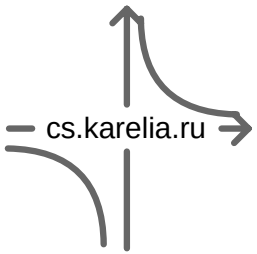
Joint analysis of Squid and Netflow log files using http client port information

Alexander S. Volkov, Yriy A. Bogoyavlensky



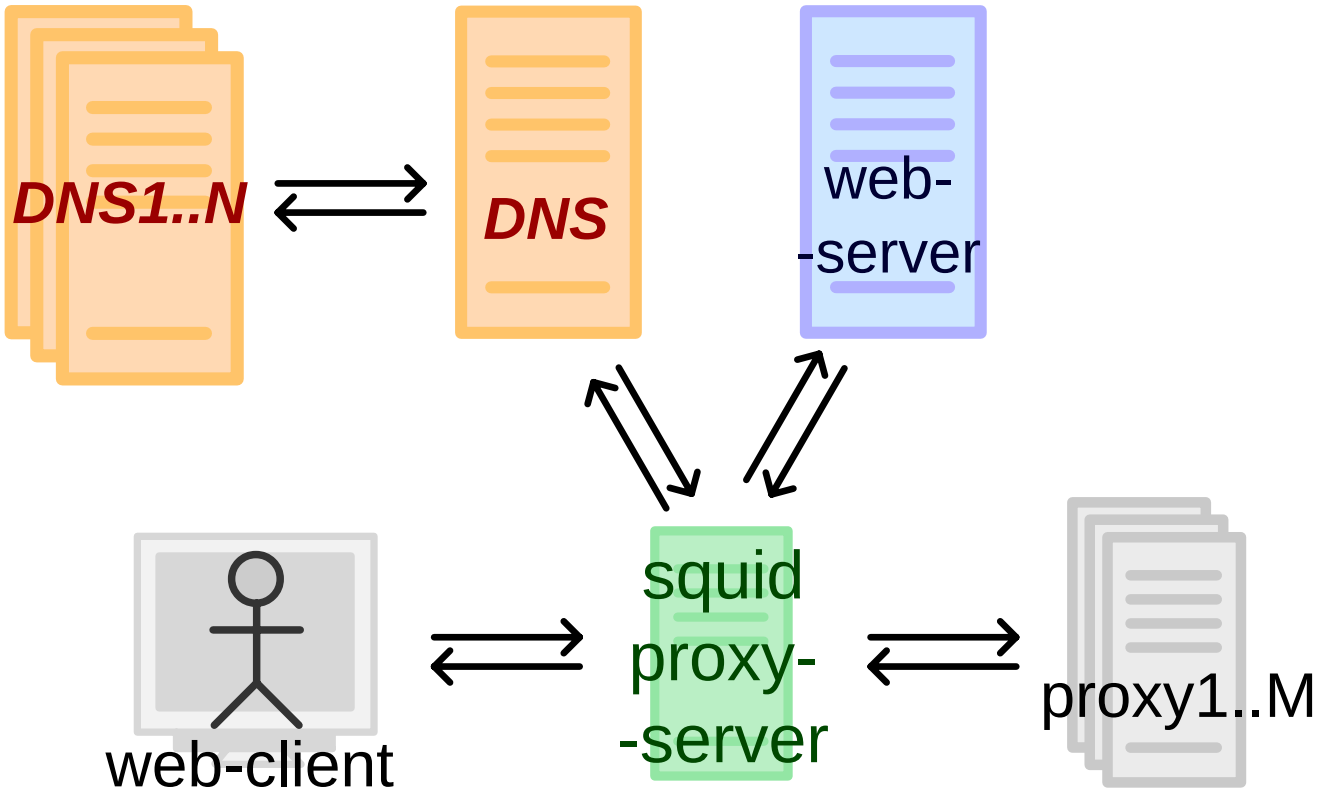
Petrozavodsk state university

AMICT'07
Petrozavodsk, 2007

Introduction

- **Joint analysis of network traffic data.**
- **New features of the joint analysis of traffic:**
  - **traffic data aggregation;**
  - **higher accuracy of web traffic;**
  - **more common traffic units.**

