Measuring and Analyzing DoS Flooding Attacks

Amir Farhat
Sam DeLaughter
Karen Sollins

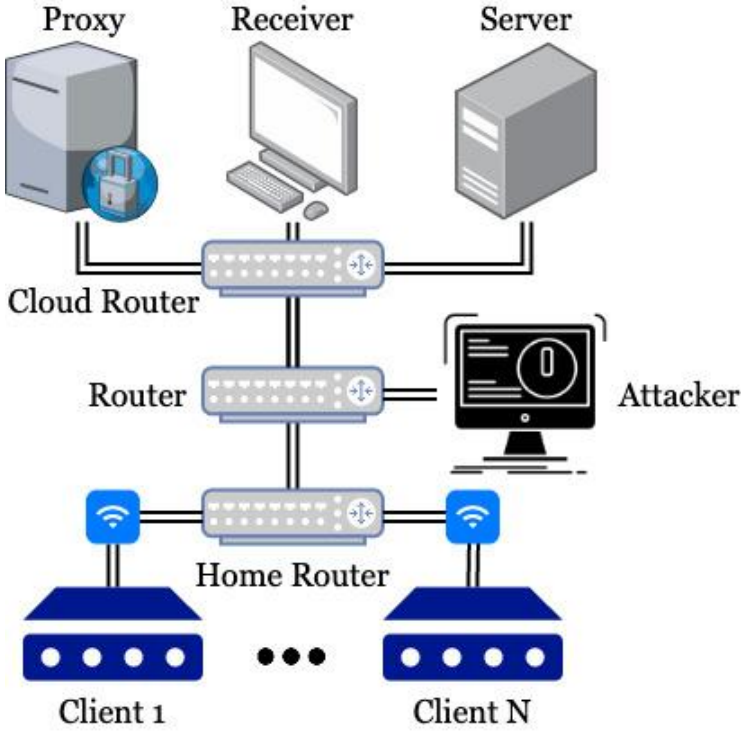
Setup

- Flooding DoS experiments
- 144 experiments (5 trials each), 354GB of data
- ⇒ Difficult to collect data, unwieldy to manage
- ⇒ We developed a **toolkit** for this

