

NDNルータにおけるFIBの構成法による比較評価

1. 研究目的

- 情報指向ネットワーク(ICN)の転送テーブル(FIB: forwarding information base)のメモリ消費量と検索時間低減の為に、コンテンツの構成法を変えることでFIBを集約する方式が提案
→ **BloomFilterを用いたFIBの構成**
- FIB構成法にはハッシュテーブル、パトリシア木、BloomFilterを用いた構成が広く検討されているが、有効性が明らかにされていない

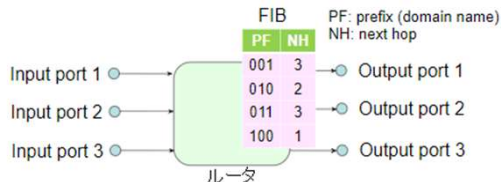


BloomFilterを用いてFIBを構成し、**メモリ消費量、検索時間低減の効果を明らかにする**

2. FIBとは

- 各ルータのPrefix (識別子)とNH (next hop)の対応表

ルータは到着パケット毎にFIBを参照し、Prefixに対応するNHにパケットを転送



- 従来のインターネット:

目的ホストのサブネットワークアドレスをPrefixに使用

- ICN:

コンテンツ名をPrefixとしてFIBを構成



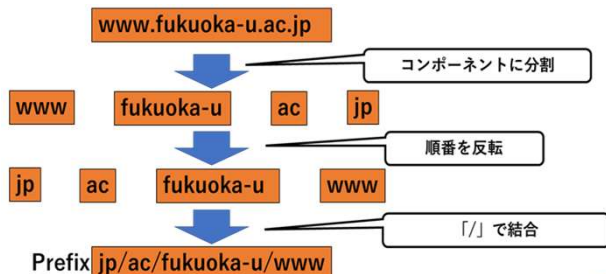
サブネットワーク数と比較して遥かに数が多い

⇒ FIBで必要とするエントリ数が遥かに多い

FIBサイズの低減が重要

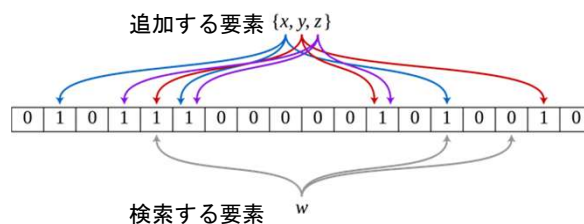
3. コンポーネントとPrefixの構成

- コンポーネント: コンテンツ名(URL)の「.」で区切られた各パート



4. BloomFilter

- 大量のデータから特定の要素が存在するかどうかを高速に判定するための確率的データ構造
- メモリ効率がよく、検索速度が早い
- 固定サイズのビット配列と複数のハッシュ関数から構成
- 要素を追加
それぞれのハッシュ関数を適用し、ビット配列に該当する位置を1に設定



- 要素を検索
同様のハッシュ関数を適用し、ビット配列に該当する位置を調べる
- いずれかの位置が1でない場合
→ **要素は存在しない**
- 全ての位置が1である場合
→ **要素が存在する可能性が高い**
- 全てが1であっても要素が存在しない可能性(**偽陽性**)
- 偽陽性の確率
配列のビット数m、追加する要素数n、ハッシュ関数の数kとすると、以下の式で表される

$$\left(1 - \left(1 - \frac{1}{m}\right)^{kn}\right)^k \approx \left(1 - e^{-kn/m}\right)^k$$

- 配列のビット数mを増加すると発生率**低下**
- 追加する要素数nを増加すると発生率**上昇**
- 偽陰性はない
一度設定されたビットは勝手に変更されることはない、該当するビットの一つでも0が存在すれば要素は存在しないと断定可能
- 要素の削除は不可能
同じビットに異なる要素を表している可能性あり

5. 今後の予定

BloomFilterでFIBを構成し検索時間、メモリ消費量について評価する
→ 3つのNHを持つある1つのノードにおいてNH毎にFIBを用意しルーティングを決定するプログラムを作成