# Research Networking Technical WG Update

Shawn McKee / University of Michigan and Marian Babik / CERN

#48 LHCONE/LHCOPN Meeting

(https://indico.cern.ch/event/1110783/)

March 30, 2022

# History

- HEPiX Network Functions Virtualisation Working Group
  - Working Group Report was published at the end of 2019 with three chapters
    - Cloud Native DC Networking
    - Programmable Wide Area Networks
    - Proposed Areas of Future Work
- LHCOPN/LHCONE workshop (spring 2020)
  - Requirements on networks from the WLCG experiments
- Research Networking Technical Working Group
  - Formed after the workshop in response to the requirements discussion
  - 98 members from ~ 50 organisations have joined
  - Three main areas of work:
    - **Network Visibility: Packet and Flow Marking** - viewed as the appropriate first step; regular meetings every ~2 months since summer 2020
      - Packet Marking Document
        - Outlines available technologies, standards and stakeholders perspectives
        - This has led to Scientific Network Tags (scitags) initiative, which is presented today
    - **Traffic Shaping** - Using techniques like packet pacing to achieve consistent throughput.
    - **Network Orchestration** - followed up by GNA-G, SENSE and FABRIC

# Network Visibility Motivation

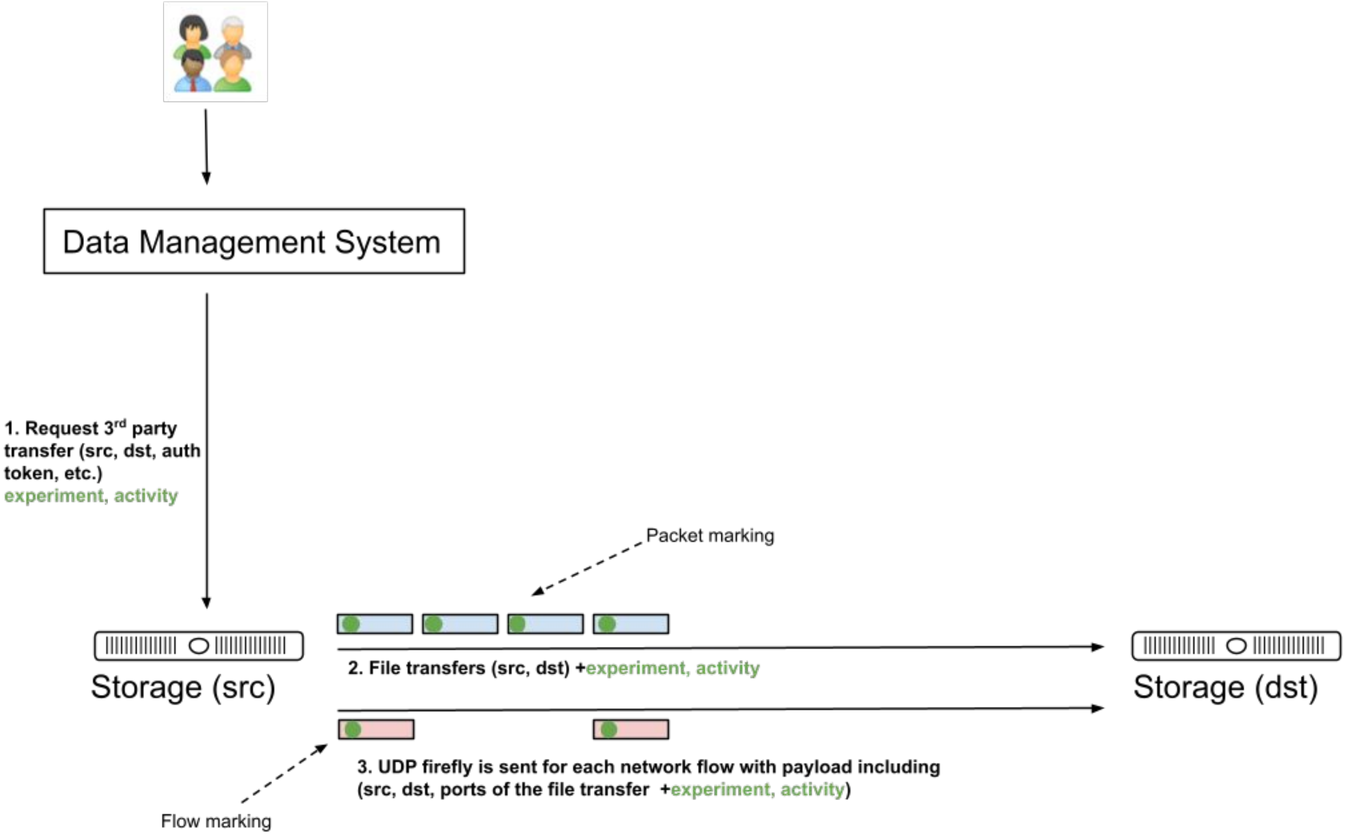
- Networks are becoming more programmable and capable with technologies such as P4, SDN, virtualisation, eBPF, etc.
- But with less and less context about the traffic they carry.
  - Cloud deployments, Kubernetes, encryption, tunneling, privacy, etc.
- Understanding scientific traffic flows in detail is critical for understanding how our complex systems are actually using the network.
  - Current monitoring/logging tell us where data flows start and end, but is unable to understand the data in flight.
  - Dedicated L3VPNs can be created to track high throughput science domains, but with more domains requiring high throughput this will become expensive, it won't scale, won't work at big sites having to support multiple domains at the same time.
- In general the monitoring we have is experiment specific and very difficult to correlate with what is happening in the network. We suggest this is a general problem for users of the Research and Education Networks (RENs).

