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

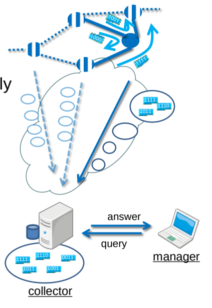
Background

- 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

[1] Tatsuya Otoshi, "Prediction-based control theoretic approach for robust traffic engineering," Master's thesis, Graduate School of Information Science and Technology, Osaka University, February 2014.

Simple approach to obtaining traffic information

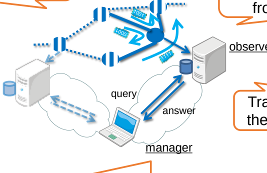
- Steps
  - Monitor traffic data per flow at each router
  - Obtain traffic data from all routers periodically
  - Calculate the obtained traffic data with the required granularity
- Problem:
  It takes a large overhead to collect all traffic data



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

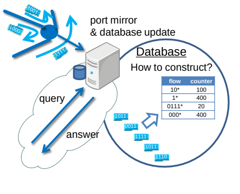
Network manager requests time series of traffic information with the required granularity

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



[2] T. Benson, A. Anand, A. Akella, and M. Zhang, "MicroTE: Fine Grained Traffic Engineering for Data Centers," in Proc. ACM CoNEXT, 2011, pp. 8:1–8:12

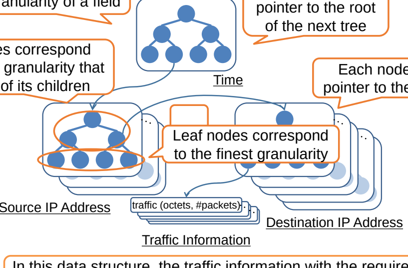
Overview of our data structure

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 counter



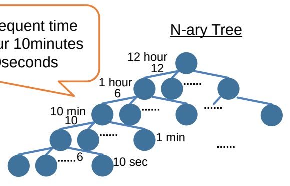
Leaf nodes correspond to the finest granularity

In this data structure, the traffic information with the required granularity is obtained by only retrieving the trees

Time tree

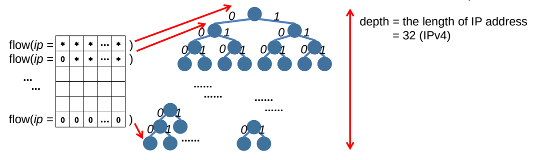
- N-ary tree
  - We prepare the node corresponding to the time slots whose lengths are likely to be requested

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



Source/Destination IP tree

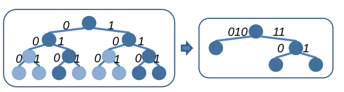
- Binary tree
  - Each level of tree indicates the number of bits of IP address prefix



- <u>Problem</u>: Complete binary tree requires too many nodes
  - We use approaches to compressing the tree

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

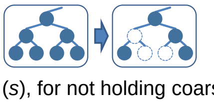


complete binary tree Patricia tree

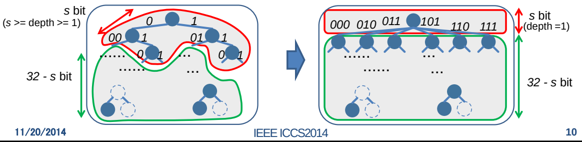
[3] D. E. Knuth, "The Art of Computer Programming. vol. 3: Sorting and Searching," Atmospheric Chemistry & Physics, vol. 1, 1973.

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

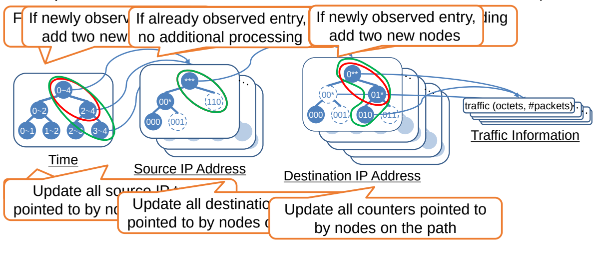


- Setting threshold (s), for not holding coarse granularity
  - It is realized by combining N-ary tree ($N = 2^s$) around root of the tree



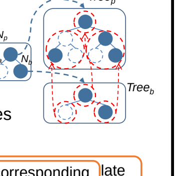
Steps to update

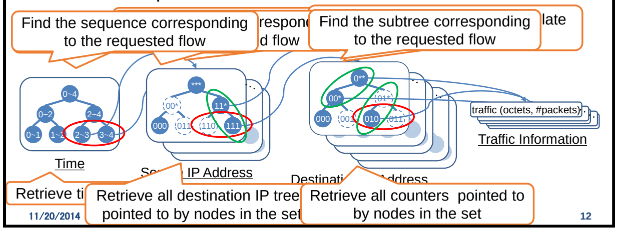
- Update observed flow
  (time=3~4, source IP address='110', destination IP address='011')



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

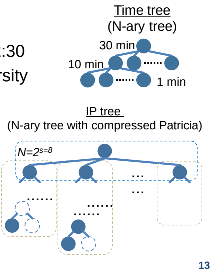




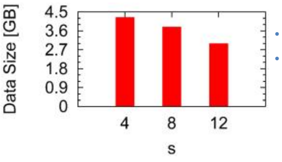
Evaluation environment

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

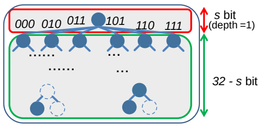


Evaluation: data size

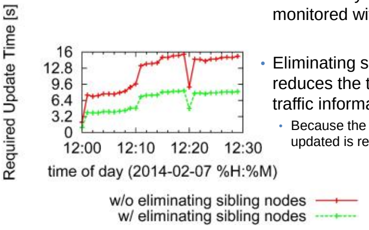


- The large s reduces the size of data
- Even if *s=4*, data size is less than 4.5GB

Parameter S:
the length of prefix whose corresponding information is eliminated



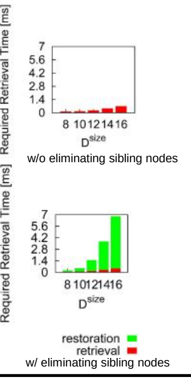
Evaluation: update time



- 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.

[4] Micron, "Micron Technical Note Using DDR4 in Networking Subsystems," http://www.micron.com/-/media/documents/products/technical%20note/dram/tn_4003_ddr4_network_design_guide.pdf.

Evaluation: retrieval time



Generated query
- Request all information of traffic
  - Generated within a 10-min time slot
  - Whose 8-bit prefix equals the randomly defined one
- Required granularity
  - Time slot of traffic information: 1 min
  - The length of prefix of IP address: 8, 10, 12, 14, and 16

- Retrieving 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