The CINBAD Project Update

24th June 2008

Milosz Marian Hulboj - CERN/Procurve

Ryszard Erazm Jurga - CERN/Procurve

Agenda

- Anomaly definition and detection – the survey
- sFlow data source
- sFlow datagram structure
- Estimates of sFlow data from CERN network
- Scalable collector design

- Large scale sFlow collection – initial testing
- Data aggregation
- Visit at HP Procurve in Roseville

Anomaly Definition (1)

- Anomalies are a fact in computer networks
- Anomaly definition is very domain specific:

Network faults	Malicious attacks	Viruses/worms
Misconfiguration	…	…

- Common denominator:
  - *“Anomaly is a deviation of the system from the normal (expected) behaviour (baseline)”*
  - *“Normal behaviour (baseline) is not stationary and is not always easy to define”*
  - *“Anomalies are not necessarily easy to detect”*

Anomaly Definition (2)

- Just a few examples of anomalies:
  - Unauthorised DHCP server (either malicious or accidental)
  - NAT (not allowed at CERN)
  - Port Scan
  - DDoS attack
  - Spreading worms/viruses
  - Exploits (attacker trying to exploit vulnerabilities)
  - Broadcast storms
  - Topology loops
- Examples of potential anomaly indicators:
  - TCP SYN packets without corresponding ACK
  - IP fan-out and fan-in (what about servers – i.e. DNS?)
  - Unusual packet sizes
  - Very asymmetric traffic to/from end system (what about servers?)
  - Unwanted protocols on a given subnet (packets ‘*that should not be there*’)
  - Excessive value of a certain measure (i.e. TCP Resets)
  - ICMP packets

Anomaly Detection (1)

- Signature based detection methods:
  - Perform well against known problems

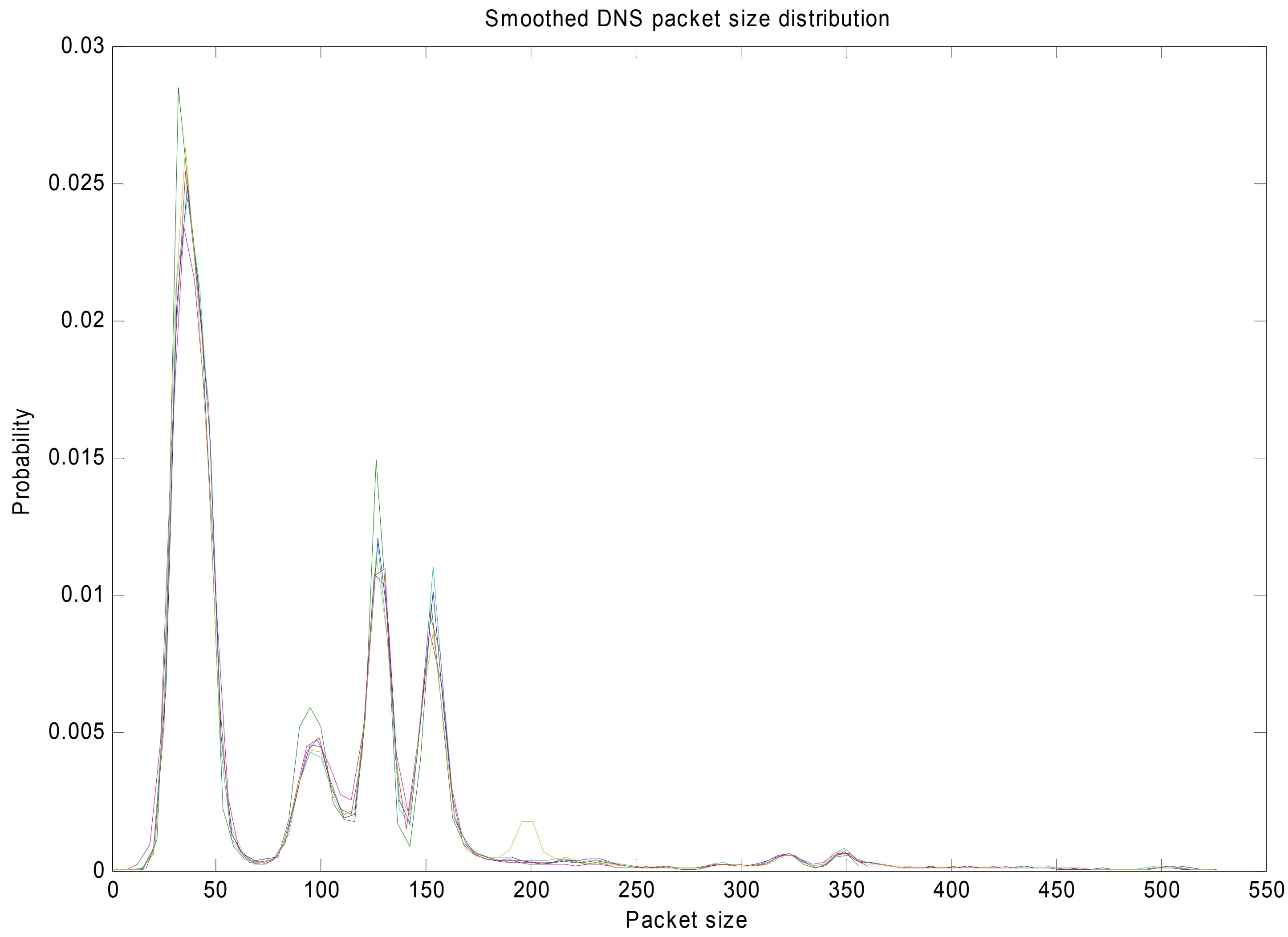
Example:
Martin Overton, “Anti-Malware Tools: Intrusion Detection Systems”, European Institute for Computer Anti-Virus Research (EICAR), 2005

```
00000760  E7 6F 8C 88 3A 79 B3 9D 9D 52 44 AD 62 61 3D 8F   ço▌▌:y³▌▌RD-ba=▌
00000770  98 6D 4C 07 C2 00 E5 4C 48 F0 91 4E EB 87 89 77   ▌mL▌Â.åLHð´Në▌▌w
00000780  7E E0 83 B1 94 94 CC E9 F5 97 97 53 95 5C 95 AF   ~à▌±▌▌Ìéõ▌▌S▌\▌¯
00000790  C6 40 C5 CA AC 25 8E 47 F1 5D 0B 9F BB CB A6 67   Æ@ÅÊ¬%▌Gñ]▌▌»Ë¦g
000007A0  DB 44 E8 D2 48 3B 8F 76 CB 9E E1 53 FB FB 41 11   ÛDèÒH;▌vË▌áSûûA▌
```

