Scitags: A Standardized Framework for Traffic Identification and Network Visibility in Data-Intensive Research Infrastructures

Shawn McKee (University of Michigan Physics), Marian Babik (CERN), Tim Chown (Jisc), Andrew Hanushevsky (Presenter, SLAC National Accelerator Laboratory), Andy Lake (ESnet), Tristan Sullivan (University of Victoria), Bruno Hoeft (Karlsruhe Institute of Technology (KIT)), James Letts (University of California, San Diego), Dale Carder (ESnet), Garhan Attebury (UNL), Michael Lambert (Pittsburgh Supercomputing Center), Joe Mambretti (Northwestern University), Karl Newell (Internet2)

CHEP 2024, Krakow, Poland

https://indico.cern.ch/event/1338689

October 21-25, 2024

Presentation Overview

For High-Energy Physics (HEP), we have identified a need to better understand and optimize our network traffic to ensure we are using the network as effectively (for our science) as possible.

One of the challenges we have faced is being able to understand and identify the source of our traffic within the Research and Education (R&E) networks, especially when **critical links** are **overloaded**, impacting our workflows and data transfers.

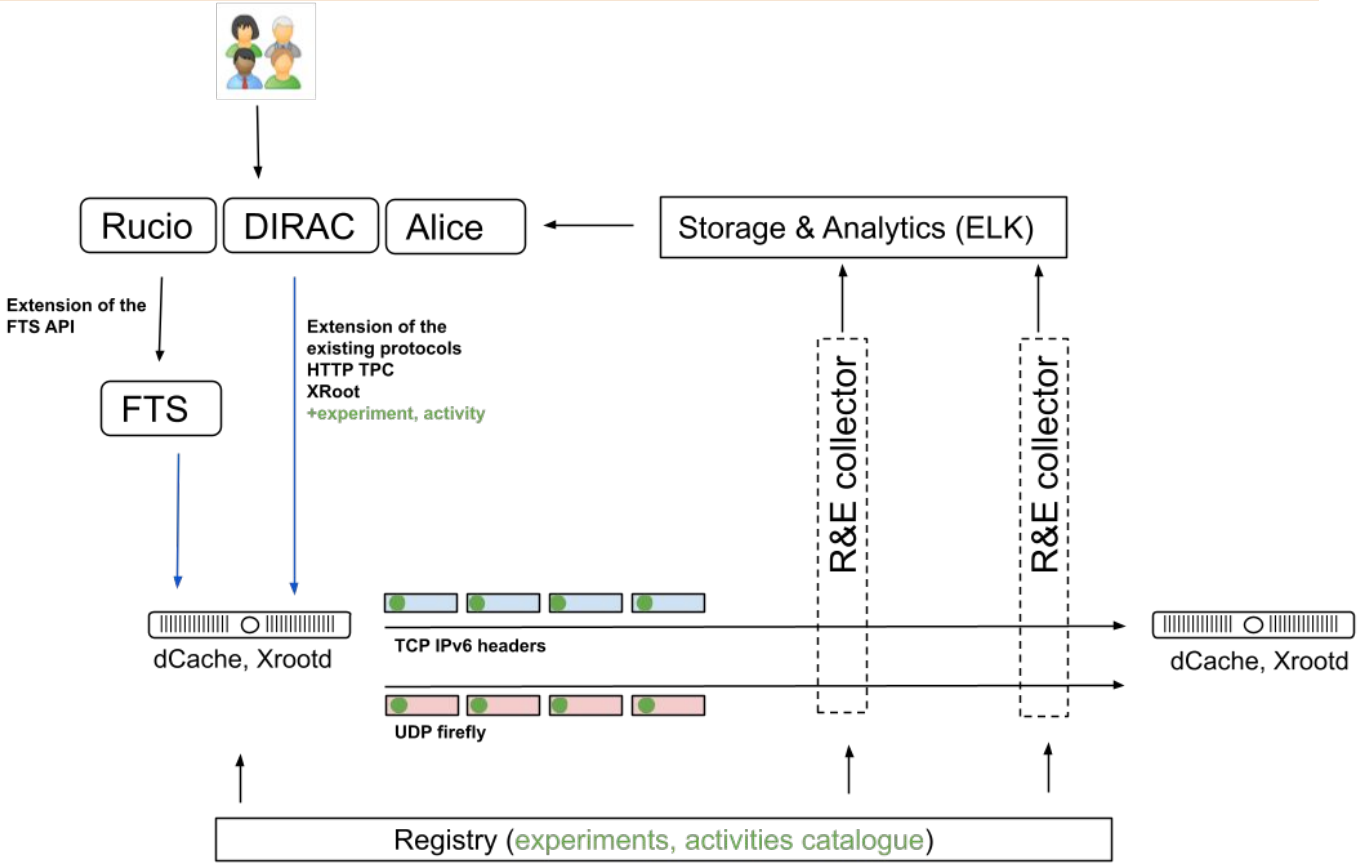
This presentation will cover the ongoing work to understand and identify our scientific network flows.

Network Visibility and Scitags

- Scientific Network Tags (scitags) is an initiative promoting identification of the science domains and their high-level activities at the network level.

