Data Structure Enabling Retrieval of Time Series of Traffic with the Requested Granularity

Yoshihiro Tsuji, Yuichi Ohsita, Masayuki Murata
Graduate School of Information Science and Technology, Osaka University, Japan

Background
• The time variation of internet traffic is increasing
  - Owing to advent of streaming, cloud service, etc.
• Traffic control with predicted traffic information has been studied [1]
  - Time series of traffic is required to predict future traffic
  - Required granularity of traffic information may change in time
  - The dynamic routing control requires following traffic information
  - Temporal granularity: various data size in time series
  - Spatial granularity: various aggregated flows

Simple approach to obtaining traffic information
• Steps
  - Monitor traffic data per flow at each router
  - Obtain traffic data from all routers periodically
  - Calculate the obtained traffic data with the required granularity
• Problem:
  It takes a large overhead to collect all traffic data

Architecture to obtain traffic information
• Multiple traffic observers are deployed over the network
• Each traffic observer collects traffic observation from nearby routers
• Traffic observer returns the time-series of traffic
• Network manager requests time series of traffic information with the required granularity
• This approach reduces overhead, because only the information with the required granularity is sent

Requirements for traffic observer
• Traffic observer should store the traffic information with the data structure satisfying
  - Immediate update (< one time slot)
  - Immediate retrieval (<a few seconds [2])
  - Small size (< the size of HDD)
• Research Goal:
  Data structure suitable for the traffic observer

Overview of our data structure
• Each tree maintains the granularity of a field
• Each node has a pointer to the root of the next tree
• Parent nodes correspond to the coarser granularity that includes all of its children
• Each node has a pointer to the root of the next tree
• Leaf nodes correspond to the finest granularity
• In this data structure, the traffic information with the required granularity is obtained by only retrieving the trees


Each bit information subtraction in IP tree

- We use approaches to compressing the tree

Nodes’ information subtraction in IP tree

- If we request as frequent time slots 12 hours, 1 hour 10 minutes, 1 minute and 10 seconds

N-ary tree

- We prepare the node corresponding to the time slots whose lengths are likely to be requested

**IP tree using Patricia tree**

- Traffic information monitored by each traffic observer includes only limited IP addresses

- We use Patricia tree [3] as the IP tree.

- Nodes with only one child are eliminated from the tree

**Source/Destination IP tree**

- Each level of tree indicates the number of bits of IP address prefix

- Problem: Complete binary tree requires too many nodes

- We use approaches to compressing the tree

**Steps to update**

- Update observed flow

  (time=3~4, source IP address=110, destination IP address=011)

- If newly observed entry, no additional processing

- If newly observed entry, add two new nodes

- If already observed entry, update all destined pointed to by nodes

**Steps to retrieve**

- Nodes’ information subtraction in IP tree

  - Trees’ information subtraction

  - Trees’ information subtraction is realized by the multiple subtractions of corresponding nodes

  - Retrieve requested flow

**Additional approach to reduction of nodes**

- Eliminating sibling node information

- The information is recovered by calculating recursively the difference of the parent node and the other sibling node

- Setting threshold (s), for not holding coarse granularity

  - It is realized by combining N-ary tree \((N = 2^n)\) around root of the tree

- Retrieve subtree’s information

- If restorations are needed, calculate recursively the subtree’s information

- Retrieve all source IP trees

- Update observed flow

- Retrieve all source IP tree pointed to by nodes in the set

- Retrieve all counters pointed to by nodes in the set

- Retrieve all destination IP trees

- Retrieve all subtree corresponding to the requested flow

- Find the sequence corresponding to the requested flow

- Find the subtree corresponding to the requested flow

- Retrieve all IP address pointed to by nodes in the set

- Retrieve all counters pointed to by nodes in the set

- Retrieve all destination IP trees

- Retrieve all source IP trees

- Update all source IP tree pointed to by nodes

- Update all destination pointed to by nodes

- Time

  - Source IP Address

  - Destination IP Address

  - Traffic information

  - Time

  - Source IP Address

  - Destination IP Address

  - Traffic information

  - Time

  - Source IP Address

  - Destination IP Address

  - Traffic information

  - Time

  - Source IP Address

  - Destination IP Address

  - Traffic information

  - Time

  - Source IP Address

  - Destination IP Address

  - Traffic information

  - Time

  - Source IP Address

  - Destination IP Address

  - Traffic information

  - Time

  - Source IP Address

  - Destination IP Address

  - Traffic information

  - Time

  - Source IP Address

  - Destination IP Address

  - Traffic information

  - Time

  - Source IP Address

  - Destination IP Address

  - Traffic information

  - Time

  - Source IP Address

  - Destination IP Address

  - Traffic information

  - Time

  - Source IP Address

  - Destination IP Address

  - Traffic information

  - Time

  - Source IP Address

  - Destination IP Address

  - Traffic information

  - Time

  - Source IP Address

  - Destination IP Address

  - Traffic information

  - Time

  - Source IP Address

  - Destination IP Address

  - Traffic information

  - Time

  - Source IP Address

  - Destination IP Address

  - Traffic information

  - Time

  - Source IP Address

  - Destination IP Address

  - Traffic information

  - Time

  - Source IP Address

  - Destination IP Address

  - Traffic information

  - Time

  - Source IP Address

  - Destination IP Address

  - Traffic information

  - Time

  - Source IP Address

  - Destination IP Address

  - Traffic information

  - Time

  - Source IP Address

  - Destination IP Address

  - Traffic information

  - Time

  - Source IP Address

  - Destination IP Address

  - Traffic information

  - Time

  - Source IP Address

  - Destination IP Address

  - Traffic information

  - Time

  - Source IP Address

  - Destination IP Address

  - Traffic information

  - Time

  - Source IP Address

  - Destination IP Address

  - Traffic information

  - Time

  - Source IP Address

  - Destination IP Address

  - Traffic information

  - Time

  - Source IP Address

  - Destination IP Address

  - Traffic information

  - Time

  - Source IP Address

  - Destination IP Address

  - Traffic information

  - Time

  - Source IP Address

  - Destination IP Address

  - Traffic information

  - Time
Evaluation environment

We evaluated our data structure using actual traffic trace

- Used traffic trace
  - Monitoring Period: 2014/02/07 12:00 – 12:30
  - Monitored point: gateway at Osaka university
  - Line speed: 1Gbps

- Evaluation metrics
  - Data size
  - Update time
  - Retrieval time

Evaluation: data size

- The large s reduces the size of data
- Even if s=4, data size is less than 4.5GB

Evaluation: update time

- It takes only 8 seconds to store all flows monitored within 60 seconds
- Eliminating sibling node information reduces the time required to update the traffic information by half
  - Because the number of nodes required to be updated is reduced.

Evaluation: retrieval time

- Retreiving the finer granularity requires a larger time
  - Because more nodes must be retrieved
- Even if eliminating sibling node information, the required time to extract is less than 7.0ms

Conclusion & future work

- Conclusion
  - We proposed a data structure enabling the quick retrieval of the time series of the traffic with the requested granularity
- Future work
  - Discussion on more reduction of the nodes in the trees so as to save the memory size and the update time

Thank you for your listening