Signature found at W32.Netsky.p binary sample
Rules for Snort:

```
alert tcp $EXTERNAL_NET any -> any any (msg:"W32.NetSky.p@mm - SMB";content:"|4E EB 87 89
77 7E E0 83 B1 94 94 CC E9 F5 97 97 53 95 5C 95 AF C6 40 C5 CA AC 25 8E 47 F1 5D 0B|";
classtype:misc-activity;rev:1;)
```

Anomaly Detection (2)



Anomaly Detection (3)

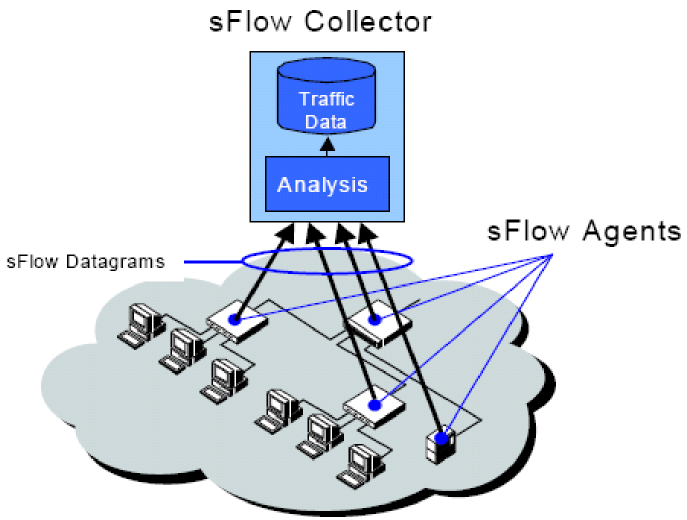
- Statistical detection methods – examples:
  - Threshold detection:
    - Count occurrences of the specific event over ΔT
    - If the value exceeds certain threshold -> fire an alarm
    - Simple and primitive method
  - Profile based:
    - Characterise the past behaviour of hosts (i.e. extract features, patterns, sequential patterns, association rules, classify into groups)
    - Detect a change in behaviour
    - Detect suspicious class of behaviour

