマップ集約の応用

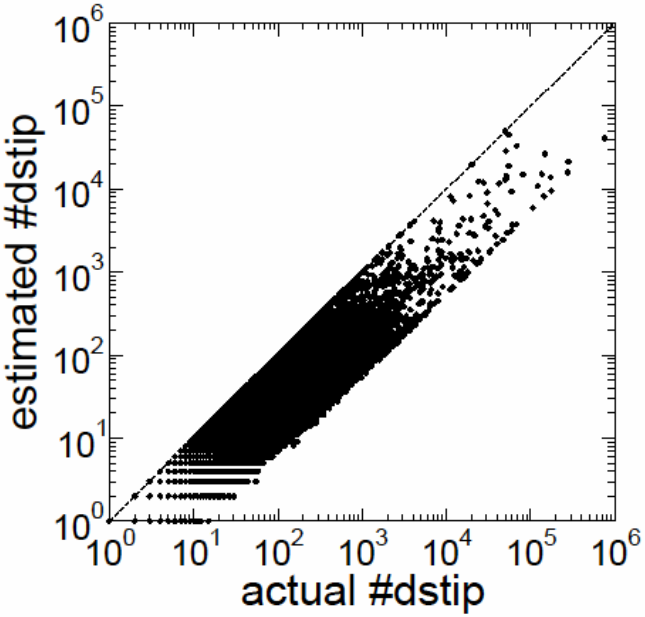
- 時間的な集約
  - 複数に分割したトラヒックサマリを集約可能
  - これにより、固定長のメモリによる実装を実現
  - あるホストの、2005/12/1～ 12/31 における発生フロー数の総数(重複を除く)は？

- 空間的な集約
  - 複数ホスト、複数計測ポイントのトラヒックサマリを集約可能
  - AS単位、prefix 単位、PoP単位・・・
  - 2006/3/1 に 129.60/16 のホストに対して通信したホストの総数(重複なし)の推定値は？

異なり数の集約: ビットマップを用いない場合

- 3分間毎のスロットで計測した異なり数データを1時間に集約するケース
- すべてのスロットでアイテムの重複なし (左)
- 異なり数が最大となるスロットに出現したアイテムが他のスロットに出現するアイテムをすべて包含する (右)

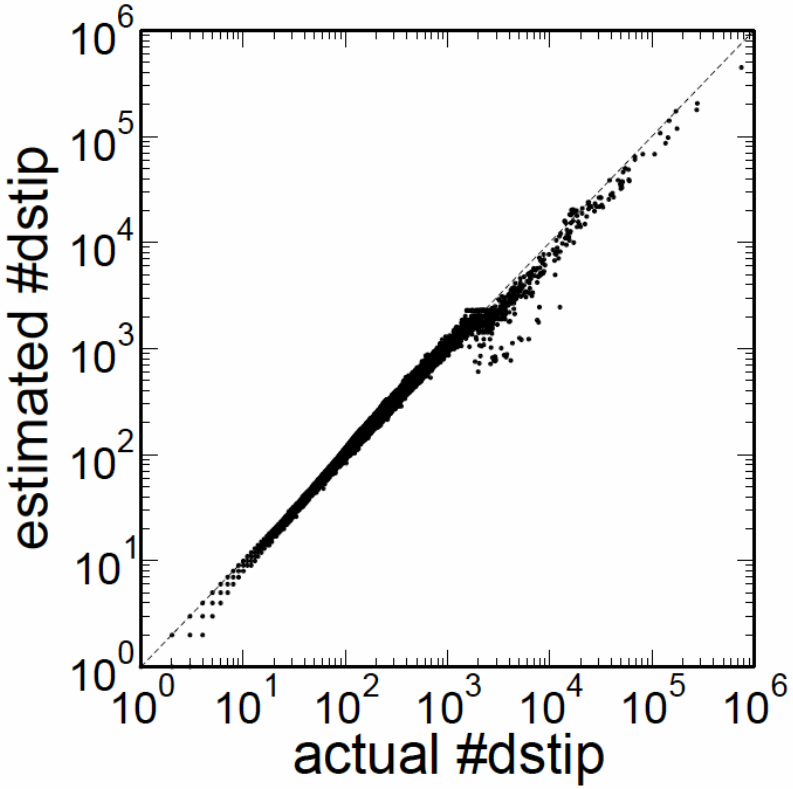




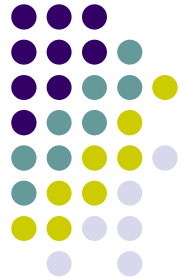
NTT

異なり数の集約: MACCを適用した場合

- 3分間毎のスロットで計測したデータを1時間に集約するケース
- 重複の情報がビットマップに保存されるため、正しく集約できる



まとめ



- ホスト毎のトラヒックサマリ集約方法NetDeltaを提案

- Hash + bitmap (MACC)による異なり数計数法を適用
  - Light weight measurement
    - Memory, one-pass processing
  - ビットマップの性質を用いることで、従来方法では不可能だった異なり数の集約が可能になる
  - 固定長メモリによるデータ管理
  - 時間的・空間的な集約

今後の課題

- コストに関する詳細な評価
  - メモリ量（probe）
  - メモリアクセススピード（probe）
  - トラヒックサマリの出力レート (probe → aggregator)
  - ストレージ容量　(aggregator)

- トラヒックサマリDBの応用
  - 大規模データの高速検索
  - データマイニング（過去にさかのぼる分析）
  - 可視化