# 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
- 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 experiment is using the networks, get additional data from R&Es on behaviour of our transfers (traffic, paths, etc.)
- Sites can get visibility into how different network flows perform
  - Network monitoring per flow (with experiment/activity information)
    - E.g. RTT, retransmits, segment size, congestion window, etc. all per flow

# How scitags work



Data Management System

1. Request 3rd party transfer (src, dst, auth token, etc.) experiment, activity

Packet marking

Storage (src)

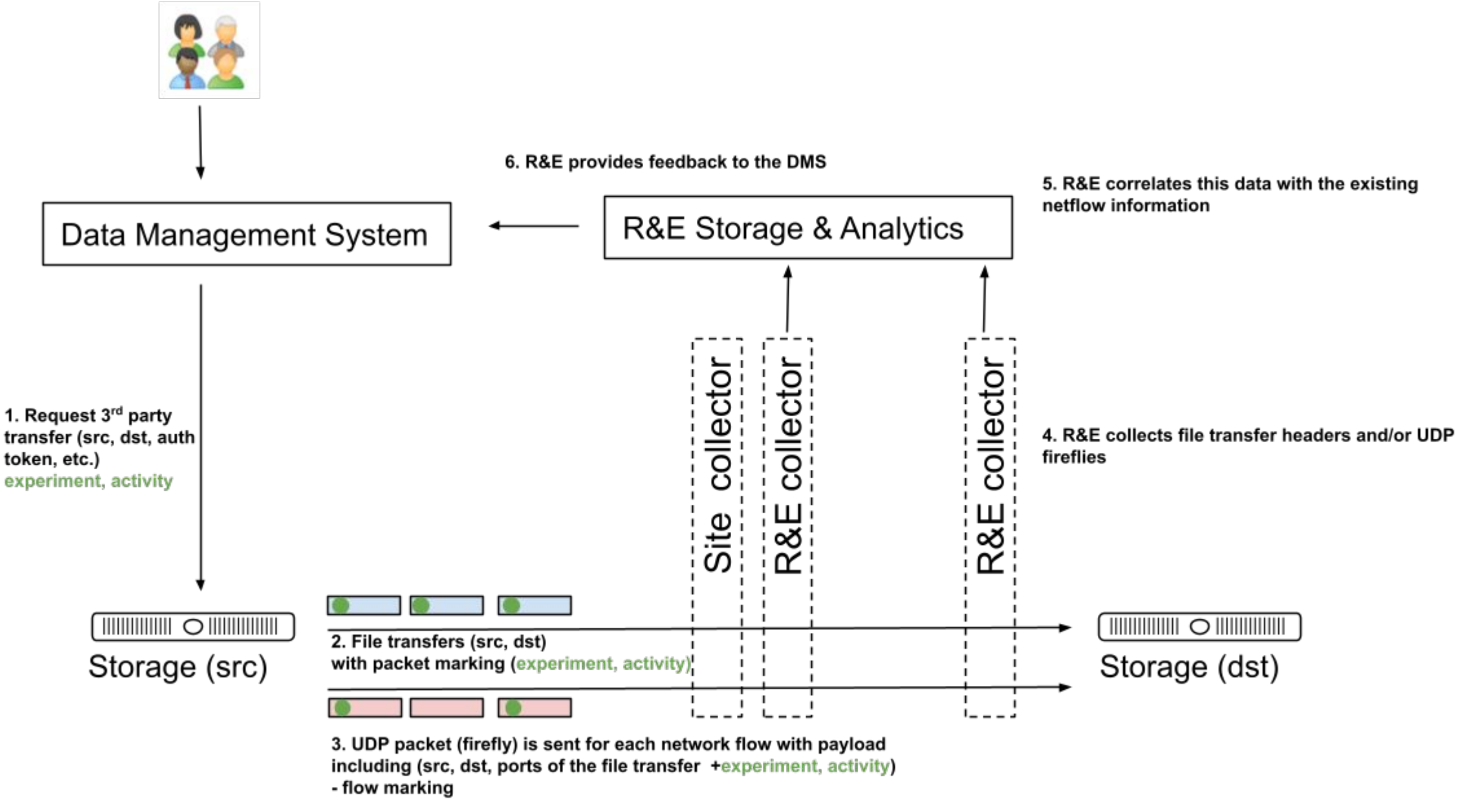
2. File transfers (src, dst) +experiment, activity