Anomaly Detection (4)

- Important questions:
  - Which metrics provide good input for anomaly detection?
  - Do the same types of anomalies affect the metrics in similar way? (Is there a pattern?)
  - Are we able to observe sufficient amount of network data (are the anomalies **observable**)?
  - Are we able to do post-mortem analysis?
    - Can we understand what had happened with the collected metrics?
    - It is not an online analysis – it is not possible to get any more data!

sFlow Packet Sampling – Overview

- A mean of passive network monitoring
- RFC 3176
- Multi-vendor standard
- Complete packet header and switching/routing information
- Some SNMP counters information
- Low CPU/memory requirements – scalable

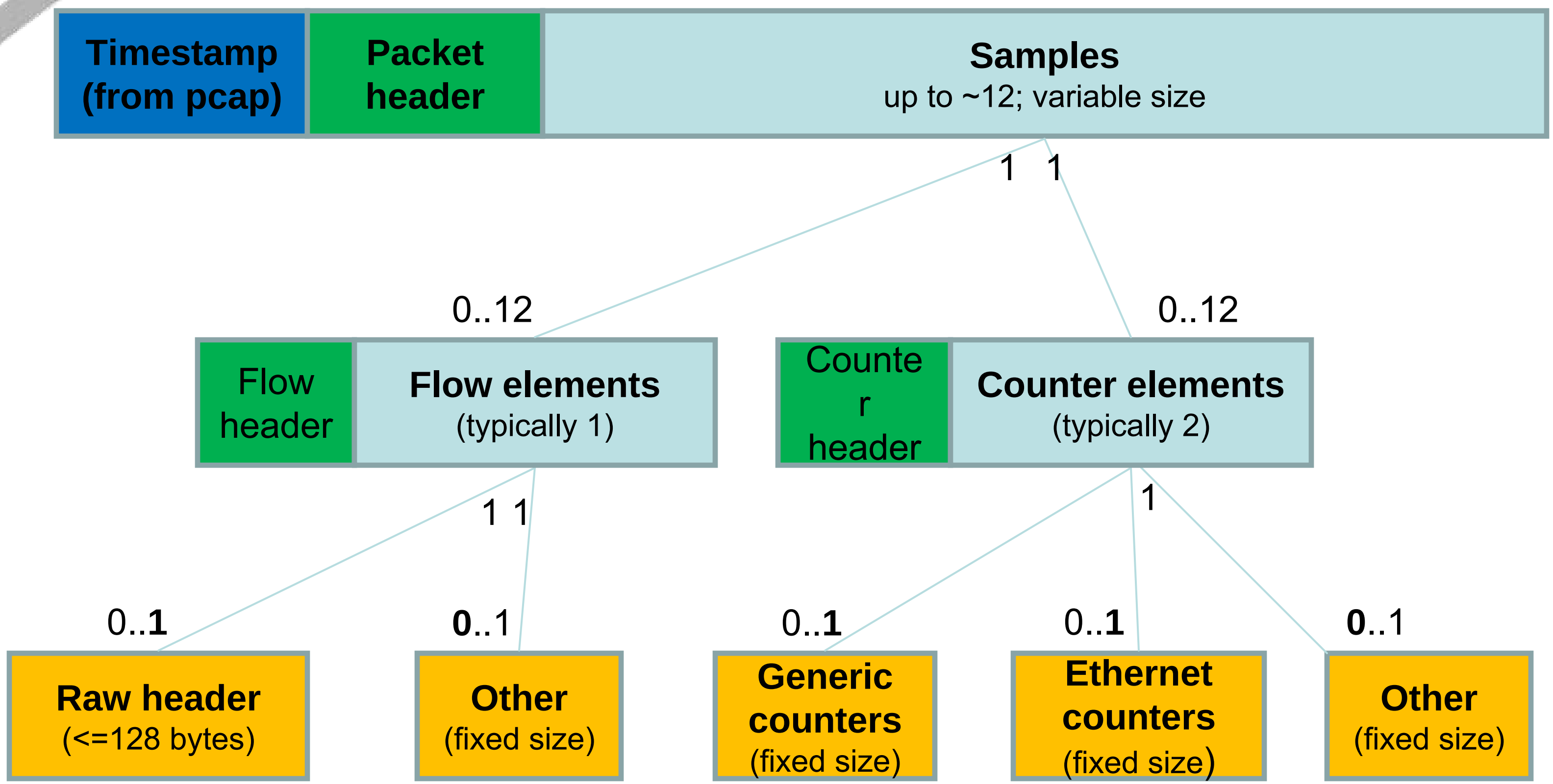


