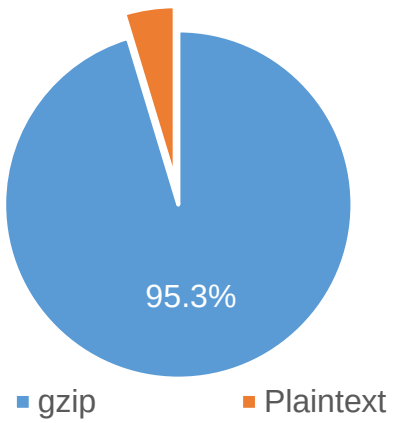
Towards A Fast Packet Inspection over Compressed HTTP Traffic

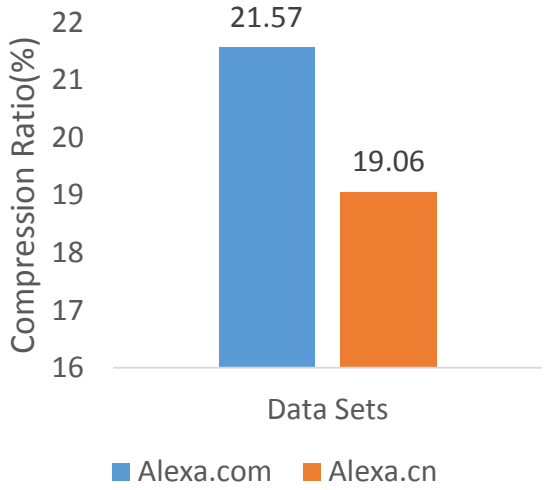
Xiuwen Sun, Kaiyu Hou, Hao Li, Chengchen Hu
Xi'an Jiaotong University

西安交通大學
XI'AN JIAOTONG UNIVERSITY

Two Ratios

- **95%**: most HTTP traffic is compressed with gzip[1] .
- **20%**: the average compression ratio of web pages [1,2].





[1] “Alexa top 500 global sites,” “http://www.alexa.com/topsites/”, accessed Oct. 2016.
[2] “Alexa top china sites,” “http://www.alexa.cn/siterank/”, accessed Feb. 2017.

Two Straw Methods

Lazy

- ✓ Simply ignores the compressed traffic.
- ✓ Easy to bypass the detection by compressing the anomaly traffic.

Naive

- ✓ Matching patterns after a decompression processing first.
- ✓ Slow and hungry for memory (800Mbps).

Idea

Motivation

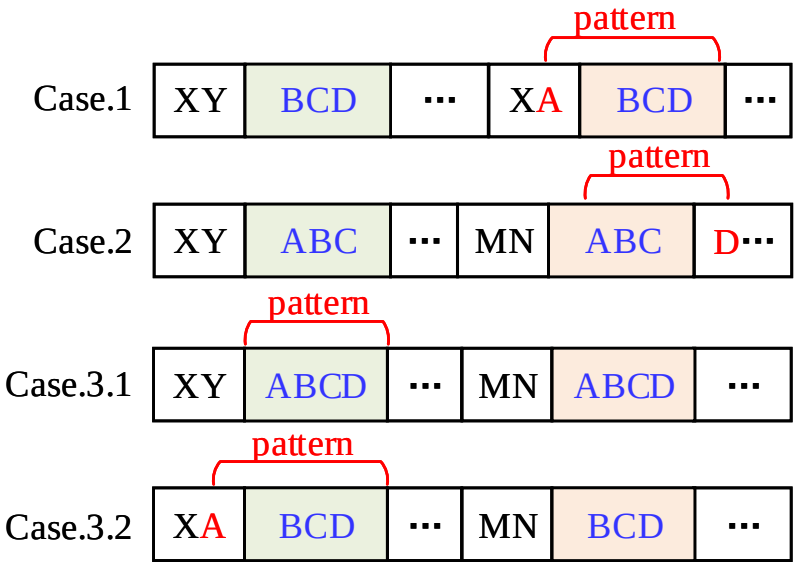
Accelerate multi-pattern matching over compressed traffic.

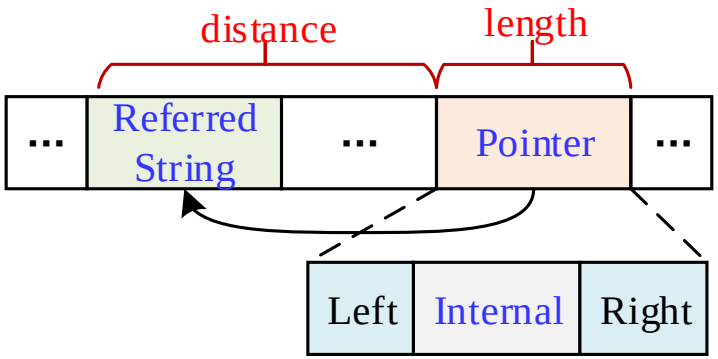
Basic idea

Stop to recheck the pattern within compressed data if it has been matched before.

