

任意の時間粒度・空間粒度の検索が可能となる フローデータの格納方法の提案

辻 喜宏、大下裕一、村田正幸
大阪大学大学院 情報科学研究科

2014/3/8 IEICE IN研究会 1

細粒度のトラフィックエンジニアリング

- ・ 予測を用いたトラフィックエンジニアリング (TE)
 - ・ 目的: トラフィック変動量が大きい近年のネットワーク内の安定的なトラフィック収容
 - ・ 過去から現在にわたって様々な時間粒度の時系列データが必要
- ・ OpenFlowを用いた TE
 - ・ 目的: 柔軟に定義できるフロー単位での経路制御
 - ・ プレフィックス長込みの IP アドレスやポート番号などの様々な空間粒度指定されるフローデータが必要

2014/3/8 IEICE IN研究会 2

研究目標

- ・ トラフィック観測の要件
 - ・ 任意の空間粒度・時間粒度でのトラフィック情報を取得できる
 - ・ 時間粒度: タイムスロット単位で時間範囲を指定
 - ・ 空間粒度: 1bit 単位での送信元・宛先 IP アドレスのプレフィックス指定
- ・ 観測機器構成に関する制約
 - ・ 情報取得は TE の制御周期以内に行われるべき
 - ・ 情報更新は観測の最小時間単位 (タイムスロット) 以内に行われるべき
 - ・ 観測器はできるだけ少ない資源で達成されるべき

↓

任意の時間粒度・空間粒度の検索が可能となる
フローデータ格納方法についての議論を行う

2014/3/8 IEICE IN研究会 3

フローデータ格納方法の概要

- ・ 木構造による階層的な粒度管理
 - ・ 根からの深さで粒度の細かさを表現 (親ノードは子ノードを集約したものになる)
- ・ 複数の木構造を連結し任意の粒度管理
 - ・ 一つのキーは一つの木構造に対応
 - ・ 複数の木構造を次のように連結しフロー集合とそれに対応するトラフィックカウンタを管理
「時間 - 送信元 IP - 宛先 IP - トラフィックカウンタ」

↓

各キーについて任意の粒度を指定でき、
任意の粒度のフロー情報検索が可能となる

- ・ パトリシア木を用いて最小限のノード数での粒度管理

↓

データ構造のデータサイズや参照・更新の側面から効率的

2014/3/8 IEICE IN研究会 4

データ構造の参照: フロー情報の検索

- ・ フロー情報の検索
= 該当粒度のトラフィックカウンタの参照
- ・ 参照手順
 1. 一つのキーの粒度検索 (左子 / 右子を選択し下降)
 - A) 探索キーとノードキーを比較
 - B) より細かい粒度を要求するならば子ノードへ遷移
 - C) ノードキーの方が細かい粒度ならば終了
 2. キーの切り替えを行い木構造であれば 1へ (次の木構造へ遷移)
 3. 該当トラフィックカウンタの取得

2014/3/8 IEICE IN研究会 5

データ構造の更新: フロー情報の追加

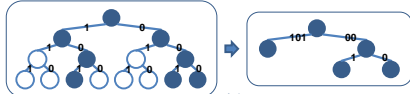
- ・ フロー情報の追加
= 該当粒度以上のトラフィックカウンタの更新
- ・ 更新手順
 1. 一つのキーの粒度検索 (左子 / 右子を選択し下降)
 - A) 探索キーとノードキーを比較
 - B) より細かい粒度を要求するならば子ノードへ遷移、このとき同時に通過したノードを記録
 - C) ノードキーが最細粒度に相当するならば終了 (アノードが持つキー (size))
 2. 手順1で通過した各ノードについてキーの切り替えを行い木構造であれば手順1へ (次の木構造へ遷移)
 3. 該当トラフィックカウンタの更新

2014/3/8 IEICE IN研究会 6

パトリシア木を用いた粒度管理

- ・パトリシア木
 - ・根ノードからの内部ノードまでの経路でキー、内部ノードでそのキーに対応する値を保持
 - ・バイナリ集合の要素間の違いを木構造の分岐によって実現
- ・パトリシア木のデータサイズ
 - ・一つのノードを追加する上で新たに必要となるノード数は高々2で、線形増加で抑えることができる

取りうるビット空間に対して観測されるビットパターンがスパースであることが予想される今回のデータとは相性が良いと考えられる



2014/3/8

IEICE IN研究会

7

評価環境

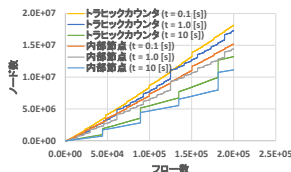
- ・評価目的
 - ・提案構成で制約の時間以内に更新・検索ができ、データサイズがどれくらいになるかを調査
- ・評価項目
 - ・データ構造のデータサイズ、フロー情報追加時間、フロー情報検索時間
- ・トラヒック観測器の実行環境
 - ・計算機: Intel(R) Xeon(R) の CPU、32GB のメモリを搭載
- ・トラヒックデータ
 - ・観測地点: 大阪大学のゲートウェイ
 - ・観測日時: 2014年 2月 7日 12:00 ~ 13:00
 - ・観測パラメータ: タイムスロット長 (t 秒) として、 $t = 0.1, 1.0, 10$ の三種類

2014/3/8

IEICE IN研究会

8

データサイズ



1時間分のトラヒック情報保存に必要なデータサイズ (GByte)

| タイムスロットの長さ | 0.1 [s] | 1.0 [s] | 10 [s] |
|------------|---------|---------|--------|
| 必要な領域のサイズ | 243.4 | 113.0 | 49.5 |

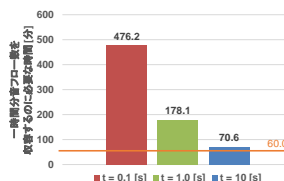
- ・追加フロー数に対して木構造のノード数は線形的に増加
- ・タイムスロットの長さが小さくなると、一つのフロー追加あたりに増加するノード数は大きくなる
- ・タイムスロットが10 [s] であると1台の計算機で処理できるデータサイズである

2014/3/8

IEICE IN研究会

9

フロー情報追加にかかる時間



- ・タイムスロットの長さが小さくなると、ノード数が大きくなるため、一つのフロー追加あたりに更新するトラヒックカウンタ数は大きくなり時間も同様
- ・タイムスロットが10 [s] でも追加にかかる時間は70.6 [min] 必要で一時間以上の計算時間がかかってしまっている

2014/3/8

IEICE IN研究会

10

フロー情報追加時間の削減方法

- ・フロー情報追加にかかる時間はフロー到着間隔よりも短くあることが理想

方針

- ・更新するノード数を減らす

具体案

兄弟ノードが保持する情報の削減

- ・削除された弟ノードの情報は親ノードと兄ノードの情報から計算できる
- ・各木構造で半数のノード削減が期待できる
- ・情報検索が十分高速な今回の場合は有効であると考えられる



2014/3/8

IEICE IN研究会

11

まとめと今後の課題

まとめ

- ・任意の粒度検索が可能となるトラヒック観測機器を提案した
- ・実現するために、実データを用いてデータサイズやデータ構造の更新・検索の観点からの評価を行い、問題点とそれを解決するための方針を整理した
- ・タイムスロットの長さの違いによりどのような特性が現れるかを調査した

今後の課題

- ・木構造のノード数の削減手法の実装・評価
- ・トラヒック観測機器の間の連携
- ・並列化についての議論

2014/3/8

IEICE IN研究会

12