sFlow Packet Sampling – Usage

- Profiling network traffic
- Building flow statistics
- Accounting and billing
- Route profiling (forwarding information)

- Security analysis / intrusion detection:
  - Packet headers analysis
  - Traffic pattern analysis

sFlow Datagram Structure

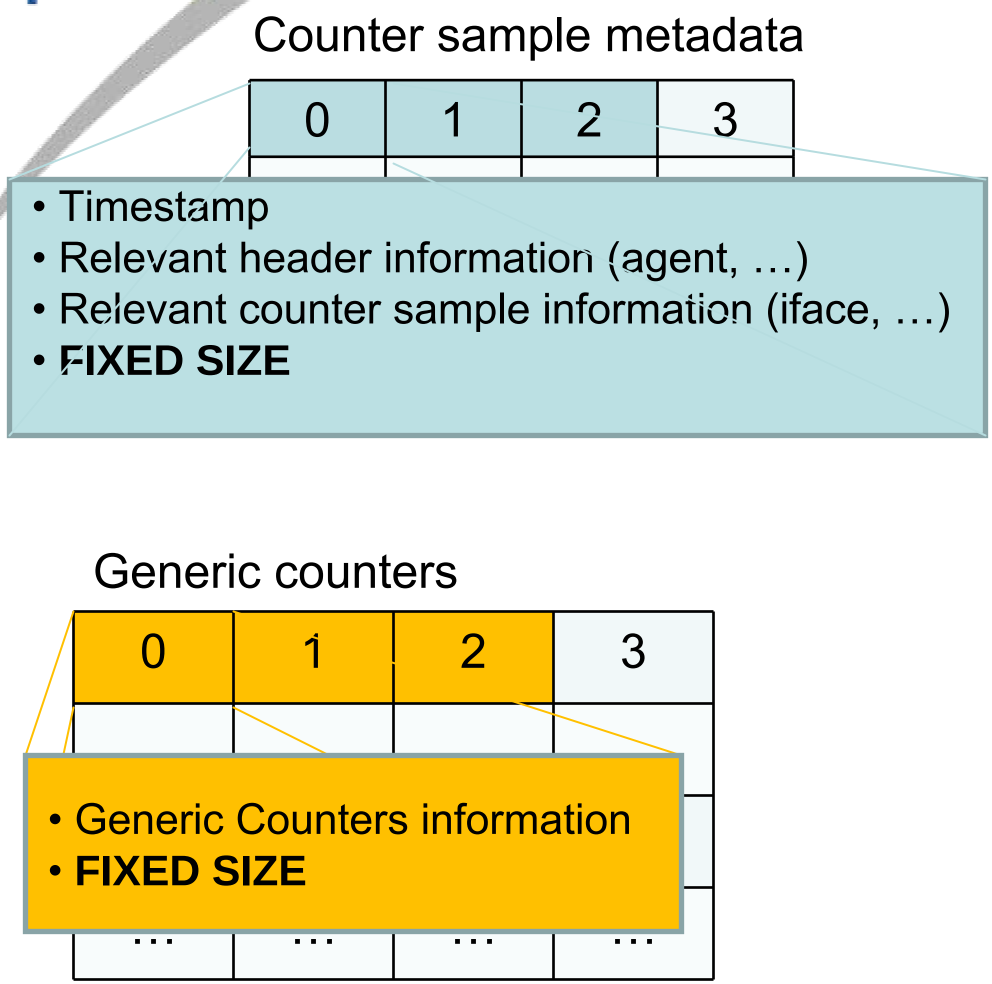


- Variable format of datagram makes direct access to sample elements impossible → parsing needed

sFlow Datagram Structure Issues

- sFlow datagram tree-like format is not ideal
- Our main wishes:
  - Fast direct access to all sample elements
  - Having all the needed data in one place
  - Avoiding multiple parsing of the sFlow tree
- At least two possible solutions:
  - **Flattening of the tree**
  - Introducing some indirection level (pointer-like)