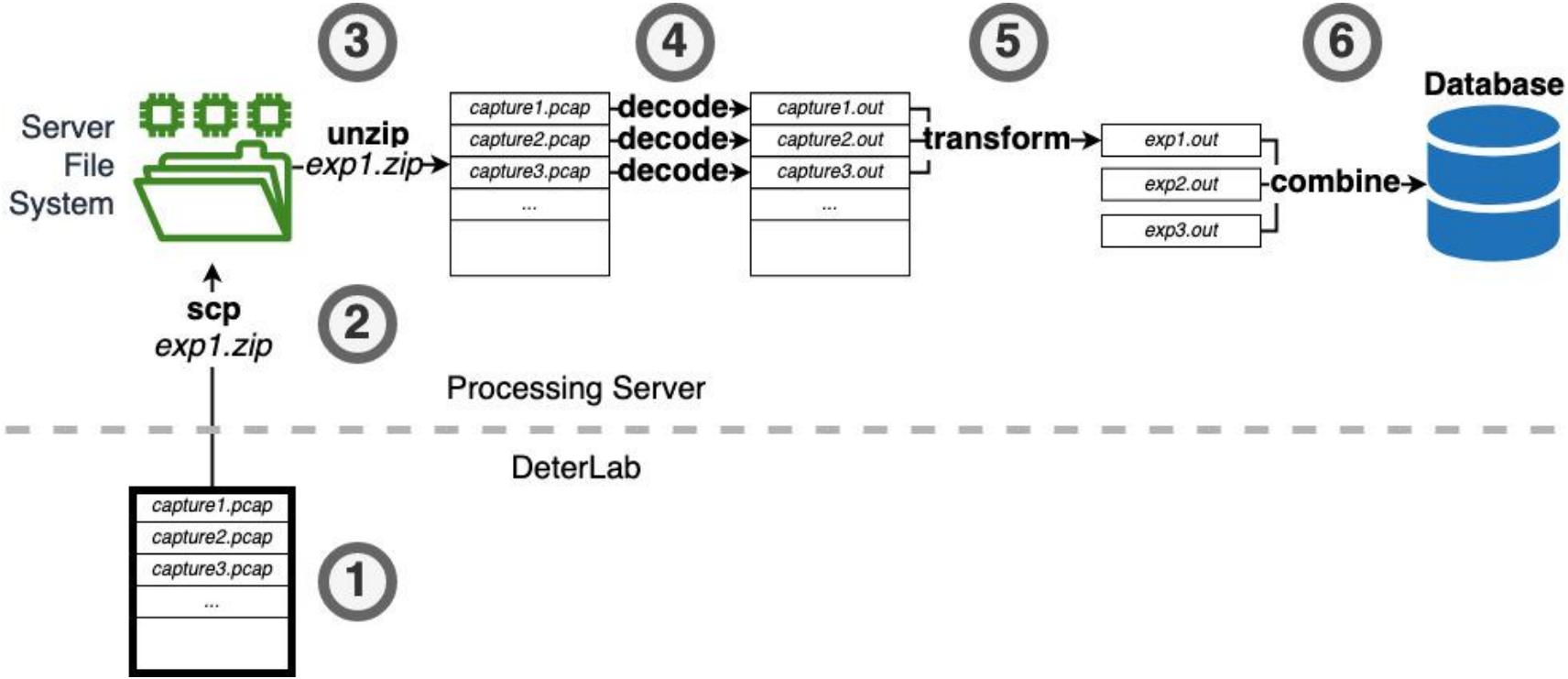
Design Goals

➔ Low-overhead instrumentation via packet captures
  - Get 3% CPU and 0.5% memory utilization in the worst case

➔ High performance and automated where possible
  - Decoding and transforming packet captures
  - Grouping experiments for analysis