Storage (dst)

3. UDP firefly is sent for each network flow with payload including (src, dst, ports of the file transfer +experiment, activity)

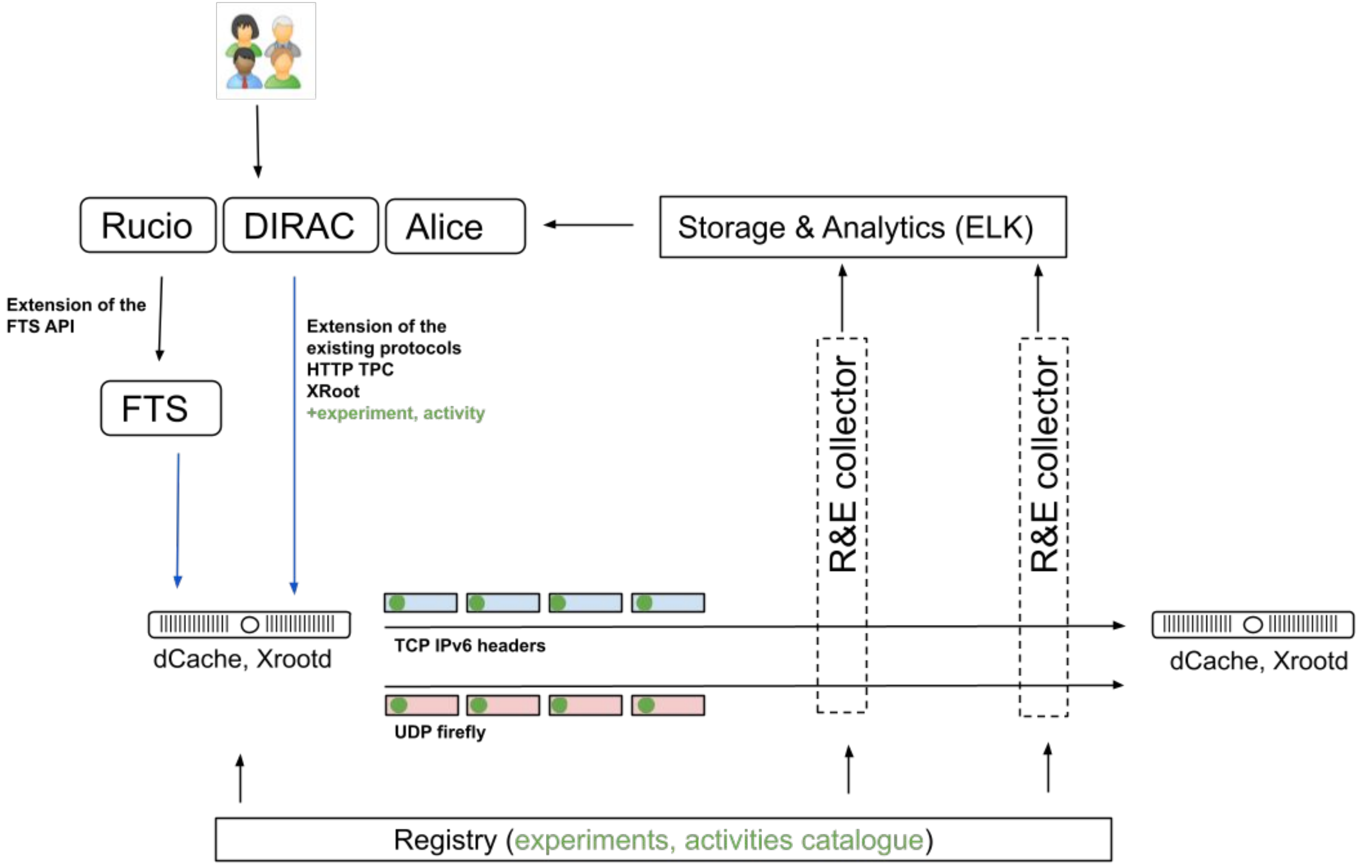
Flow marking

# How scitags work



6. R&E provides feedback to the DMS
Data Management System
R&E Storage & Analytics
5. R&E correlates this data with the existing netflow information
1. Request 3rd party transfer (src, dst, auth token, etc.) experiment, activity
Site collector
R&E collector
R&E collector
4. R&E collects file transfer headers and/or UDP fireflies
Storage (src)
2. File transfers (src, dst) with packet marking (experiment, activity)
Storage (dst)
3. UDP packet (firefly) is sent for each network flow with payload including (src, dst, ports of the file transfer +experiment, activity) - flow marking