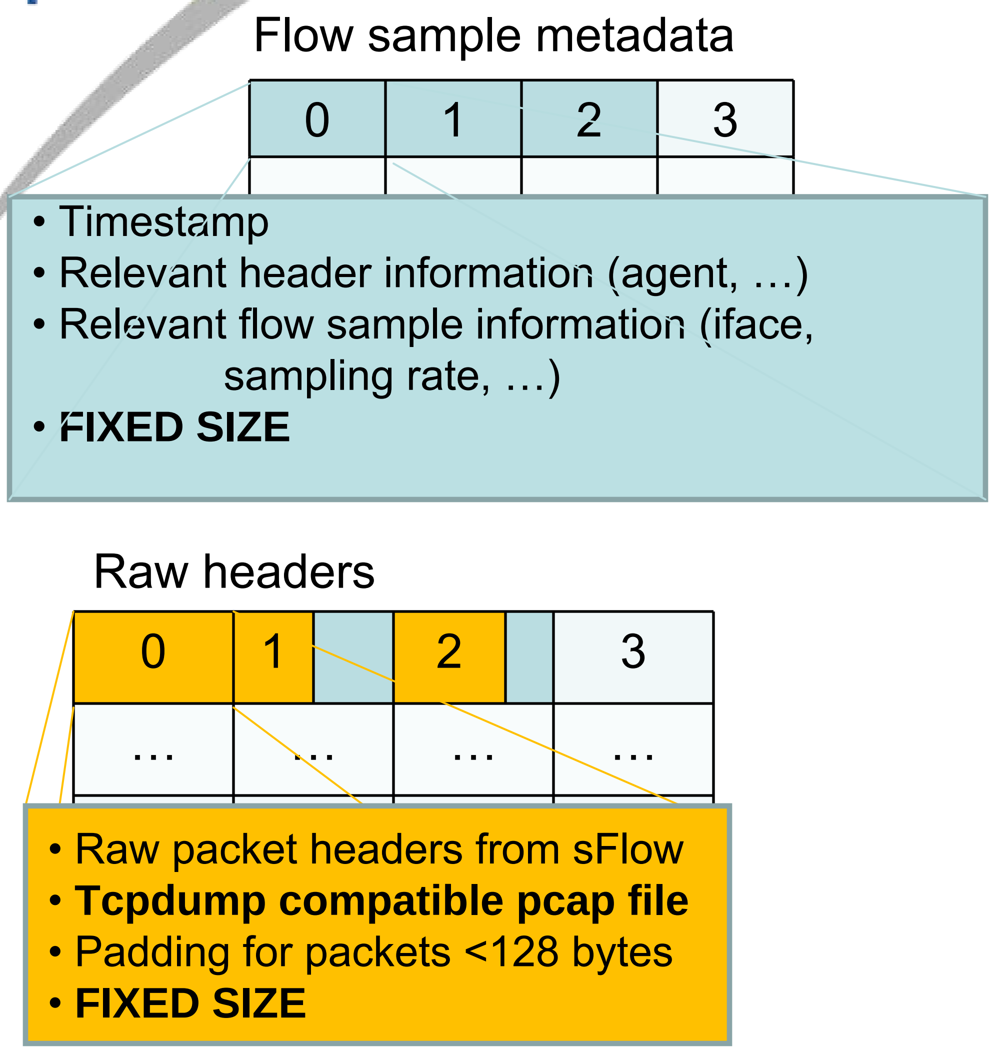
sFlow Flattened Approach (1)



Flattened Approach:

- Each metadata entry describes one counter data entry
- Could be stored in one file if only one type of counters is to be stored
- Random and direct access to all the data
- Space overhead
  - more repetition of metadata than in tree structure

sFlow Flattened Approach (2)