Toolkit Design

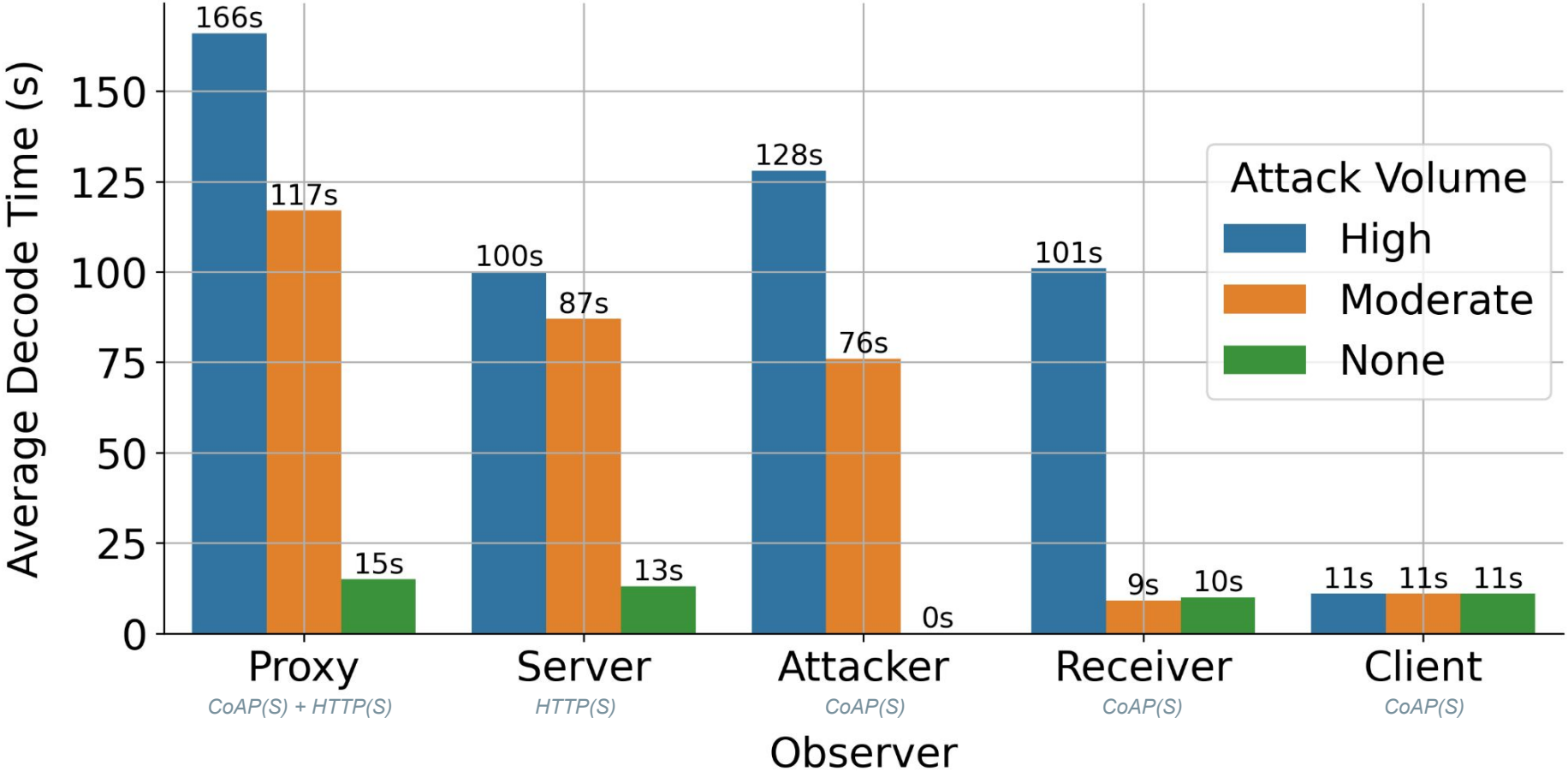


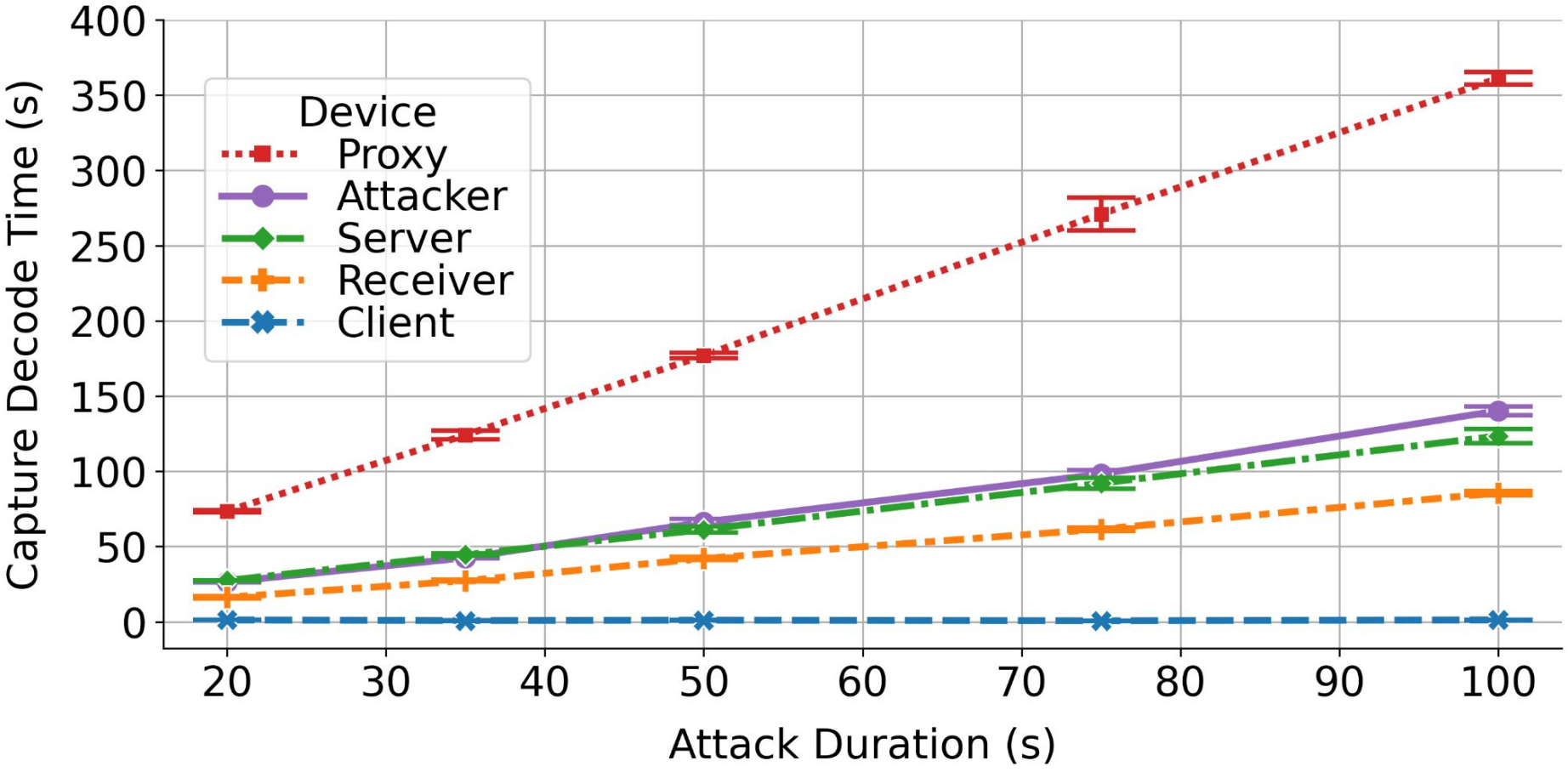
Next

Capture Decoding Times

Experiment Group Storage Sizes

Read Query Times


Average Decode Time (s)
0
25
50
75
100
125
150
166s
117s
15s
100s
87s
13s
128s
76s
0s
101s
9s
10s
11s
11s
11s
Attack Volume
High
Moderate
None
Proxy
CoAP(S) + HTTP(S)
Server
HTTP(S)
Attacker
CoAP(S)
Receiver
CoAP(S)
Client
CoAP(S)
Observer


Capture Decode Time (s)
400
350
300
250
200
150
100
50
0
Device
Proxy
Attacker
Server
Receiver
Client
20
30
40
50
60
70
80
90
100
Attack Duration (s)

