Packet Marking WG Update & Status

Shawn McKee, Marian Babik

on behalf of the RNTWG

HEPiX IPv6 Working Group - Virtual F2F Meeting

June 29, 2021

Motivation: Making our Network Use Visible

Understanding HEP 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. **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 our RENs (Research and Education Networks)**

- The proposed work is to identify how we might label our traffic at the **packet level** to indicate which **experiment** and **activity** it is a part of.
- The technical work encompasses how to **mark traffic** at the network level, defining a standard set of markings, **provide the tools** to the experiments to make it easy for them to participate and define how the NRENs can **monitor/account** for such data.

Review: RNTWG Workplan

- Based upon the interests of the experiments, sites and R&E networks, we are working to implement specific capabilities which can provide benefits as quickly as possible
- The experience learned during the monthly USATLAS, USCMS and ESnet Network Blueprinting meetings put the focus on marking our traffic
  - This seemed to be the low-hanging fruit and the one which would be easiest and quickest to deliver upon.
- The primary focus for RNTWG has been on packet marking.

Work Since the Last Presentation

- In January we presented on the **R**esearch **N**etwork **T**echnical **W**orking **G**roup (RNTWG) and its Packet Marking working group at the IPv6 WG meeting
- Meetings and presentations since then
  - RNTWG Packet Marking March 3
  - OSG All-hands, Network Discussion March 5
  - LHCONE meeting March 24
  - Storage Systems May 6
  - RNTWG Packet Marking June 2
  - Next RNTWG Packet Marking: Storage June 29 (in about 1.5 hours :))
- Today we will review our status and provide a summary

RNTWG Recent Meeting Summary

Our meetings this year have focused on possible implementation details for packet marking

- We converged on IPv6 as the only reasonable way to provide a location to put the marking into each packet
- The marking tools and technologies may vary
- The bit definitions are independent of where the bits are placed in packets or otherwise made accessible for R&E networks and end-users.
- The WLCG storage technology providers have been engaged in how we can implement the first steps for packet marking
- We have developed a “Flow and Packet Marking Technical Specification” to guide prototyping and implementation
- We need a way to show this effort is important and supported

Flow and Packet Marking Tech. Spec.

In the March working group meeting Marian Babik introduced a draft “Flow and Packet Marking Technical Specification” document to outline what the working group had identified as the relevant tools, technologies and methods for identifying our network traffic.

We have already had a focused storage provider meeting to discuss and evolve the draft document.

Today we will continue the discussion (but unfortunately it overlaps with the end of this meeting). See https://indico.cern.ch/event/1051825/

A possible target for early implementation is the upcoming WLCG Data Challenge

Fireflies (Out of band flow tagging)

One new option to identify traffic without actually tagging each packet was suggested by Stacey Sheldon and Yatish Kumar from ESnet: **Fireflies**

- The idea is to have the data source occasionally send a UDP packet with a specific format that will identify both the packet marking label along with the flow details.
- The packet will be sent to a specific port on the destination system and will therefore likely follow the same network path
- Anyone along the path could capture such packets to identify the details of the correlated flow

