

Scalable and Efficient Data Streaming Algorithms for Detecting Common Content in Internet Traffic

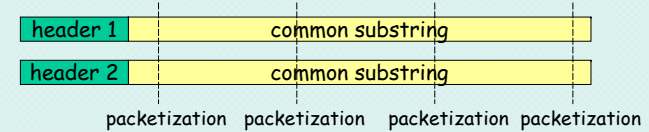
Minho Sung

Networking & Telecommunications Group
College of Computing
Georgia Institute of Technology

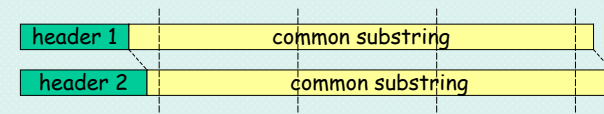
Joint work with A. Kumar(Georgia Tech), L. Li (Bell lab),
J. Wang(AT&T - Research) and J. Xu(Georgia Tech)

Problem statement

- Aligned case (e.g. Web browsing, CodeRed worm)



- Unaligned case (e.g. Email worms such as Nimda or Sircam)

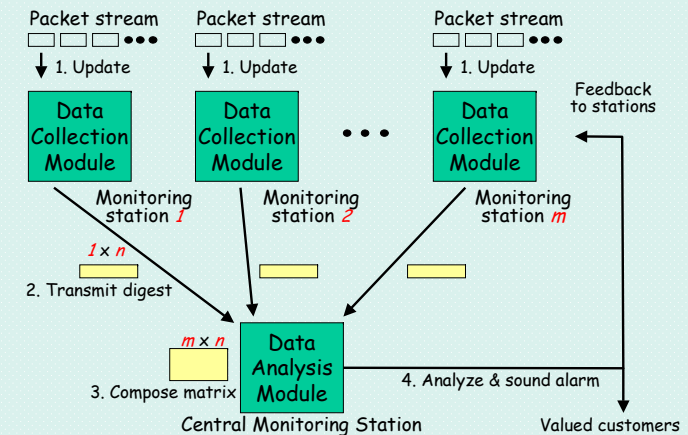


- Assumption : fixed MTU size is used
 - few popular packet sizes in Internet (> 500) (e.g. 576,1500)
 - our algorithm can be extended later

Monitoring and analyzing aggregate traffic

- Essential for detecting "global" events
 - important in network measurement/management
- We focus on : detecting common content
 - motivations
 - early stage detection of Internet worms/viruses/spam emails
 - flash crowd event detection in web browsing or P2P traffic
 - challenge : hard to detect by per-link monitoring
 - signal is usually too weak to be detected locally
 - correlation of traffic among many links is required
 - our approach : distributed data streaming

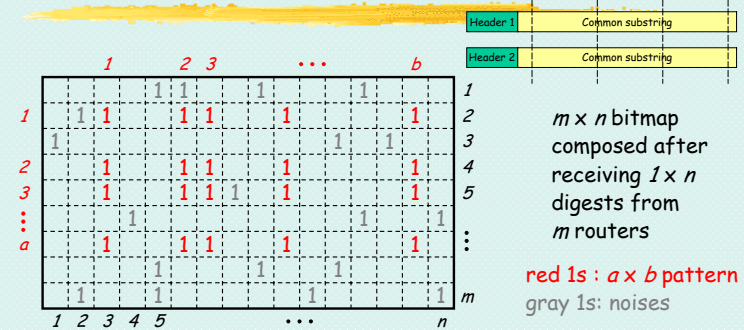
Solution framework



Challenges & Requirements

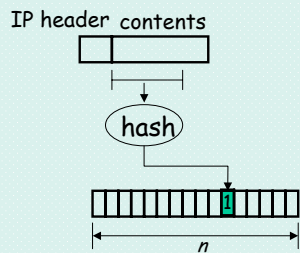
- Data Collection Module
 - fast enough for high-speed links
 - much smaller size of digests than original traffic
 - digests should contain enough information
- Data Analysis Module
 - low computational complexity
 - high accuracy in identifying patterns

Aligned case : Data Analysis Module



- $a \times b$ submatrix : common content of b packets seen by a points
- problem of finding the largest all-1 submatrix in general case
 - equivalent to "maximum edge biclique problem" : known to NP-hard
- our case is more tractable : each element is drawn by Bernoulli trial
 - our polynomial greedy algorithm utilizes this property

Aligned case : Data Collection Module



1. pkt arrives
2. Get an index
3. Set a bit

- size of bitmap n
 - determined by link speed & measurement epoch
 - Assuming 1000 bits average packet size, 4 Mbits for OC-48 (2.4 Gbps) is enough for 1 sec traffic
- hash packets only if enough payload exist (≥ 500 bytes)