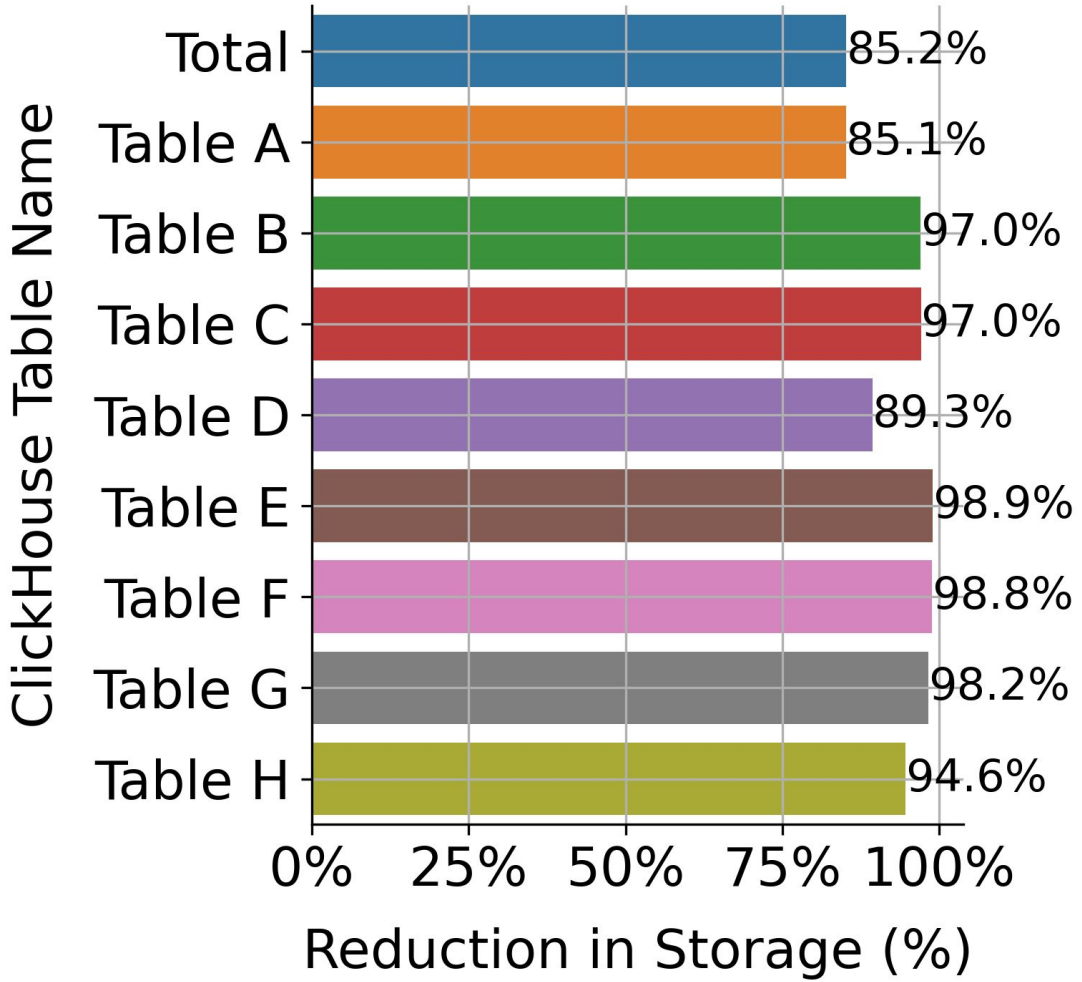
Next

~~Capture Decoding Times~~

Experiment Group Storage Sizes

Read Query Times

	Storage Space (% of Total Storage)	
Table Name	**PostgreSQL**	**ClickHouse**
Table A	**1,394 MB (99.7%)**	**207 MB (99.9%)**
Table B	4 MB (< 0.5%)	135 KB (< 0.01%)
Table C	32 KB (< 0.01%)	945 B (< 0.01%)
Table D	32 KB (< 0.01%)	3 KB (< 0.01%)
Table E	32 KB (< 0.01%)	357 B (< 0.01%)
Table F	32 KB (< 0.01%)	382 B (< 0.01%)
Table G	32 KB (< 0.01%)	569 B (< 0.01%)
Table H	24 KB (< 0.01%)	1 KB (< 0.01%)