A common web-request's scheme



Multitudes of Netflows between nodes

- **Request hit (object is found in local proxy-server's cache):**
  2 Netflows
  (1 TCP connection)

- **Sibling cache hit (object is found in neighbour's cache):**
  2*m+4 Netflows
  (2 TCP connections и 2*m UDP flows)

- **Request miss (there are no object in any cache):**
  2*m+2*n+6 Netflows
  (2 TCP и 2*m+2*n+2 UDP flows/TCP connections)

n – number of dns-requests to official dns-servers passing through router
m – number of icp-requests to sibling proxy-servers passing through router

Algorithm requirements

Algorithm requirements:

- **presence of clients' ports in access.log;**
- **most of data flows should pass through the router;**
- **timed traffic data;**
- **time sorted access.log file.**

The algorithm

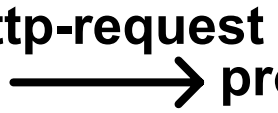
```
for each access.log items {
    while( next data flow selection() ) {
        search for corresponding Netflows();
    }
}
```

Next data flow selection

- **Next data flow selection rules**
  (depend on proxy settings and requests' results that was obtained from preceding data flows).

- **Check for Netflow(s) presence**
  (network configuration and servers' settings).

Next data flow selection rules. Example

if current data flow is «web-client 

proxy-server» then {

if requested object was founded in local proxy--server's cache (*hierarchy code-1* field of access.log item equals NONE and *result codes* equals one of next values: TCP_HIT, TCP_NEGATIVE_HIT, TCP_MEM_HIT, TCP_DENIED, TCP_OFFLINE_HIT), then the next (and the last in sequence) data flow is

«proxy server 

web client»;

else if ...

...

}

Netflow presence check

- **Dynamic Netflow presence check for «web-client – proxy-server» and «web-server – proxy-server» data flows.**

- **Hand Netflow presence check for other data flows (due to static source and destination IP addresses for all access.log items).**

Search for Netflows. Example

Example:

Data flow «web-client 

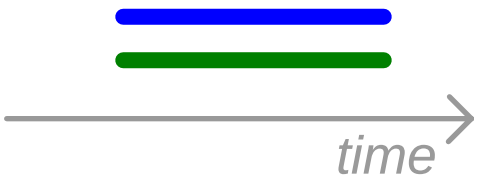
proxy-server»:

source IP address = client address (access.log);
destination IP address = proxy address(squid.conf);
source TCP application port client = client port (access.log);
destination TCP application port = http_port (squid.conf);
IP Protocol = 6 (TCP);

Problem: random ports for «proxy-server – web-server», «proxy-server – local dns-server», «local dns – official dns» data flows.
Solution: first right Netflow selection (in timeline)