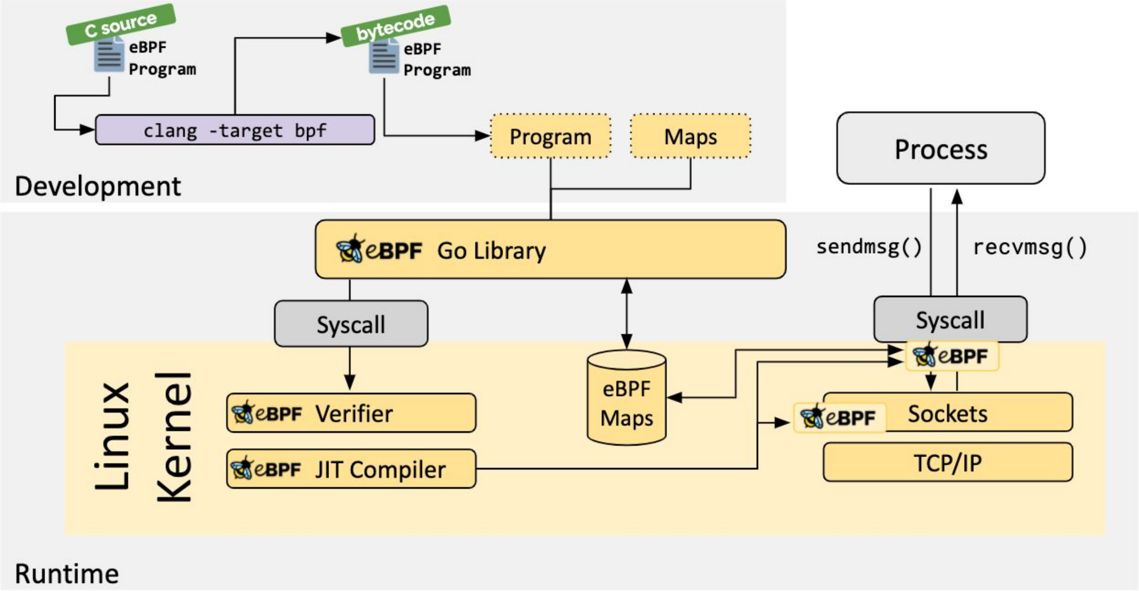
eBPF (New Marking Method)

Tristan Sullivan/U Vic suggested the use of eBPF as a possible marking mechanism. This may be a good technology to use to mark packets.

eBPF (extended Berkeley Packet Filter) is a project to allow linux kernel interactions without kernel models

Shown on the right is an example diagram using GO.   Bytecode can be generated to perform various manipulations which are executed by various Syscalls.

Could we use eBPF to support our packet marking use case?



Reminder: Packet Marking Challenges

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

- Required us to think broadly during design

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

- We have standardized bits (next few slides) and published
- Now we need to **maintain** it!!

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

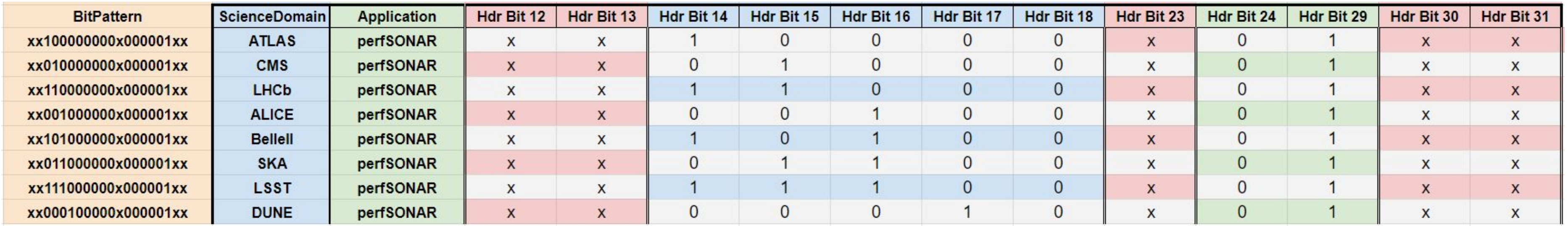
Are the bits easily consumed by hardware / software?

What can the network operators provide for accounting?

Reminder: Packet Marking Scheme

We combine two tables, one for Science Domain and one for Application, along with **5 entropy bits** to produce a master table of bit definitions for our 20 bits.

The spreadsheet Reference Table allows selection by bit patterns. The table below shows selecting on the “perfSONAR” Application type (**note** some columns are hidden), X = 0 or 1



BitPattern	ScienceDomain	Application	Hdr Bit 12	Hdr Bit 13	Hdr Bit 14	Hdr Bit 15	Hdr Bit 16	Hdr Bit 17	Hdr Bit 18	Hdr Bit 23	Hdr Bit 24	Hdr Bit 29	Hdr Bit 30	Hdr Bit 31
xx100000000x000001xx	ATLAS	perfSONAR	x	x	1	0	0	0	0	x	0	1	x	x
xx010000000x000001xx	CMS	perfSONAR	x	x	0	1	0	0	0	x	0	1	x	x
xx110000000x000001xx	LHCb	perfSONAR	x	x	1	1	0	0	0	x	0	1	x	x
xx001000000x000001xx	ALICE	perfSONAR	x	x	0	0	1	0	0	x	0	1	x	x
xx101000000x000001xx	BelleII	perfSONAR	x	x	1	0	1	0	0	x	0	1	x	x
xx011000000x000001xx	SKA	perfSONAR	x	x	0	1	1	0	0	x	0	1	x	x
xx111000000x000001xx	LSST	perfSONAR	x	x	1	1	1	0	0	x	0	1	x	x
xx000100000x000001xx	DUNE	perfSONAR	x	x	0	0	0	1	0	x	0	1	x	x

