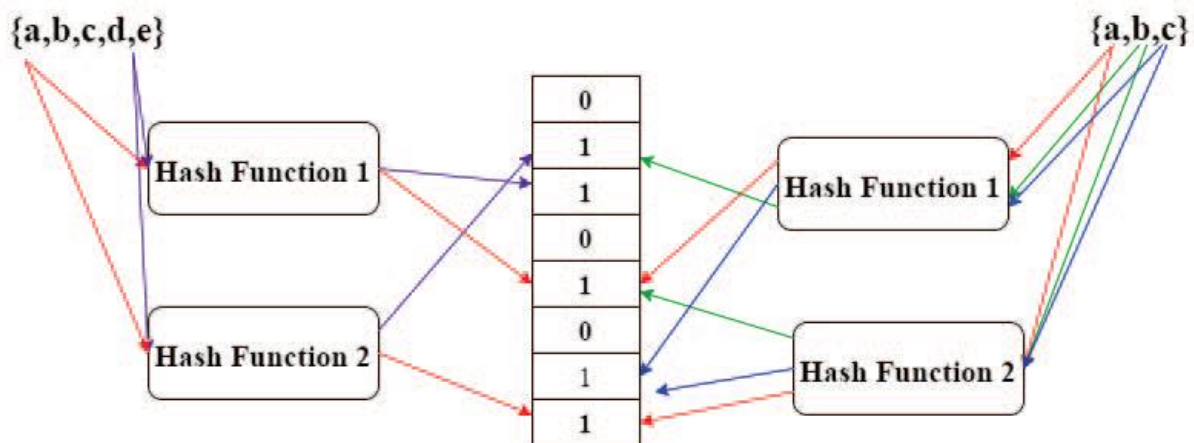


Bandwidth Reduction in SNMP Monitoring System with Bloom Filter using Lossless Compression

1. Metode

Bloom Filter

Dalam paper ini penulis menggunakan metode Bloom Filter yaitu dengan mengacak bit array dari suatu data. Bit Bloom Filter pada awalnya berisi 0 di semua lokasi.. Ketika nilai anggota melewati Hash 1 dan hash 2, maka hasilnya disimpan pada bit array dengan mengubah dari 0 ke 1 pada indeks yang tersedia.



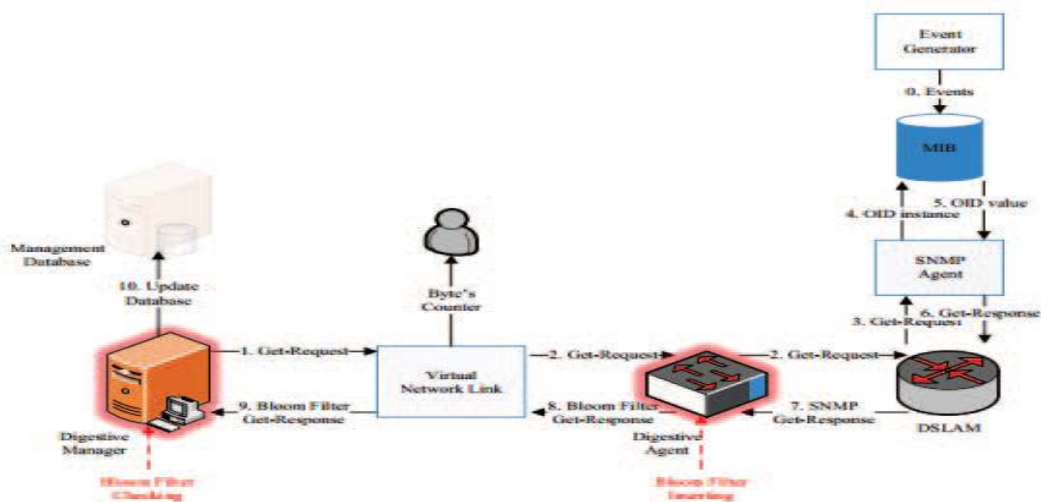
Di sisi kanan tunjukkan diatur dengan anggota $\{a, b, c\}$, menunjukkan proses pemeriksaan Bloom Filter, itu harus menggunakan fungsi Hash yang sama 1 dan Hash function 2, untuk memeriksa setiap anggota yang ada di dalamnya sedikit array Filter Bloom. Perhatikan bahwa jumlah hash yang dipilih fungsi dapat mempengaruhi tingkat kesalahan positif palsu sedangkan ukuran yang dipilih dari masing-masing anggota di set tidak berpengaruh dengan ukuran.

2. Approach

Lossless Compression

Kompresi Lossless merupakan metoda kompresi data yang memungkinkan data asli dapat disusun kembali dari data hasil kompresi maka rasio kompresi pun tidak dapat terlalu besar untuk memastikan semua data dapat dikembalikan ke bentuk semula. Metode Lossless menghasilkan data yang identik dengan data aslinya. Paper ini cara terbaik untuk mengurangi penggunaan bandwidth di SNMP dengan Bloom Filter adalah dengan menerapkan kompresi lossless.

3. Framework



Penjelasan dari gambar di atas:

Gambar diatas menunjukkan sistem jaringan SNMP konvensional dengan modul Bloom Filter sebagaimana diimplementasikan. Untuk menggunakan Filter Bloom dalam SNMP konvensional. Bloom Filter digunakan sebelum SNMP memasuki jaringan dan terjadi pemeriksaan Bloom langsung dari jaringan.



Sekarang untuk lebih menekan Bloom Filter yang digunakan pada paper ini, agen enkripsi ditempatkan setelah memasukkan Bloom Filter dan agen dekripsi ditempatkan sebelum Filter Bloom memeriksa.

4. Hasil Framework

The actual size of input group (byte)	Group name
$2^{17} - 2^{16} = 65536$	Group 1
$2^{18} - 2^{16} = 196608$	Group 2
$2^{20} - 2^{16} = 983040$	Group 3
$2^{22} - 2^{16} = 4128768$	Group 4
$2^{24} - 2^{16} = 16711680$	Group 5
$2^{26} - 2^{16} = 67043828$	Group 6
$2^{28} - 2^{16} = 268369920$	Group 7
$2^{30} - 2^{16} = 1073576288$	Group 8

Percobaan ini menguji seberapa besar rasio kompresi yang dapat dicapai dengan memanfaatkan kompresi zip. Lebih besar tingkat ransum itu adalah penggunaan bandwidth yang lebih rendah. Penulis membuat set vektor bit dengan ukuran input yang berbeda dan memberi mereka masukan modul kompresi kemudian membandingkan kompresi.