# How scitags work


Rucio
DIRAC
Alice
Storage & Analytics (ELK)
Extension of the FTS API
FTS
Extension of the existing protocols HTTP TPC XRoot +experiment, activity
R&E collector
R&E collector
dCache, Xrootd
TCP IPv6 headers
UDP firefly
dCache, Xrootd
Registry (experiments, activities catalogue)

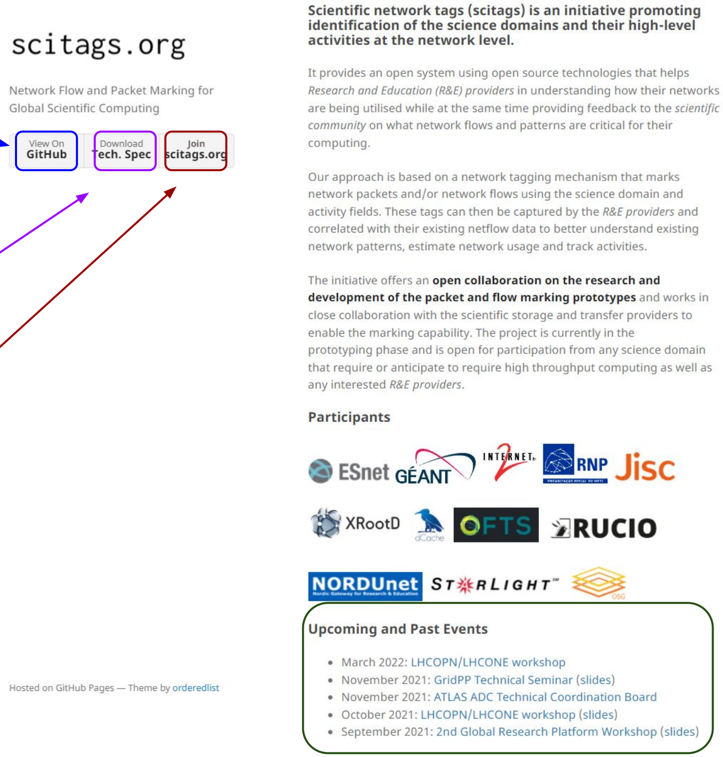
# Finding More Information: https://scitags.org

Code

Technical Spec

Mailing List

Presentations



scitags.org

Network Flow and Packet Marking for Global Scientific Computing

Hosted on GitHub Pages — Theme by orderedlist

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

It provides an open system using open source technologies that helps *Research and Education (R&E) providers* in understanding how their networks are being utilised while at the same time providing feedback to the *scientific community* on what network flows and patterns are critical for their computing.

Our approach is based on a network tagging mechanism that marks network packets and/or network flows using the science domain and activity fields. These tags can then be captured by the *R&E providers* and correlated with their existing netflow data to better understand existing network patterns, estimate network usage and track activities.

The initiative offers an **open collaboration on the research and development of the packet and flow marking prototypes** and works in close collaboration with the scientific storage and transfer providers to enable the marking capability. The project is currently in the prototyping phase and is open for participation from any science domain that require or anticipate to require high throughput computing as well as any interested *R&E providers*.

**Participants**

**Upcoming and Past Events**

- March 2022: LHCOPN/LHCONE workshop
- November 2021: GridPP Technical Seminar (slides)
- November 2021: ATLAS ADC Technical Coordination Board
- October 2021: LHCOPN/LHCONE workshop (slides)
- September 2021: 2nd Global Research Platform Workshop (slides)

# Technical Spec

The detailed technical specifications are maintained on a Google doc

- The spec covers both Flow Labeling via UDP Fireflies and Packet Marking via the use of the IPv6 Flow Label.
  - **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.
  - **Packet marking** is intended to use the 20 bit flow label field in IPv6 packets.
    - To meet the spirit of RFC6437, we use 5 of the bits for entropy, 6 for activity and 9 for owner/experiment.
- The document also covers methods for communicating owner/activity and other services and frameworks that may be needed for implementation.

# Status

- **Flow Marking** (UDP firefly) implementations
  - Xrootd 5.4.0 supports UDP fireflies
    - https://xrootd.slac.stanford.edu/doc/dev54/xrd_config.htm#_pmark
    - map2exp - can be used to map particular path to an experiment
    - map2act - can be used to map particular user/role to an activity
  - Flowd - prototype service
    - Issue fireflies from netstat for a given experiment (only for dedicated storages)
- **Collectors**
  - Initial prototype was developed by ESnet (will be available on scitags github soon)
  - ESnet and Jisc/Janet*
- **Registry**
  - Provides list of experiments and activities supported
  - Exposed via JSON at api.scitags.org
- Simplified deployment was tested during the last DC (& still operating)
  - Flowd + ESnet collector + Registry
  - **AGLT2, BNL, KIT, UNL and Caltech** participated
  - Brunel, Glasgow and QMUL interested to help with further testing