Case 2 : Unaligned case



- Much more challenging than aligned case
 - same content can be packetized differently
 - detecting correlations among bitmaps is more complex
 - assuming 576 MTU size & prefix length \sim uniform[0,535], matching probability between bitmaps is only 1/536.
- Our approaches
 - data collection
 - amplify the weakened signal due to the noise (random offsets)
 - design two techniques : offset sampling and flow splitting
 - data analysis
 - design new statistical data analysis techniques

8/17

Unaligned case: Data Collection Module

- **Offset sampling**
 - use K bitmaps : can magnify signal strength to around K^2
 - update bitmaps using K random offsets in $[0,535]$

The diagram illustrates the data collection process. An IP header and its contents are shown. The contents are divided into segments of length 'len'. Each segment is passed through a 'Hash' function. The resulting bitmaps are sampled at various offsets (Offset[1] to Offset[k]) to create a set of bitmaps A[1], A[2], ..., A[k].

10/17

Overview of statistical techniques

- Phase transition theory in ER random graph
 - ER graph $G(n,p)$
 - n : number of vertices
 - p : probability that two vertices are connected by an edge
 - phase transition theory
 - if $p \leq 1/n$, all connected components are of size $O(\log n)$
 - if $p > 1/n$, a giant connected components of $\Theta(n)$ begins to emerge

The diagram shows two graphs illustrating phase transition theory. The left graph shows a sparse network with small components for $p \leq 1/n$. The right graph shows a dense network with a large connected component for $p > 1/n$.

9/17

Unaligned case: Data Collection Module

- **flow splitting**
 - split flows into t different groups of arrays
 - increases correlation between two matching arrays

The diagram illustrates the flow splitting process. An IP header and its contents are shown. The contents are divided into segments of length 'len'. Each segment is passed through a 'Hash' function. The resulting bitmaps are split into t groups (group 1 to group t). The resulting bitmaps are used to identify flows like 'flow 21,6,3...' and 'flow 2,10,9...'.

11/17

Unaligned case: Data Analysis Module

- Converting bitmaps to a graph
 - 1) **vertex mapping**
 - 2) **edge mapping**
 - if $\text{matching_test}(\text{group1}, \text{group2}) = \text{positive} \rightarrow \text{add an edge}$

12/17

Unaligned case: Data Analysis Module

- Case 1
 - no large common content
 - output : random graph $G(n,p)$ (by tuning *thresholds*)
- Case 2
 - contains popular common content
 - output : random graph $G(n,p)$ + preferential attachment

14/17

Computational complexity

- Major bottleneck : bitmap-to-graph conversion
 - memory read-only operation : can be highly parallelized
 - use multiple processors
 - use specially designed hardwares
 - reduce detection accuracy to decrease the complexity
 - can reduce number of monitoring points or groups
 - find pattern in only sampled vertices

13/17

Overview of statistical techniques

- Erdős-Renyi test : pattern presence test
 - procedure
 - 1) pairwise correlation is computed
 - 2) construct a graph with scaling factor $p (< 1/n)$
 - 3) test (the size of largest connected component $\gg O(\log n)$)
 - can provide very accurate binary(yes/no) answer
- Finding the core : identifying nodes
 - greedy algorithm to find most of the nodes that saw the common content is proposed

15/17

Unaligned case: Evaluation

- Simulation: monitor 800 links x 128 groups each = 102,400 vertices
- Erdős-Renyi test
 - common content is packetized into 100 packets
 - $p = 0.65/10^5 (< \text{phase transition probability } 1/n = 1.024/10^5)$
 - varying # of vertices seen the common content

120 vertices seen (15%)

140 vertices seen (17.5%)

Unaligned case: Evaluation

- Finding the core

# packets in common content	# of vertices seen the common content	Average Core Size	Average False Negative	Average False Positive
100	125	65.3(50%)	0.485	0.014
	144	112.1(75%)	0.241	0.025
	165	154.4(90%)	0.099	0.037
110	67	35.6(50%)	0.481	0.023
	77	59.3(75%)	0.239	0.012
	89	81.8(90%)	0.096	0.017
120	44	22.4(50%)	0.491	0.001
	51	38.5(75%)	0.249	0.006
	57	51.9(90%)	0.092	0.002

- Evaluation using tier-1 ISP trace
 - demonstrates the effectiveness of the proposed algorithms

Thank you !

Conclusion

- designed a distributed algorithms to detect common content in Internet traffic
- proposed distributed data streaming approach
- algorithms are shown to be effective through extensive simulations