- Enable tracking and correlation of our transfers with **Research and Education Network Providers** (R&Es) network flow monitoring
  - Utilizing packet and flow marking to identify traffic owner/purpose.
- **Experiments** can better understand how their network flows perform along the path
  - Improve visibility into how network flows perform (per activity) within R&E segments
  - Get insights into how experiments are using the networks, get additional data from R&Es on behaviour of our transfers (bottlenecks, troubleshooting, optimisation, etc.)
- **Sites** could get visibility into how different network flows perform

How scitags work



Registry

We have standardized the “experiment” and “activity” fields we use for both flow labeling and packet marking.

The **scitags.org** domain provides an API that can be consulted to get the standard values: https://api.scitags.org or https://www.scitags.org/api.json

The underlying source of truth is a set of Google sheets that are maintained and writeable by a few stewards.

Note: the API provides the defined values **but** how the values are used in packet marking are specified in our Google sheets (bit location in IPv6 flow label)


```
{
  - experiments: [
    - {
        expName: "default",
        expId: 1,
      - activities: [
        - {
            activityName: "default",
            activityId: 1
          }
        ]
      },
    - {
        expName: "atlas",
        expId: 2,
      - activities: [
        - {
            activityName: "perfsonar",
            activityId: 2
          },
        - {
            activityName: "cache",
            activityId: 3
          },
        - {
            activityName: "datachallenge",
            activityId: 4
          },
        - {
            activityName: "default",
            activityId: 8
          },
        - {
            activityName: "analysis download",
            activityId: 9
          },
        - {
            activityName: "analysis download direct io",
            activityId: 10
```

Finding More Information: https://scitags.org

Code

Technical Spec

Mailing List

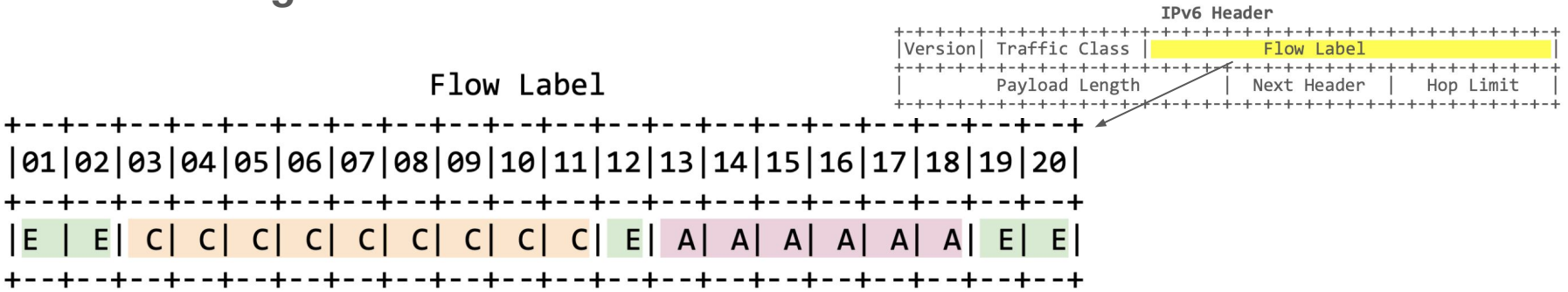
Presentations

Scitags Framework Rationale

- **Open platform** to be used by any data-intensive science community
- **Identify the owner (experiment) and purpose (activity) of the traffic**
- Define a **standard(s)** for exchange of information between scientific communities, sites and network operators
  - Packet marking - encoding exp/activity directly in packets
  - Flow labeling - sending a separate UDP packet (firefly) with metadata
- **Enable tracking and correlation with existing network flow monitoring and existing monitoring systems deployed by R&E networks**
- Quantify global behaviour and analyse trade-offs at scale

Technical Spec for Packet Marking

Packet Marking via the use of the IPv6 Flow Label



- (C) Community identifier: "Who are you affiliated with?"
- (A) Activity identifier: "What are you doing within your community?"
- (E) Entropy bits sprinkled throughout

IETF RFC-Informational Draft is available with more details

Started exploring HbH option as an alternative (eBPF-PDM, eBPF-extHeaders)

Technical Spec for Flow Labeling

- **Flow Labeling** via UDP packets (fireflies):
  - **Fireflies** are UDP packets in Syslog format with a defined, versioned JSON schema.
    - Packets are intended to be sent to the same destination (port 10514) as the flow they are labeling and these packets are intended to be world readable.
    - Packets can also be sent to specific regional or global collectors.
    - Use of syslog format makes it easy to send to Logstash or similar receivers.
    - Works for IPv4 and IPv6; content is not limited (as long as it fits in a single frame)
      - Apart from exp/act we now have also usage (bytes sent/rcv) and RTT in fireflies

- The detailed technical specifications are maintained on a Google doc
- The document also covers methods for communicating owner/activity and other services and frameworks that may be needed for implementation.