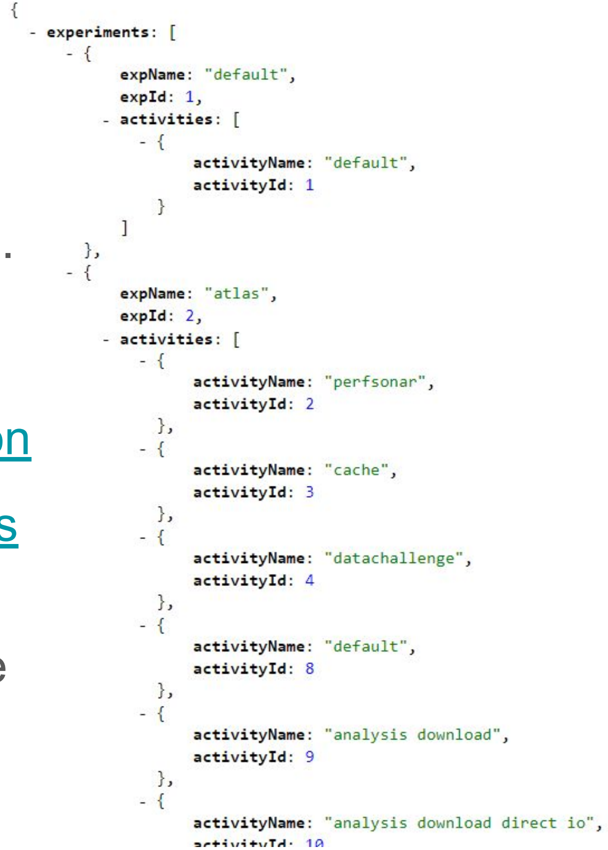
# Registry

We need to standardize 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.

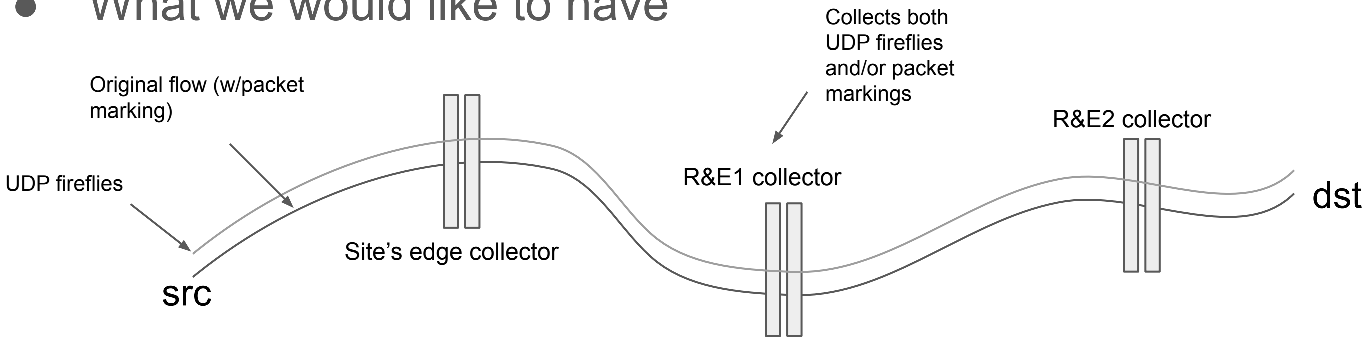
**Note**:  the API provides the defined values **but** how the values are used in packet marking are specified in our Google sheets (bit location in IPv6 flow label)



```
{
  - 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
```

# Collectors

- What we would like to have

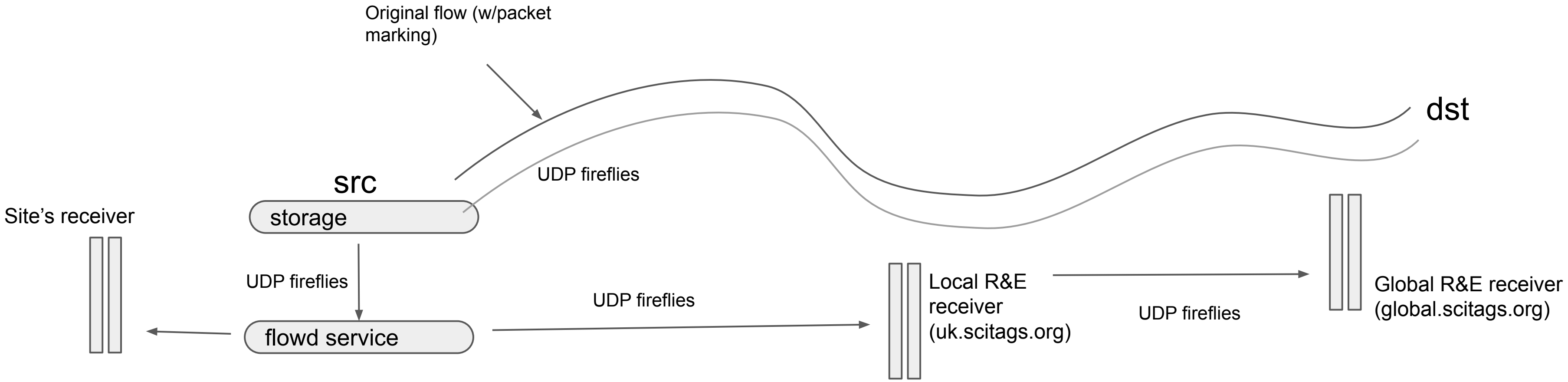

Original flow (w/packet marking)
UDP fireflies
src
Site's edge collector
Collects both UDP fireflies and/or packet markings
R&E1 collector
R&E2 collector
dst