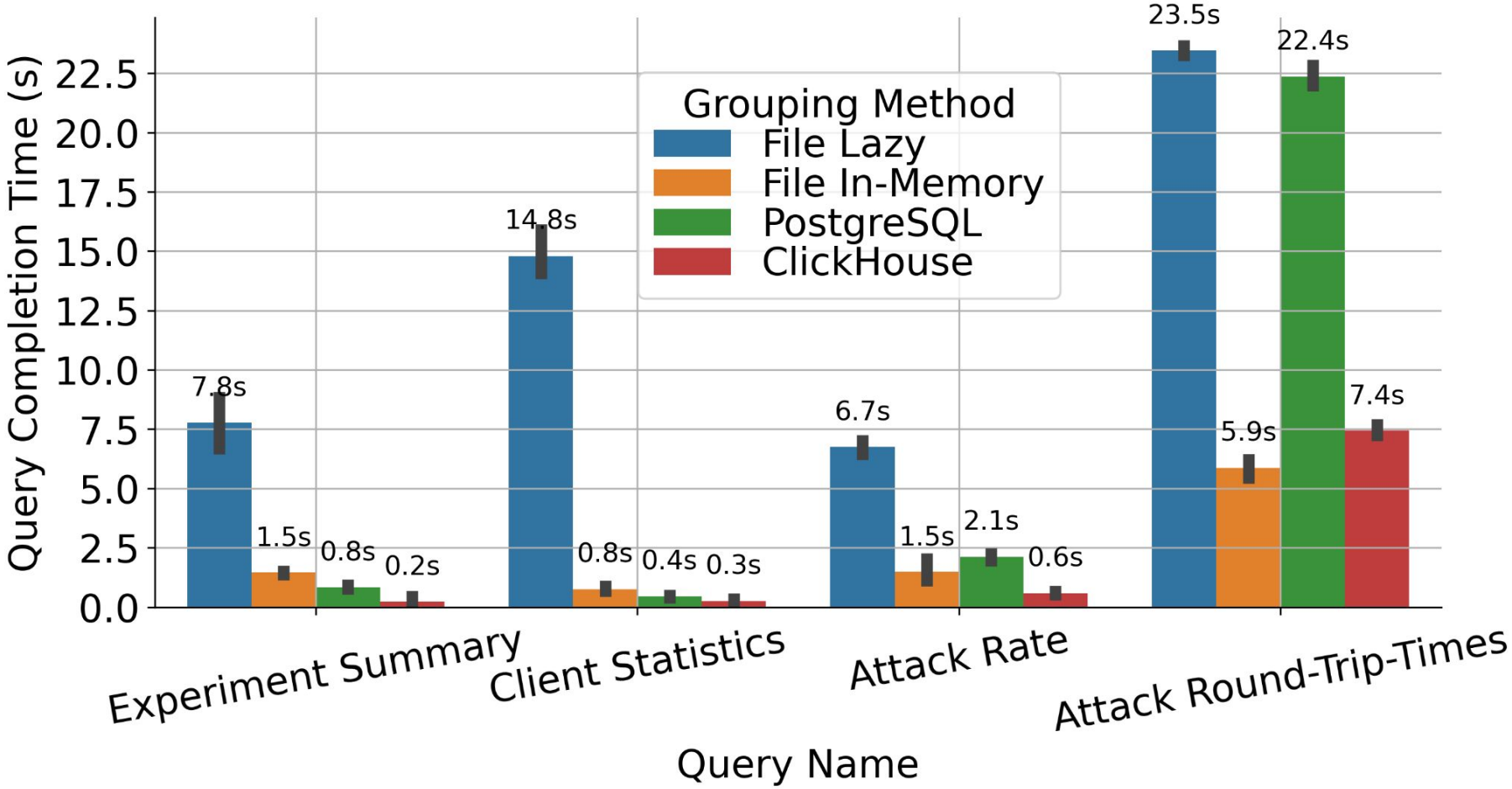


Next

~~Capture Decoding Times~~

~~Experiment Group Storage Sizes~~

Read Query Times


Query Completion Time (s)
0.0
2.5
5.0
7.5
10.0
12.5
15.0
17.5
20.0
22.5
Grouping Method
File Lazy
File In-Memory
PostgreSQL
ClickHouse
7.8s
1.5s
0.8s
0.2s
14.8s
0.8s
0.4s
0.3s
6.7s
1.5s
2.1s
0.6s
23.5s
5.9s
22.4s
7.4s
Experiment Summary
Client Statistics
Attack Rate
Attack Round-Trip-Times
Query Name

Conclusion

- Packet capture works
  - Low overhead: 3% CPU and 0.5% memory utilization
  - Minimally affects experiment results
- But it introduces complexity in processing
  - Introduces decoding and transformation stages
  - Decoding stage is a bottleneck, 6+ mins for 1.5 min attack
  - We need faster but still flexible packet capture decoders
- ClickHouse is a fantastic choice for grouping experiments
  - Fast writes and reads
  - Saves 85%+ in storage
- Code and analysis are publically available