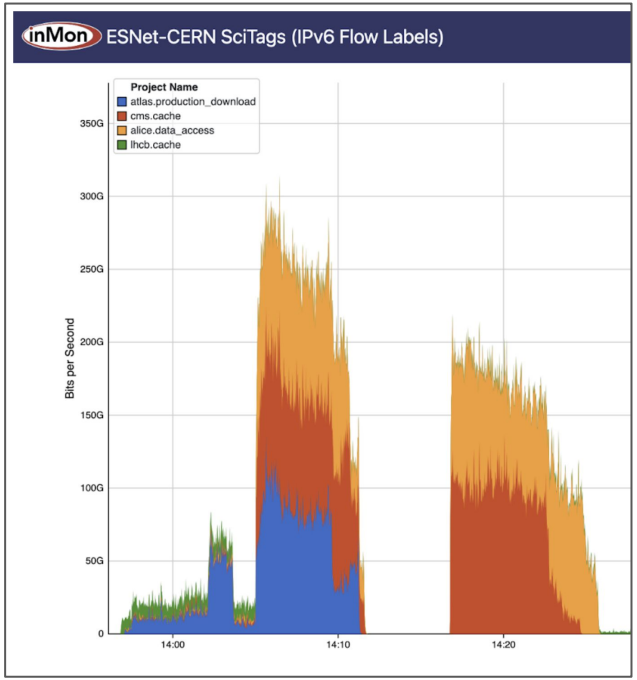
Collectors

- Different types depending on what is being collected
  - **HW/on-the-wire** to collect UDP fireflies and/or IPv6 flow label
  - **SW/network of receivers** - collecting UDP fireflies sent to them
    - Working on update to the current architecture to introduce a message bus - to interconnect different (N)REN collectors and also allow anyone to subscribe
  - **SW/Collectors**
    - Site-collector - forwards fireflies via UDP, optional local storage
    - Regional collector - receives fireflies from sites, stores locally and publishes to message bus
    - Global collector - receives all fireflies (directly or via bus), global store
    - Experiments collector - subscribes to the bus for specific fireflies

Packet Marking Demo

During Supercomputing 23 in Denver, we demonstrated a number of aspects of our packet and flow marking work.

- Showed **packet marking** at **300 Gbps** rates using **xrootd**/iperf3 (with just two nodes; using eBPF).
- Integration with **ESnet's High-Touch Service**
  - Enabling analytics at the packet-level
- In collaboration with inMon Corp, set up packet collectors via sflow and demonstrated **real-time monitoring of flows by community/activity**.
- SC23 demo was run in collaboration with Starlight, ESnet, KIT, University of Victoria, University of Nebraska and CERN



Data Challenge 24

- **Scitags Deployment**
  - 80% of EOS CMS (production), UNL production storage
  - Flow labeling functionality (fireflies)
- **Results:**
  - **Confirmed the capability to propagate Scitags all the way to the storages (for both ATLAS and CMS)**
  - Sending fireflies (from XRootd, EOS storages)
  - **Collection and visualisation at ESnet collector**
    - Results shown in live dashboard
  - With limited deployment we were able to get valuable insights into flow durations, their characteristics (splits by exp/activity), sources of IPv4 traffic (split by applications) and potential impact of new TCP congestion algorithms (performance correlated with flow data)

Current Status

Implementation status:

Propagation:

- **Rucio** supports Scitags from 32.4.0
- **FTS/gfal2** support Scitags from 3.2.10/2.21.0

Storages:

- **XRootD** provides Scitags implementation (from 5.0+)
- **EOS** provides Scitags support from 5.2.19+
  - Project on production rollout at CERN for WLCG has been approved
- **dCache** prototype exists, roadmap for release pending
- StoRM prototype exists, planning release and pre-production deployment

Collectors:

- Production deployments at ESnet and Jisc

Summary

- Scitags (flow labelling) ready for production
  - Expecting sites and experiments to gradually enable it during this year
    - **ATLAS, CMS and ALICE ready to enable in production**
    - Fireflies at CERN T0 already enabled for CMS, Alice and soon ATLAS
    - We're encouraging all sites with supported storages to enable scitags in production
  - Network providers actively working on getting SW/Collectors network in production
  - Scitags adoption in DC24 demonstrated its potential for network usage analysis, especially when integrated with external R&E monitoring tools

- Significant progress also in packet marking R&D
  - Planning to test it in production this year

Acknowledgements

We would like to thank the **WLCG**, **HEPiX**, **perfSONAR** and **OSG** organizations for their work on the topics presented.

In addition we want to explicitly acknowledge the support of the **National Science Foundation** which supported this work via:

- OSG: NSF MPS-1148698
- IRIS-HEP: NSF OAC-1836650

Backup slides

Useful URLs

RNTWG Google Folder
RNTWG Wiki
RNTWG mailing list signup

HEPiX NFV Final Report WG Report

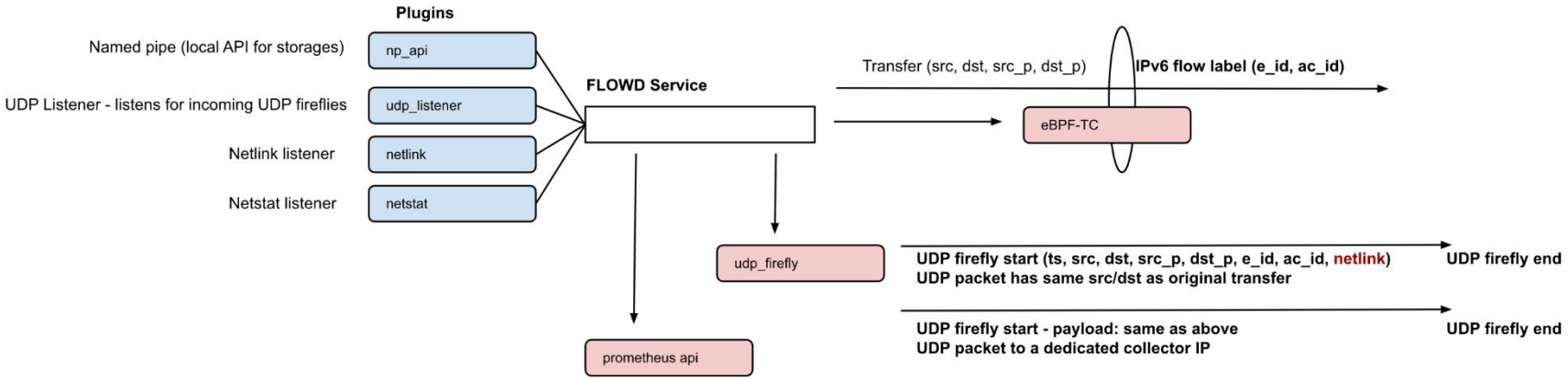
RNTWG Meetings and Notes: https://indico.cern.ch/category/10031/

The scitags web page: https://scitags.github.io

Code at https://github.com/scitags/scitags.github.io

End-system Utility: Flowd Service

- Flow and Packet Marking service developed in Python

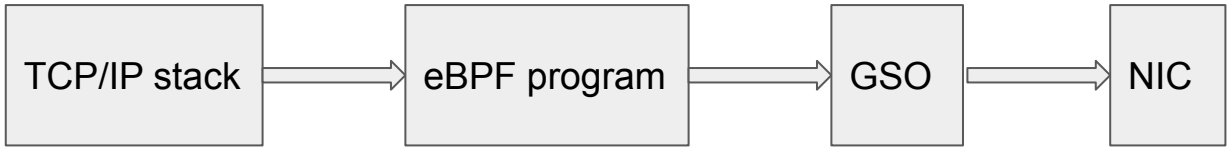


- Plugins provide different ways get connections to mark (or interact with storage)
  - New plugins were added to support netlink readout and UDP firefly consumer
- Backends are used to implement flow and/or packet marking
  - New backends were added to mark packets (via eBPF-TC) and expose monitored connection to Prometheus

Flowd: Packet Marking via eBPF-TC Backend

- eBPF is a general-purpose RISC instruction set that runs on an in-kernel VM; programs can be written in restricted C and compiled into bytecode that is injected into the kernel (after verification)
- Can sometimes replace kernel modules
- eBPF-TC programs run whenever the kernel receives (ingress) or sends (egress) a packet

Egress path:



- The flowd backend maintains a hash table of flows to mark. The plugin sends the backend (src address, dst address, src port, dst port); this is used as the key in the hash, and the flow label to put on the packets is the value
- Each packet is inspected, and if the attributes match an entry in the hash, the corresponding flow label is put on the packet

NOTE: SciTag Firefly Implications

One quick heads-up for sites and network providers: we are beginning to send **UDP fireflies** from some of our sites.

UDP fireflies (by default) are sent to the same destination as the data transfer flow. This means UDP packets arriving at storage servers on port 10514.

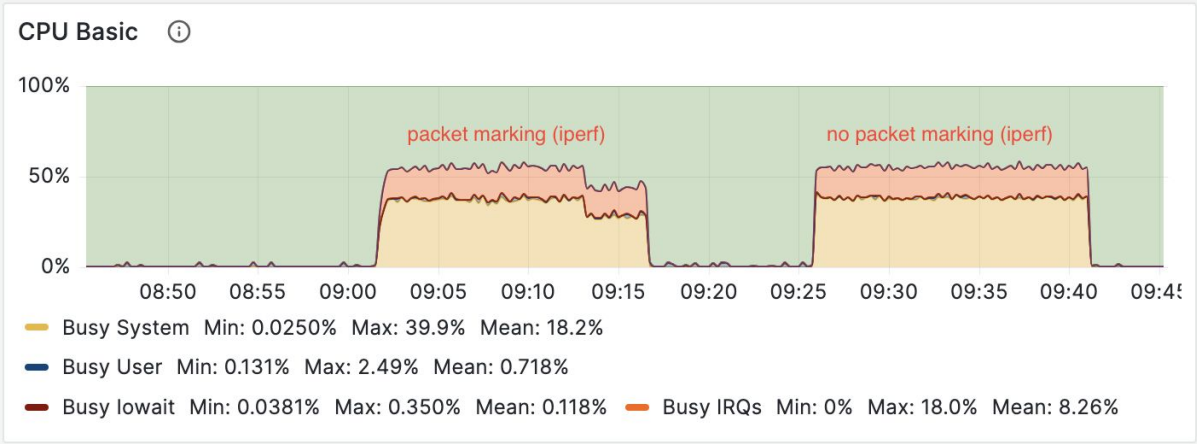
A site can choose to ignore, block or capture these packets