- Enable collection of packet and flow markings along the path
  - In order to extend R&E netflow information with flow identifier (experiment + activity)
  - UDP firefly packets needs to be collected and relayed to ensure they reach all collectors
- Each R&E **can** setup and operate one or more collectors
- Sites have an option to set up their own collector at the edge

# Collectors

- Our **recommendation** is to use hardware/in-line collectors where possible
    - Requires port mirroring or other means to capture the fireflies.
    - Easiest to organise and operate as there is no need for a separate collector network.
    - Only way to capture flow markings along the path.
- However, in-line collectors require the ability to either selectively identify and capture fireflies or the ability to capture IPv6 flow labels from packets
    - Many possible ways to implement.
    - Strategy and technology to implement will depend on the R&E, their topology and hardware.
    - Would be great to get example implementations that can be shared between R&E network operators.

# Network of receivers


Original flow (w/packet marking)
dst
UDP fireflies
src
Site's receiver
storage
UDP fireflies
UDP fireflies
flowd service
Local R&E receiver (uk.scitags.org)
UDP fireflies
Global R&E receiver (global.scitags.org)

- Storages are configured with predefined DNS aliases (based on region; hosted by scitags.org)
  - Flowd service will expose API for site's local receivers and will also forward UDP fireflies to R&E collector (storage will send fireflies along the path)
  - Local R&E collector can be established (optional) and will need to pass all received fireflies to the global one (can switch to TCP)
- Works with inline/hardware collectors (which can be setup in parallel)
- Easy way to setup local R&E receiver (and correlate with local netflow)
- Lightweight - should be easy to operate, but requires some development in flowd and in the R&E collector
- DNS aliases will give us flexibility to make changes in the future (e.g. move to anycast)

# Network Visibility Plans

- Near-term objectives
  - Start rollout and testing of Xrootd implementation
    - Detect flow identifiers from storage path/url, activities from user role mapping
    - Test proxies, cached proxies, private networks (K8s)
  - Finalise development and deploy network of receivers
  - Instrument Rucio/FTS to pass flow identifiers to the storages
  - Involve other storage systems (dCache, etc.); discuss possible design/implementation

- Engage other R&Es and explore available technologies for collectors
  - Deploy additional collectors and perform R&D in the packet collectors
  - Improve existing data collection and analytics
- Test and validate ways to propagate flow identifiers
  - Engage experiments and data management systems
  - Validate, test protocol extensions and FTS integration
  - Explore other possibilities for flow identifier propagation, e.g. tokens
- R&D activities
  - **Packet marking** - further testing and validation is required for IPv6 flow label implementation (next meeting or two)

# Traffic Shaping and Network Orchestration

The RNTWG has (so far) focused on the network visibility area due to limited manpower, but we have two additional areas the are part of the overall group goal:  traffic shaping and network orchestration

**Traffic Shaping:**

- The WLCG experiments would like to explore traffic shaping/packet pacing.
  - Without packet pacing, network packets are emitted by the network interface in bursts, corresponding to the wire speed of the interface.
    - **Problem**: microbursts of packets can cause buffer overflows
    - The impact on TCP throughput, especially for high-bandwidth transfers on long network paths can be **significant**.
- Instead, pacing flows to match expectations [min(SRC,DEST,NET)] smooths flows and significantly reduces the microburst problem.
  - An important extra benefit is that these smooth flows are much friendlier to other users of the network by not bursting and causing buffer overflows.
  - Broad implementation of pacing could make it feasible to run networks at much higher occupancy before requiring additional bandwidth

**Network Orchestration:**

- This effort is being led by the GNA-g and includes work from the SENSE and FABRIC projects.

# Summary

The RNTWG has made significant progress on network traffic visibility through the work on flow labeling and packet marking.

- There remains a significant amount of work to do, especially regarding enabling packet marking on our storage infrastructure and in the area of collecting, aggregating and making visible the marked traffic.

We have additional work to pursue in traffic shaping:

- While network orchestration has significant activity underway, we need to find new effort interested in developing, prototyping and evaluating traffic shaping

**We are always looking for additional manpower to join the effort!**

# Acknowledgements

We would like to thank the **WLCG**, **HEPiX**, **perfSONAR, CERN** 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

# Questions or Comments

Happy to take any questions, comments or suggestions!