Example perfSONAR Packet Marking

- The first application used to test flow-label marking was perfSONAR Iperf3
  - **PWA** was able to centrally configure `--flow-label` for IPv6 **Iperf3** tests
  - Labels were manually verified via `tcpdump` at the destination
  - We now set a flow label on one perfSONAR test mesh (**US ATLAS**: CERN, AGLT2, MWT2, BNL, BU, LUNET, NERSC and Stanford; the flow label (65540) is set on iperf3 tests for this mesh.)
- Tim Chown has started an engagement with the **perfSONAR** developers, bringing in IPv6 expert Fernando Gant
  - Fernando and Mark Feit are discussing creating a new tool/test which sets a flow-label in the packet header and sends the same label as the data, then verifies they match (or not) at the destination? **A new path6 tool exists but still needs integration in the toolkit**
- perfSONAR, as an extensible framework, should be a good tool to use for the Packet Marking work
  - Can we get all standard perfSONAR tools to support a centrally defined `--flow-label` option? (traceroute already supports it but not in **PWA**)

Packet Marking for Storage Elements

The bulk of WLCG traffic is generated by our storage elements.

The primary challenge here is in two areas:

1. Augmenting the existing storage system to be able to set the label in the network packets and/or to emit “Fireflies”
2. Communicating the label as part of a transfer request
    a. Likely need some protocol extension to support this
    b. We have a document from the Xrootd developers discussing this.
    c. Additional document for IPv6 HTTP-TPC as well

Each storage technology uses different programming languages or libraries making it difficult to have a single implementation.

Creating an End-Host Service

Because of the challenges involved in having each storage technology directly interact with its network sockets (very hard for Java for example), we are considering developing a **service with a clear, simple API** that would be installed on storage hosts

This service would be responsible for ensuring both that appropriate packets are marked and would handle sending Fireflies

Likely utilize **tc** or **eBPF** to do its work. *Under discussion...*

How Do We Show this is Important Work?

One of the interesting questions that has come out during our working group meetings is how important is this work to the experiments, the R&E networks and end-users?

This is a valid concern, especially when work on packet marking may preclude other important work from happening

We have pointed to the January 2020 LHCONE/LHCOPN meeting with the experiments and follow on meetings as the rationale for this effort.

To make this more concrete, we are planning to create a project website and ask for logos from major organizations indicating support and backing for the work.

Packet Marking for Jobs

We would eventually like to account for traffic generated by production and user 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 help this to be more easily put in place
- Jobs will need configuration options that specify the right bits
- Signaling to the “source” about the label also needs to be in place
- If the End-Host marking service exists, could just deploy on worker nodes?

RNTWG High Level Notes

We continue to be guided by what is useful but also feasible and possible.

Marking and shaping/pacing **must happen on the source** means there is an opportunity to extend marking service to enable shaping.

Longer term work on shaping and orchestration will be informed by what we create for **packet marking** our traffic.

Orchestration is much more feasible once marking is in place

Current Plans and Schedule

- Focus on interacting with the WLCG storage technology providers, as well as Rucio, FTS and the R&E networks
- **Prototype a host service that can implement the marking based upon call-out from the storage service**
  - **See section 4 of the Technical Specification for details**
- Consuming / Utilizing the bits (Start work when enough traffic has marks)
  - Work with R&E networks and sites to capture and measure the marked traffic
  - Verify traffic markings consistently pass end-to-end
  - Differentiate intentionally marked traffic vs standard flow-label use
- Testing in our R&E networks (ASAP), hopefully in time for the first WLCG Data Challenge at the end of September.