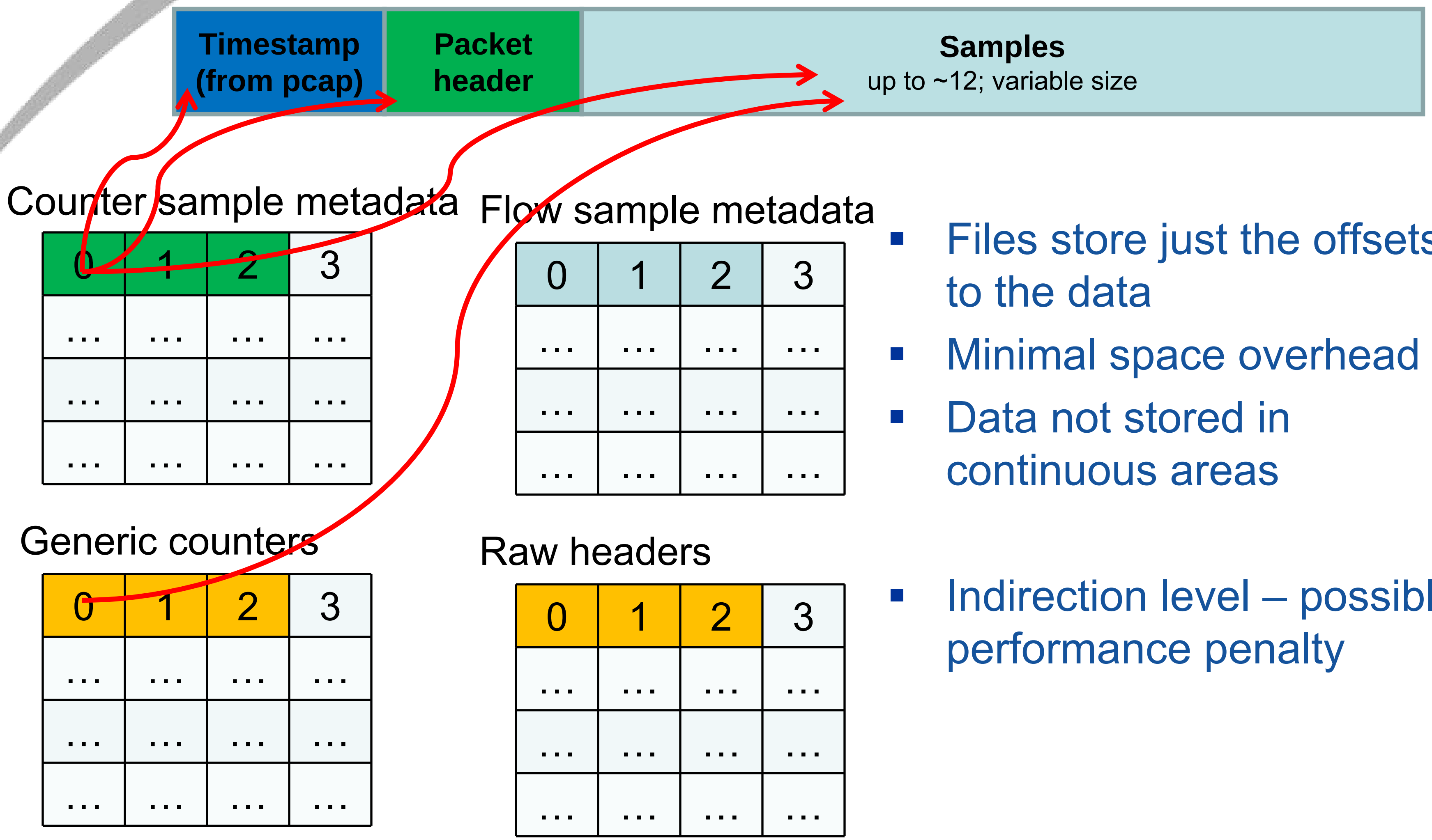
Flattened Approach:

- Each metadata entry describes one flow data entry
- Stored in two different files – pcap compatibility
- Random and direct access to all the data
- Space overhead:
  - more repetition of metadata than in tree structure
  - Internal fragmentation (due to padding)

Flattened Approach Summary

- Solution provides direct access to all the data
- All the data is available in one (two) place(s)
- Raw headers stored in pcap compatible format:
  - Wide range of tools support pcap files (i.e. tcpdump, SNORT)
- Data stored in continuous area
- Space overhead (redundant metadata + padding)
- For now we think it is a good and flexible solution
- We will have to carefully select metadata to store in the flattened form (minimise space overhead)