# Backup slides

# WLCG Network Requirements

- Many WLCG facilities need network equipment refresh
  - Current routers in some sites are End-Of-Life and moving out of warranty
  - Local area networking often has 10+ year old switches which are no longer suitable for new nodes or operating at our current or planned scale.
- WLCG planning is including networking to a much greater degree than before
  - HL-LHC computing review: DOMA, dedicated networking section
  - ATLAS HL-LHC Computing Conceptual Design Report, highlights needs
  - Both include input from HEPiX, LHCONE/LHCOPN and WLCG working groups
- **Requirements Summary**
  - **Capacity**: Run-3 moving to multiple 100G links for big sites, Run-4 targeting Tbps links
  - **Capability**: WLCG needs to understand the impact of new features in networking (SDN/NFV) by testing, prototyping and evaluating impact. They will need to evolve their applications, facilities and computing models to meet the HL-LHC challenges; *it will take time*.
  - **Visibility**: As the ESnet Blueprinting meetings have shown, our ability to understand our WAN network flows is too limited. We need new methods to mark and monitor our network use
  - **Testing**: We need to be able to develop, prototype and test network features at suitable scale

# Packet Marking Challenges

We would like this to be applicable for ALL significant R&E network users/science domains, not just HEP

- Requires us to think broadly during design

How best to use the number of bits we can get?

- Need to standardize bits and publish and **maintain**!!
- Can we agree on some standard “type” bits?

What can we rely on from the Linux network stack and what do we need to provide?

What can the network operators provide for accounting?

# Packet Marking - Storage Elements

The primary challenge here is in two areas:

1. Augmenting the existing storage system to be able to set the appropriate bits in the network packets
2. Communicating the appropriate bits as part of a transfer request
    a. Likely need some protocol extension to support this
    b. Other ideas?

# Packet Marking - Jobs

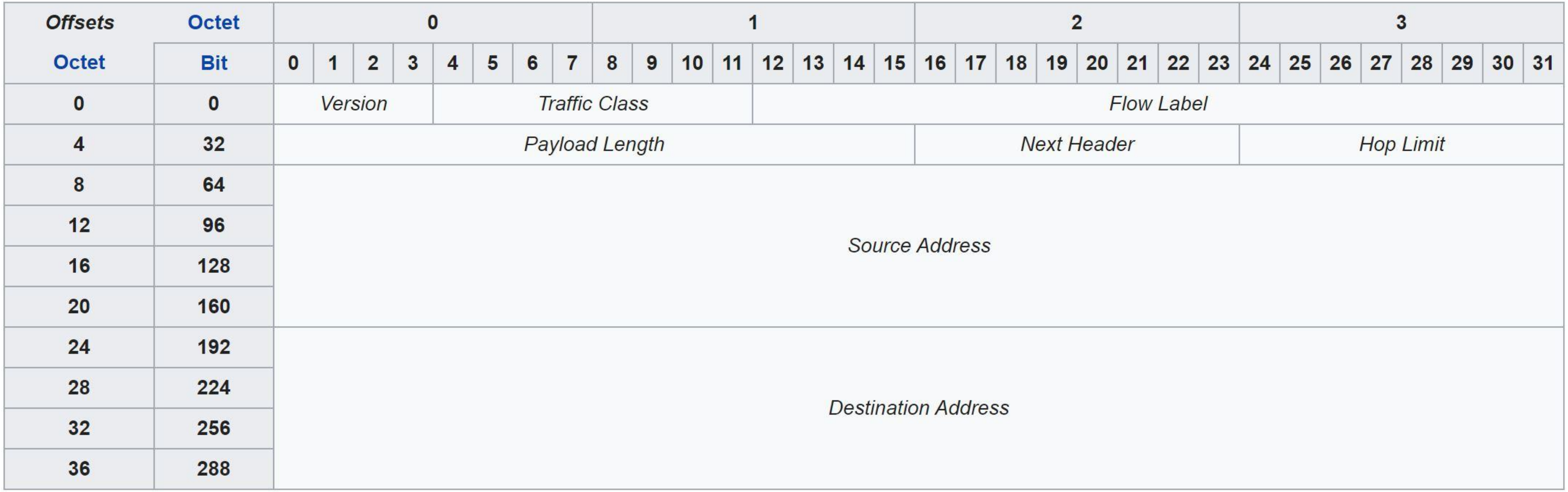
As jobs source data onto the network OR pull data into the job, we should try to ensure the corresponding packets are marked appropriately

- Containers and VMs may allow this to be easily put in place
- Still need configuration options that specify the right bits
- Signalling to the “source” about what those bits are also needs to be in place

# Packet Marking - IPv6

IPv6 incorporates a “Flow Label” in the header (20 bits)