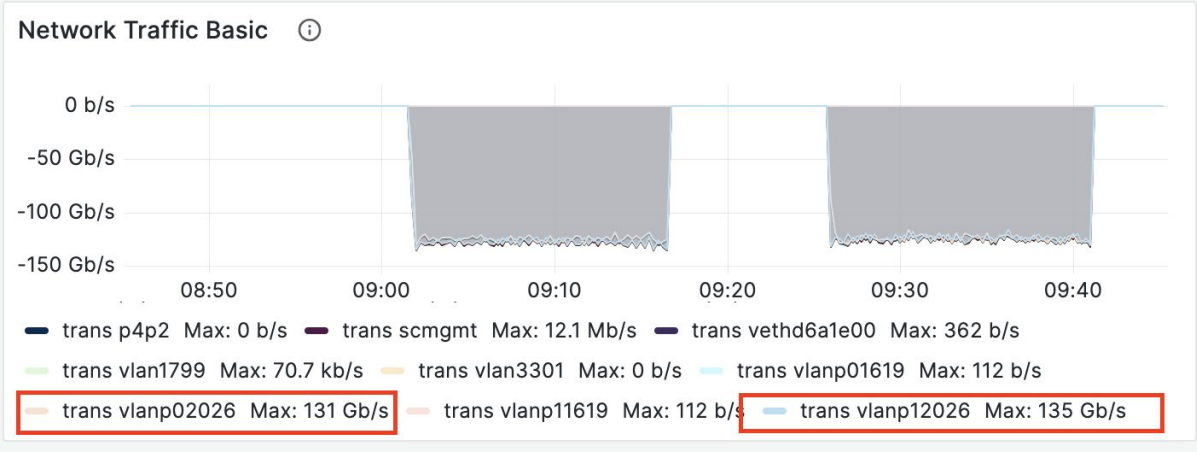
We are working on an informational RFC (target to publish Fall 2023)

One implication: if packets hit iptables, it may generate noise in the logging that may be a concern (fill /var/log?)

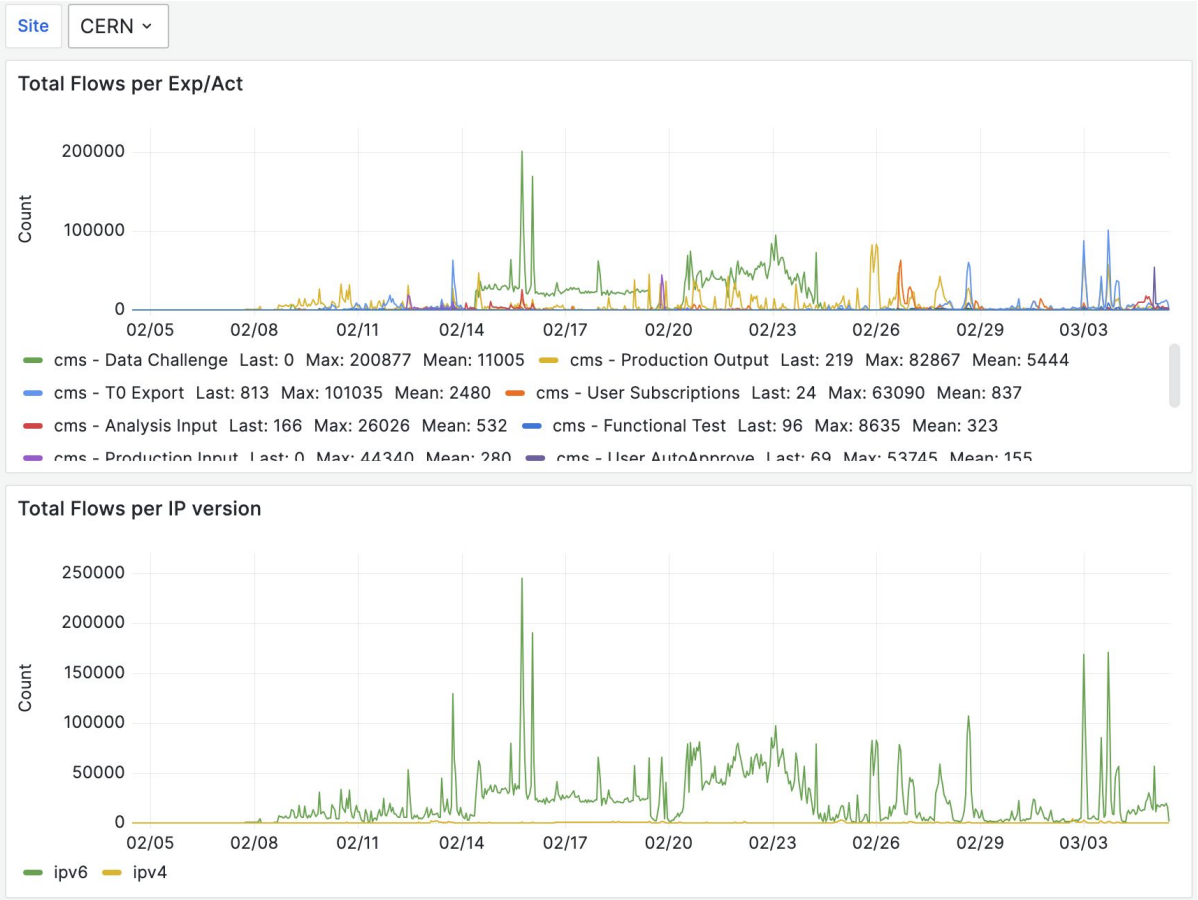
Recommendation is to open port 10514 for incoming UDP packets or explicitly ‘drop’ them.