Previous Work: ACCH

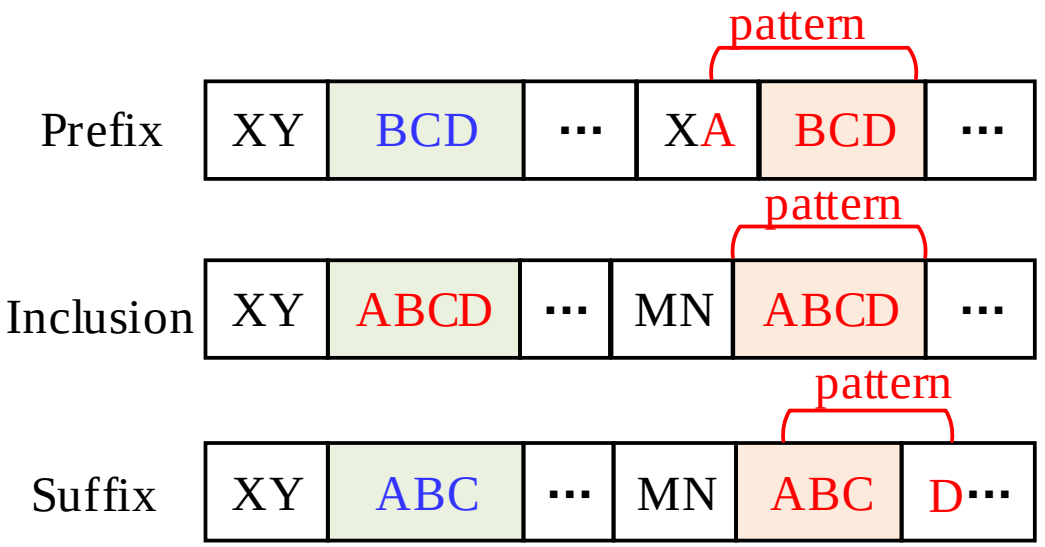
- ✓ Accelerates matching process by reusing the saved information.
- ✓ If no matched patterns occurred within the referred string, it skips the Internal area in pointer. However, if not, ACCH has to scan these bytes again.





Categories of COIN

- ***Prefix***: pattern starts before Pointer, regardless of its end position.
- ***Inclusion***: pattern is contained in Pointer entirely.
- ***Suffix***: pattern starts in Pointer, but not contained.



Comparison : Inclusion

COIN: (Skipped 6 bytes)

1. Find position with *Match* state in Pointer, mPos=5, mLen=4.
2. Pattern “ABCD” in this pointer completely. Store this record as matching infomation.

ACCH: (Skipped 2 bytes)

1. Find position with *Match* state in Referred String, mPos=5.
2. Find last position with u state before mPos, unchkPos=2.
3. Begin scanning from unchkPos - CDepth + 2, i.e. byte of “A”.

Ex.1	XY	ABABCD	MN	ABABCD
depth	0 0	1 2 1 2 3 4	0 0	1 2 1 2 3 4
COIN	0 0	0 0 0 0 0 m	0 0	0 0 0 0 0 m
ACCH	u u	u c u c c m	u u	? ? u c c m

Comparison : Suffix

COIN: (Skipped 2 bytes)

1. The **depth** of last byte ds=4.
2. Begin scanning from lastPos-ds, i.e. byte of “A”.

ACCH: (Skipped 2 bytes)

1. Find last position with u state before last byte, unchkPos=2.
2. Begin scanning from unchkPos - CDepth + 2, i.e. byte of “A”.