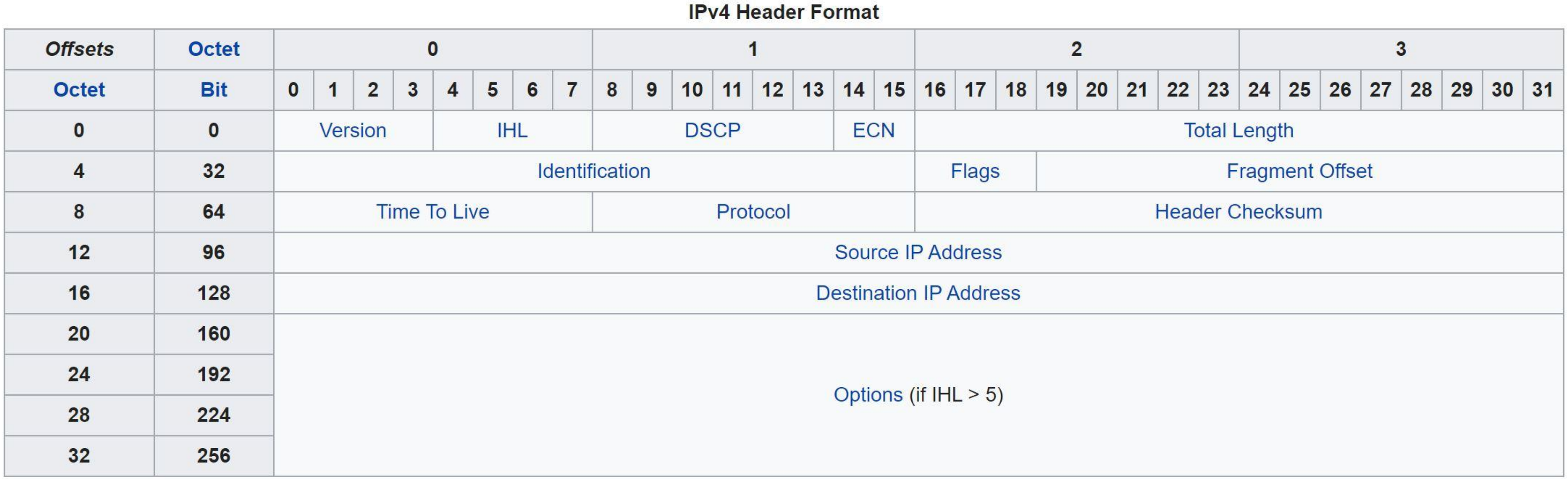
**Fixed header format**



| Offsets | Octet | 0 | | | | | | | | 1 | | | | | | | | 2 | | | | | | | | 3 | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Octet | Bit | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| 0 | 0 | *Version* | | | | *Traffic Class* | | | | | | | | *Flow Label* | | | | | | | | | | | | | | | | | | | |
| 4 | 32 | *Payload Length* | | | | | | | | | | | | | | | | *Next Header* | | | | | | | | *Hop Limit* | | | | | | | |
| 8 | 64 | *Source Address* | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 12 | 96 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 16 | 128 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 20 | 160 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 24 | 192 | *Destination Address* | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 28 | 224 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 32 | 256 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 36 | 288 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

# Packet Marking - IPv4

IPv4 incorporates a “Options” in the header (allowing to add more 32 bit words)



IPv4 Header Format

<table>
<tr><th>Offsets</th><th>Octet</th><th colspan="8">0</th><th colspan="8">1</th><th colspan="8">2</th><th colspan="8">3</th></tr>
<tr><th>Octet</th><th>Bit</th><th>0</th><th>1</th><th>2</th><th>3</th><th>4</th><th>5</th><th>6</th><th>7</th><th>8</th><th>9</th><th>10</th><th>11</th><th>12</th><th>13</th><th>14</th><th>15</th><th>16</th><th>17</th><th>18</th><th>19</th><th>20</th><th>21</th><th>22</th><th>23</th><th>24</th><th>25</th><th>26</th><th>27</th><th>28</th><th>29</th><th>30</th><th>31</th></tr>
<tr><td>0</td><td>0</td><td colspan="4">Version</td><td colspan="4">IHL</td><td colspan="6">DSCP</td><td colspan="2">ECN</td><td colspan="16">Total Length</td></tr>
<tr><td>4</td><td>32</td><td colspan="16">Identification</td><td colspan="3">Flags</td><td colspan="13">Fragment Offset</td></tr>
<tr><td>8</td><td>64</td><td colspan="8">Time To Live</td><td colspan="8">Protocol</td><td colspan="16">Header Checksum</td></tr>
<tr><td>12</td><td>96</td><td colspan="32">Source IP Address</td></tr>
<tr><td>16</td><td>128</td><td colspan="32">Destination IP Address</td></tr>
<tr><td>20</td><td>160</td><td colspan="32" rowspan="5">Options (if IHL > 5)</td></tr>
<tr><td>24</td><td>192</td></tr>
<tr><td>28</td><td>224</td></tr>
<tr><td>32</td><td>256</td></tr>
</table>