Packet and Flow Marking Demos

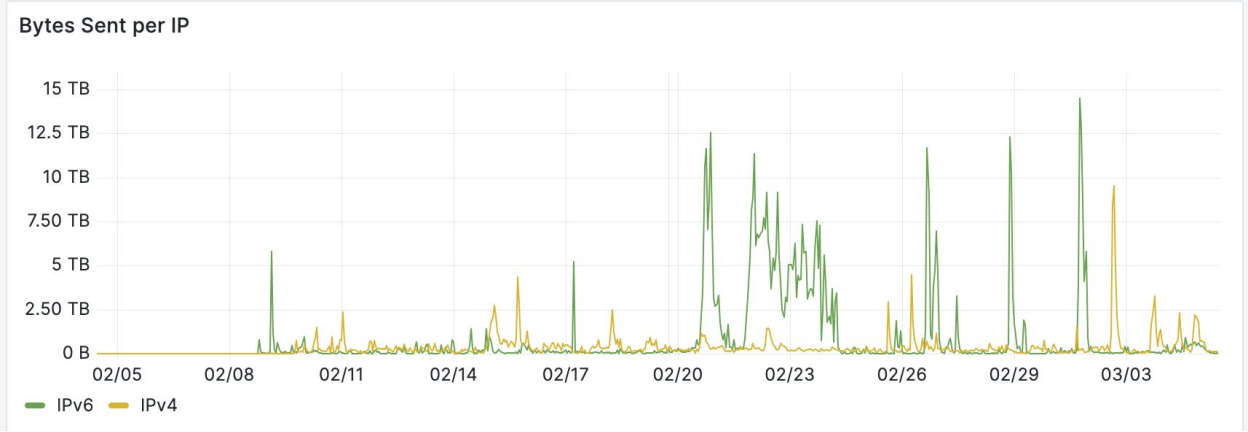


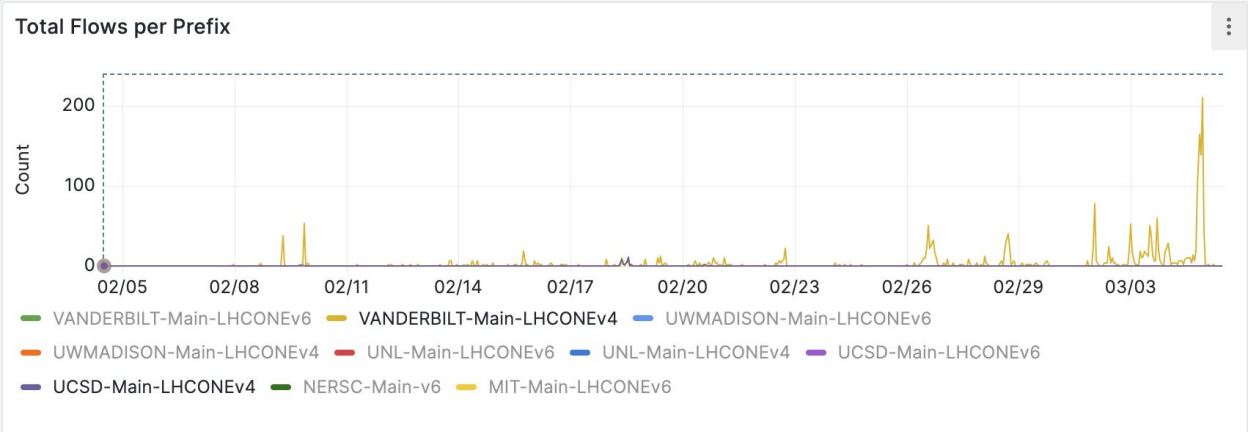


DC24: CERN EOS CMS plot showing split by experiment/activity and IP

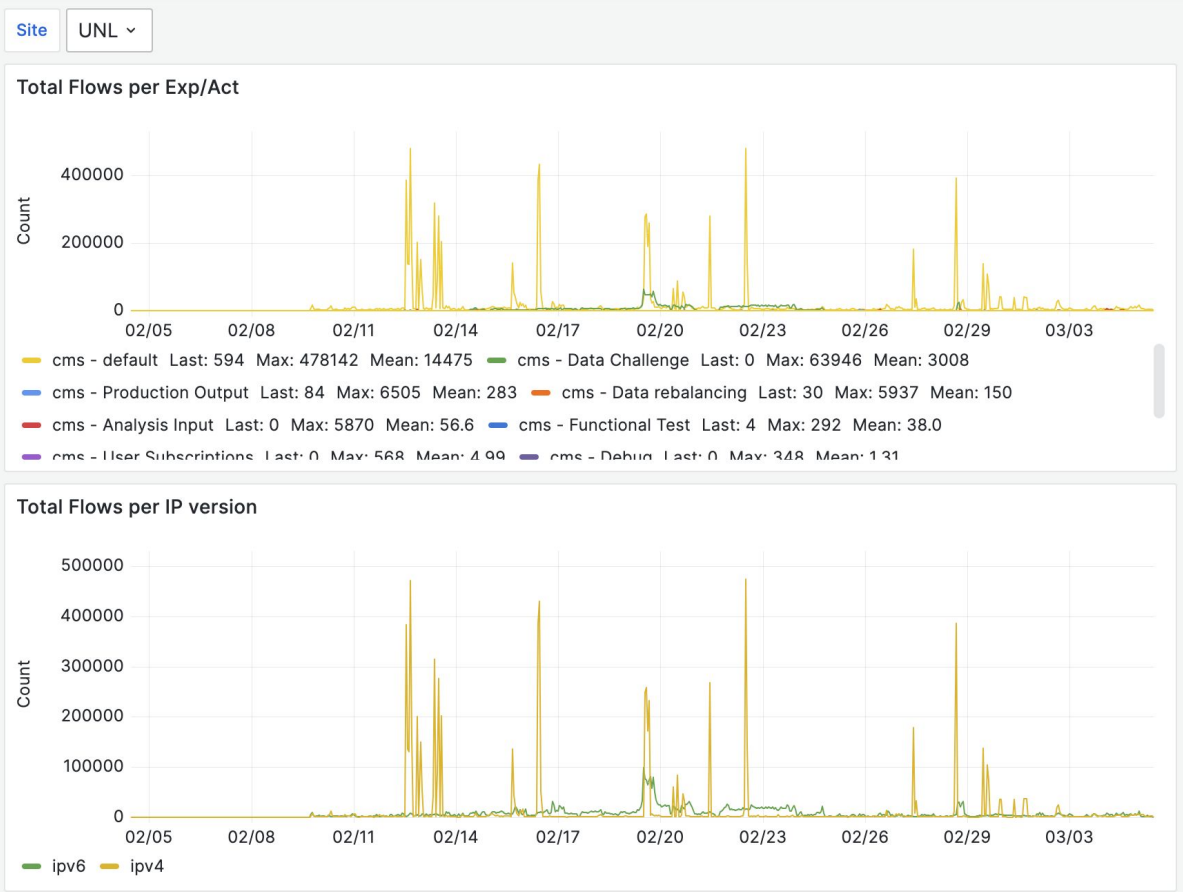


DC24: CERN EOS CMS plot showing split by IP total traffic