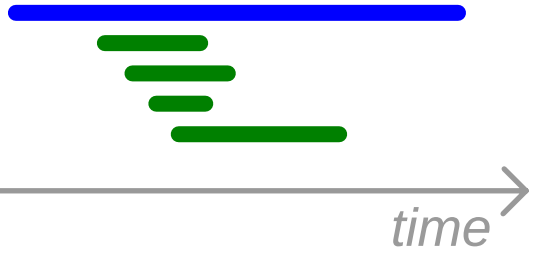
Time search for Netflows

- single Netflow corresponds to single access.log items



TCP end flag

- single Netflow corresponds to several access.log items



HTTP/1.1

- several Netflows correspond to single access.log item

Netflow cache rules

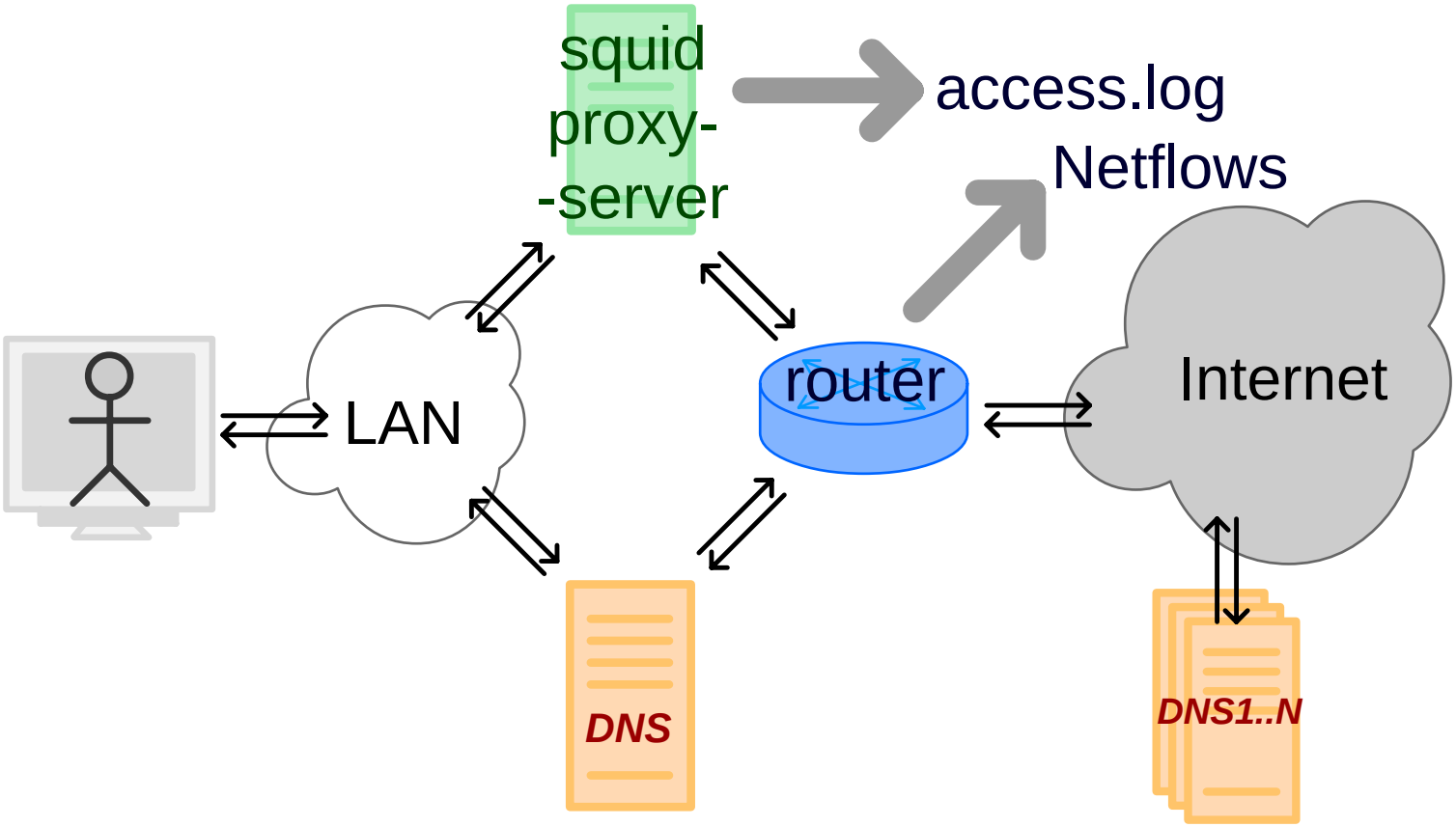
- several Netflows correspond to single access.log item

Netflow cache rules

Results and evolution

- **Full common web-request's scheme analysis is implemented**
- **Testing experimental joint analysis algorithm on real traffic data**
- **Future elaboration of web traffic aggregation methods**

Experimental environment



Thanks for listening!