Indirect Approach



- Files store just the offsets to the data
- Minimal space overhead
- Data not stored in continuous areas

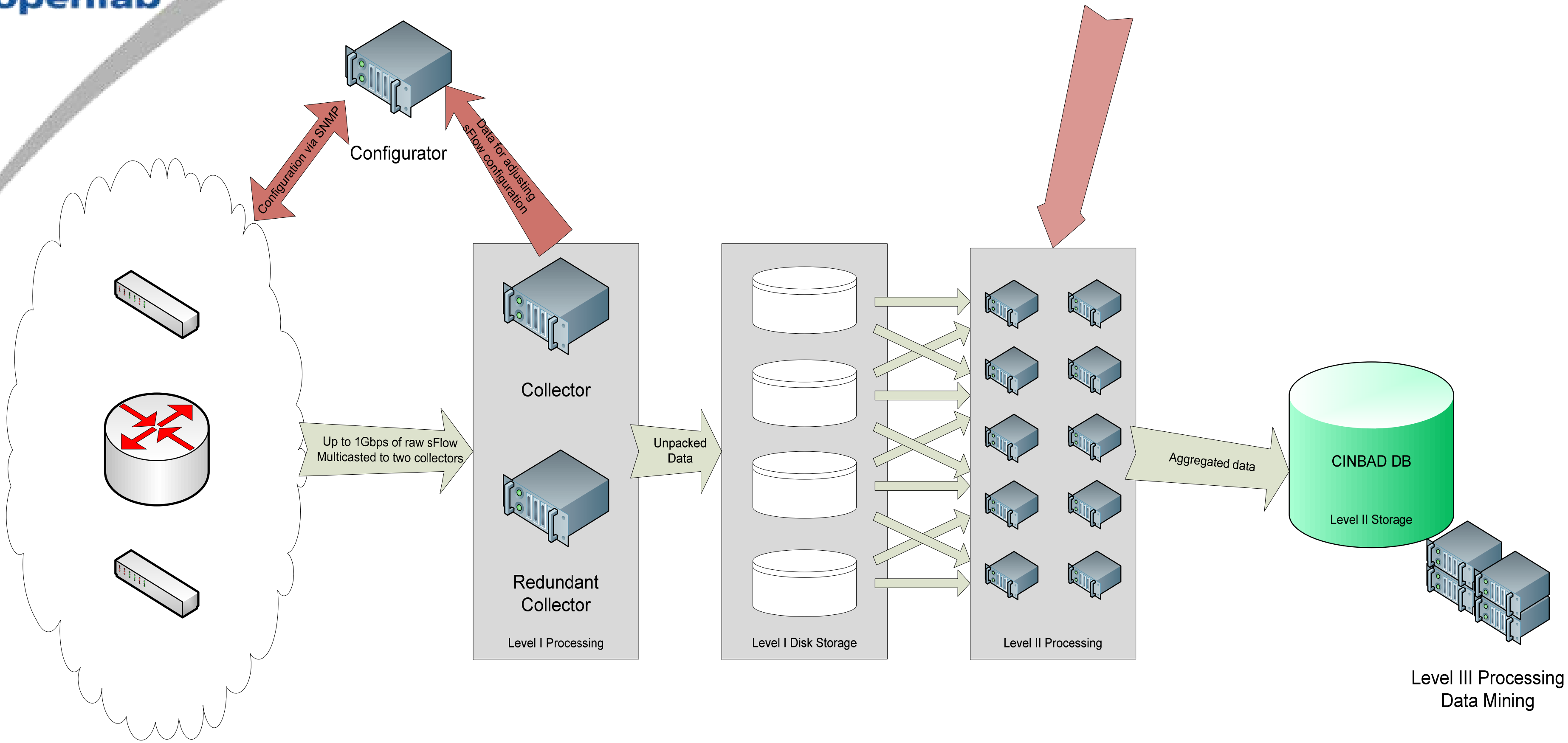
- Indirection level – possible performance penalty

sFlow Data Collector Design (1)

- Estimated data collected
  - ~3TB of raw sFlow datagrams from 2000 network devices per day