Ex.2	XY	ABABC	MN	ABABC	D
depth	0 0	1 2 1 2 3	0 0	1 2 1 2 3	4
COIN	0 0	0 0 0 0 0	0 0	0 0 0 0 0	m
ACCH	u u	u c u c c	u u	? ? u c c	m

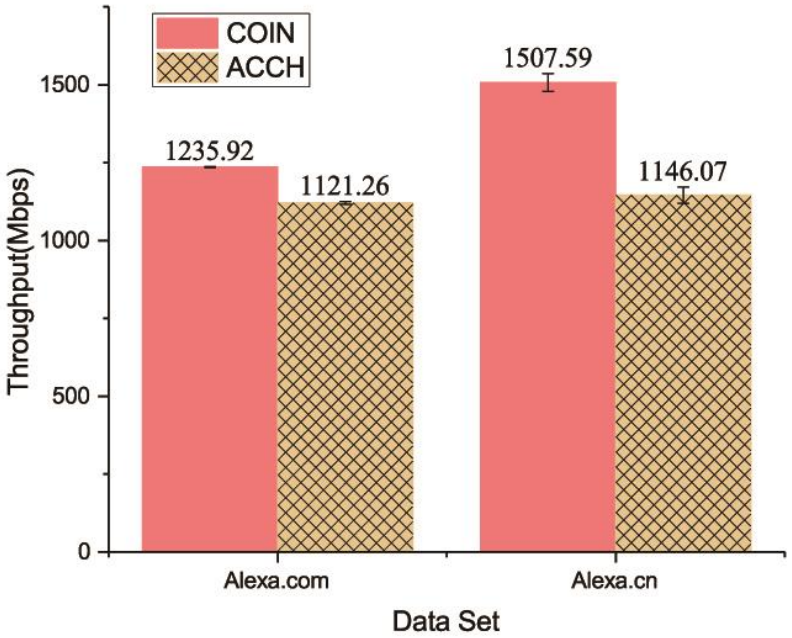
Evaluation Settings

- **Platform**: Desktop PC (Intel 4-core 3.4GHz and 8G RAM).
- **Rule Set**: String patterns from Snort rules (1430).
- **Data Sets**: Homepages of Alexa.com top 500 and Alexa.cn top 20000 sites (as below).

Characteristic	Alexa.com	Alexa.cn
Count of Compressed Pages	428	13747
Compressed Size (MB)	14.73	226.95
Decompressed Size (MB)	68.28	1190.99
Bytes represented by pointers	91.35%	91.92%
Average pointer length (B)	15.30	19.84

Performance

- **Figure**: COIN is more efficient than ACCH in both data sets.
- **Table**: Performance boost mainly due to COIN can save much time on handling the inclusion case.



Data Sets	Average pointer length (B)	Bytes represented by pointer (%)	Pattern in inclusion ratio (%)	Boost
Alexa.com	**15.30**	91.35%	92.61%	**10.23%**
Alexa.cn	**19.84**	91.92%	94.23%	**31.54%**
Subset	**19.19**	91.53%	93.64%	**28.64%**

Conclusion

COIN Present a method for multi-pattern matching on compressed HTTP traffic.

Faster 10-31% improvement in speed under the experiments with real traffic.

Simple Don’t need any pre-defined parameter settings.

Thank You

Questions ?

Discussion

Completeness on Categories of COIN

- ✓ Based on positional relationships between pointer and pattern.
- ✓ It can be assembled to all other more complex cases.

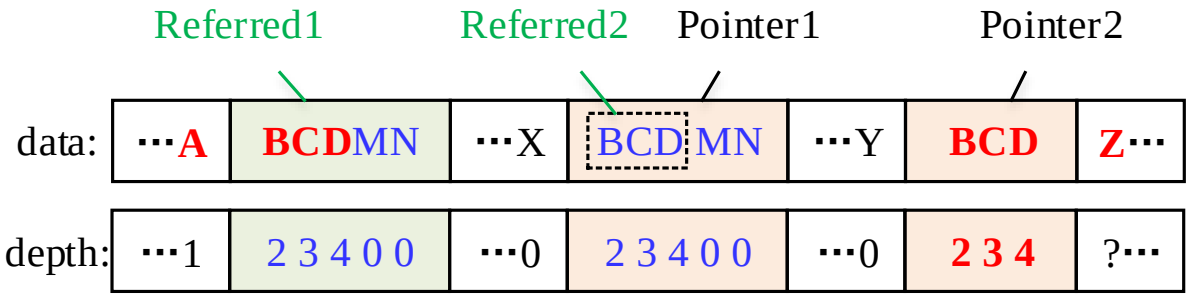
Memory Usage

- ✓ COIN and ACCH store the matching information.
- ✓ COIN needs more space to preserve the parameter of *depth*.
- ✓ ACCH has more information on *status* to be stored.

Discussion

Correctness

- ✓ ***Prefix*** and ***Inclusion***, COIN doesn't need **depth** in Pointer.
- ✓ ***Suffix***, the **depth** at last byte is no less than its actual value.
  - "BCD" is a prefix of a pattern, **equal**.
  - "BCD" is not a prefix of a pattern, **bigger**.



西安交通大學 XI'AN JIAOTONG UNIVERSITY