Split by IP prefix (incomplete, but shows dest for some of the IPv4 traffic)





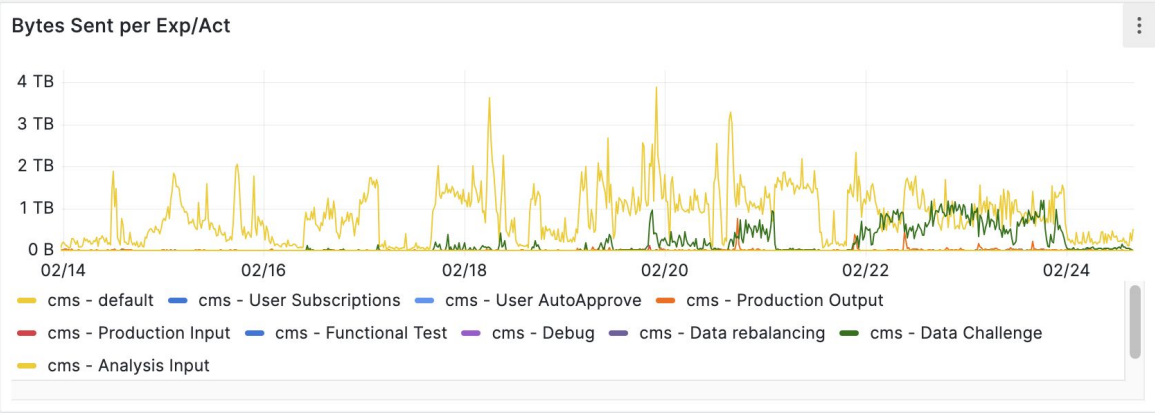
DC24: University of Nebraska (UNL) - many flows not coming from FTS
DC flows were only a small subset (IP split shows a very different profile)