- Survey on data acquisition @ CERN:
  - Current Oracle and application performance in use at CERN: Lemon, PVSS, etc
  - LHC experiments experts consulted:
    - High performance Data storage
    - Data format and representation
    - Analysis principles
- Conclusion: follow a two level strategy

sFlow Data Collector Design (2)



Highly Scalable Architecture

Rich database for investigative data mining

Why Data Aggregation? (1)

- Randomness of sFlow data
  - one random packet header is not representative
    - information carried by individual packets is not statistically interesting, except pattern matching

> Do you know what are three kinds of lies?
> Lies, damned lies, and statistics. *Benjamin Disraeli*

  - more packet headers are needed to draw conclusion about the network traffic
    - requires some time interval to collect packets
    - multiple occurrence of similar packets is interesting
    - many packets can contribute partial information into global picture of the network traffic

Why Data Aggregation? (2)

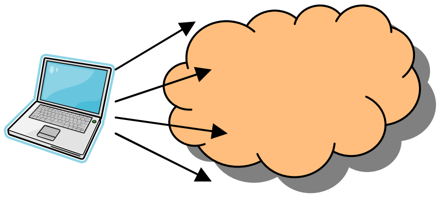
- Analysis of sFlow data
  - statistical analysis
  - classification into groups based on timestamp and packet attributes (i.e. type of protocol, source and destination addresses)
  - usage of numerical descriptors like mean, standard deviation to summarize the classified data over some time interval
  - inference about the network traffic, i.e.
    - setting up baseline,
    - modeling patterns,
    - identifying trends
  - showing the difference between the healthy and anomalous network traffic
    - correlation with other data sources, i.e. antivirus, intrusion detection systems

Key Packet Attributes

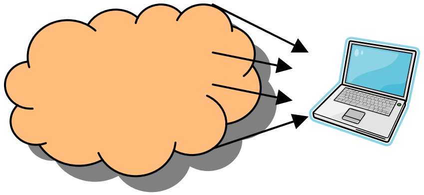
- Device and interface where the packet was sampled
- Packet size
- Source and destination MAC/IP addresses
- Source and destination TCP/UDP ports
- Protocol type (i.e. IP, ARP, ICMP, OSPF, TCP, UDP)
- Protocol specific information (i.e. TCP flags, ICMP codes)

Examples of Aggregates

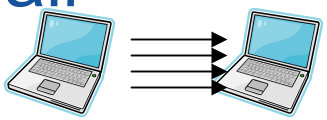
- Number of destination IPs for a given source IP
  - IP address fanout (sweep)



- Number of source IPs for a give destination IP
  - Denial of Service Attack



- Number of different TCP/UDP ports for a given source and destination address pair
  - TCP/UDP port scan



- Ratio of small packets to big packets

Current state of data collection

- 1 IA-64 1.6G server with 2GB RAM and afs scratch space as a temporary storage
  - CINBAD sflow collector
- 101 devices with sflow enabled (90 switches, 11 routers) in four buildings
  - CINBAD snmp configurator
- ~1600 active interfaces
- ~2000 samples /second
- ~40GB/ day

Visit to HP ProCurve in Roseville (1)

- Series of meetings with various ProCurve engineers and mathematician from HP Labs
  - Anomaly detection
    - aggregates that could be useful to reveal network anomalies
      - all aggregates are biased by sampling
      - flow estimation from sflow data seems to be inaccurate and computationally expensive
      - simple volume metrics are used in practical applications
      - entropy is promising since is more resistant to sampling
    - Anomaly detection algorithms
    - Review of the CERN list of network anomalies

Visit to HP ProCurve in Roseville (2)

- sFlow and snmp implementation issues in ProCurve switches
  - List of potential improvements
- Virus Throttling (VT) mechanism
  - anomaly detection (IP fanout) in the switch
  - access to the full network traffic, small computing power
- New data source for the CINBAD project
  - Information about new flows using existing traffic mirroring feature with Access Control List (ACL)

Conclusion

- We achieved the prototype implementation of a sFlow collector and snmp configurator
- We gradually collect more and more sFlow data
  - without side effects on our network infrastructure
- We have been collecting the requirements for data anomaly detection within CERN
  - to be continued at ProCurve