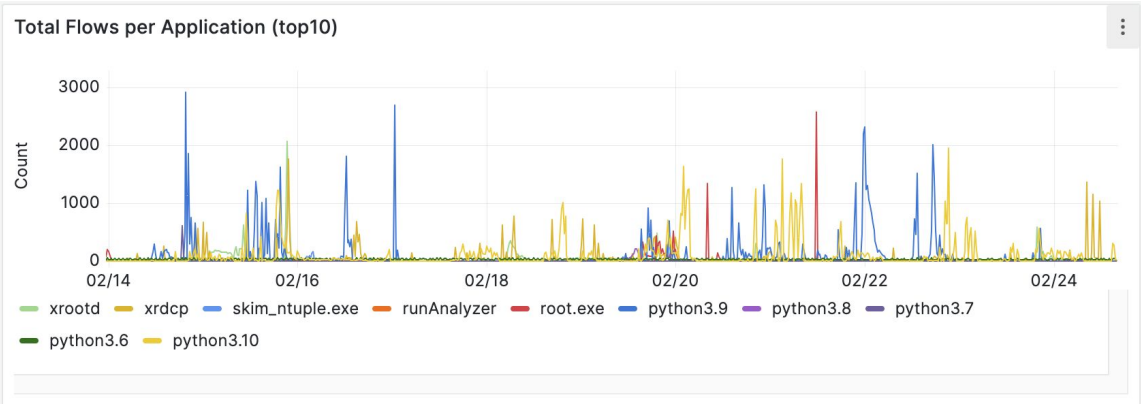


DC24: University of Nebraska (UNL)

Bytes sent per Exp/Activity - shows non-FTS traffic was dominant

Capability to show a split by application (as reported by xrootd)

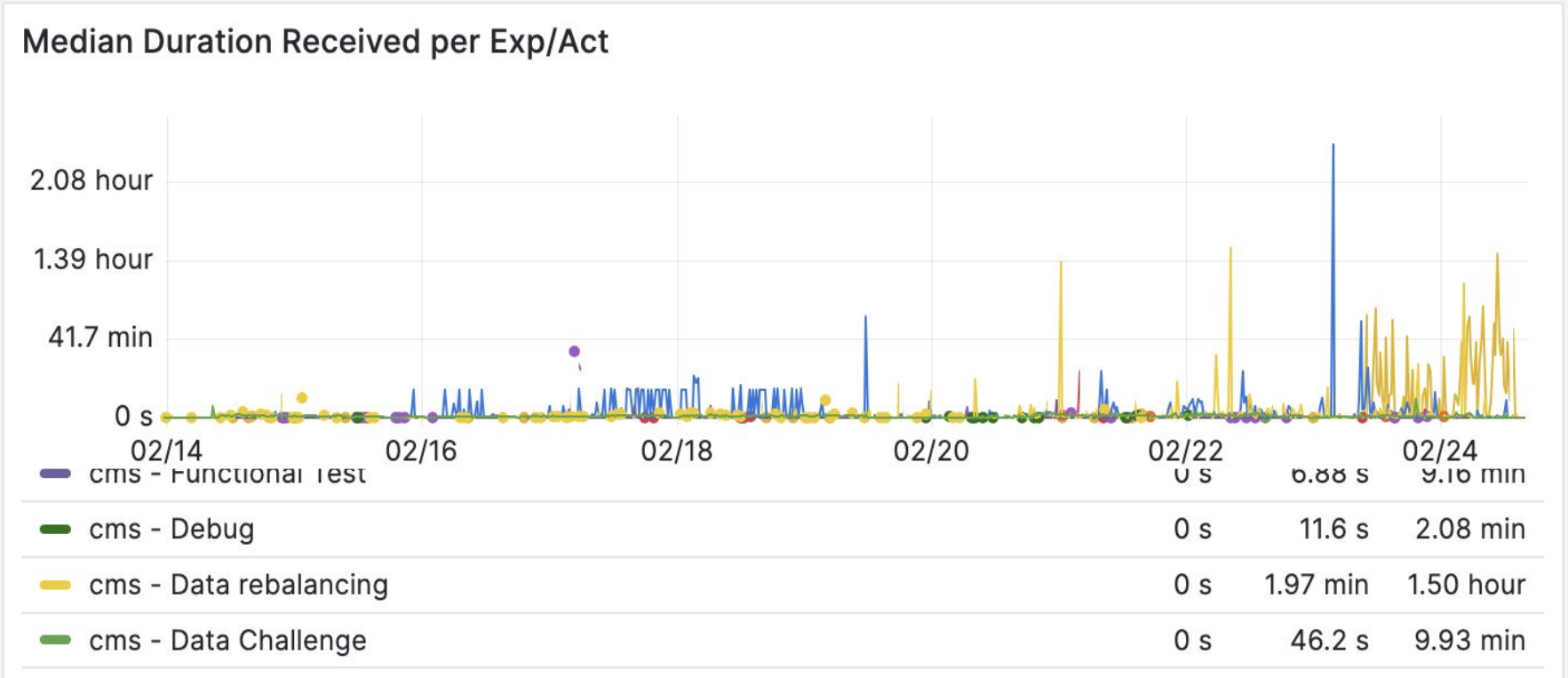




DC24: CERN EOS CMS

Median duration of flows split by Exp/Activity

Shows duration of DC flows was quite short wrt. production/rebalancing



DC24: CERN EOS CMS

Max duration of flows split by Exp/Activity

First week had a lot of “fat” flows from production activity (but none from DC)

Second week was different, some DC flows took hours to finish