Questions, Comments, Suggestions?

We are working on IP packet and flow marking which has been identified as an important capability WLCG and R&E networks.

From this group's perspective, one important aspect is that there is now **another good reason to implement IPv6!**

Want to be involved? We could use effort on:

Helping provide tools to enable marking.

Testing tools/options for marking.

Creating or testing "consuming" the marked bits inside the network or at the ends.

Questions, Comments, Suggestions?

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

References

Packet marking document

Research Networking Technical WG Google folder

RNTWG Wiki

RNTWG mailing list signup

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

Flow and Packet Marking Technical Specification

NFV WG Report

SDN/NFV Tutorial: https://indico.cern.ch/event/715631/

2018 IEEE/ACM Innovating the Network for Data-Intensive Science (INDIS) – http://conferences.computer.org/scw/2018/#!/toc/3

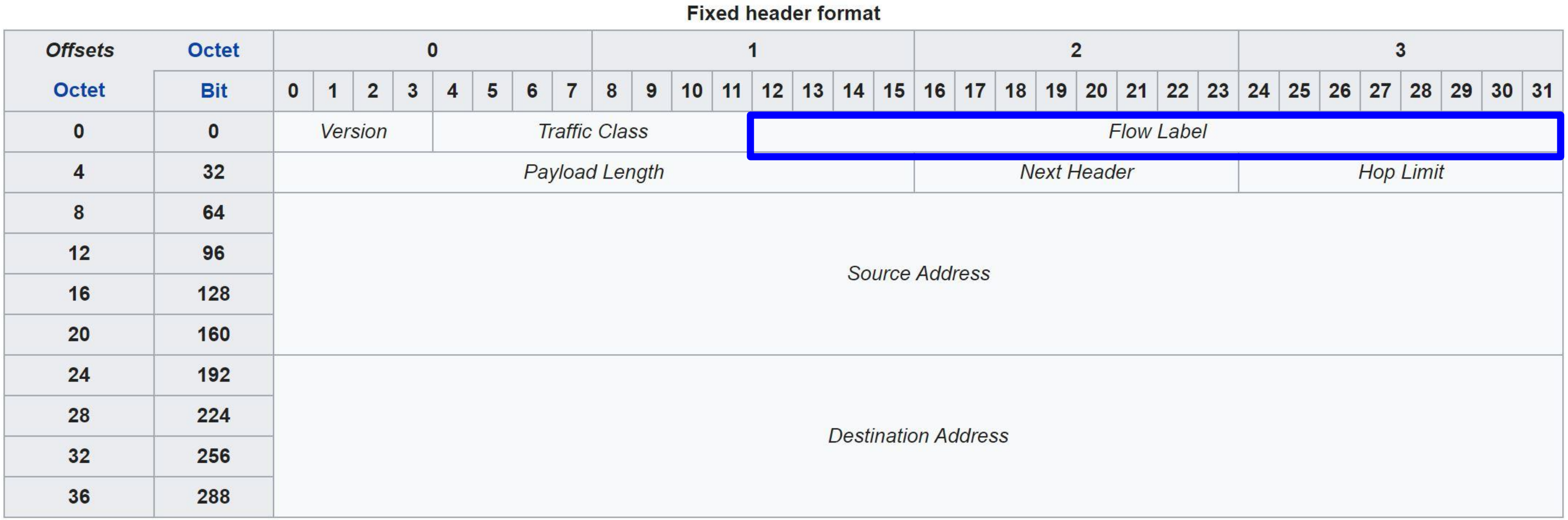
OVN/OVS overview: https://www.openvswitch.org/

Backup slides

Packet Marking - IPv6 Flow Label

The group is focusing on IPv6 and use of the flow-label

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																																

Review: Packet Marking Scheme

The draft packet marking scheme is in a Google sheet.

We started with **20 bits** (matching the size of the flow-label)

- We add 5 entropy bits to try to match the spirit of RFC6436
- We use **9 bits** to define the **Science Domain** (reserving 3 for non-Astro/HEP domains)
- We use **6 bits** to define the **Application/Type** of traffic
- We organize the bits to allow for potential adjustments in the future.

The next few slides detail what we have arrived at

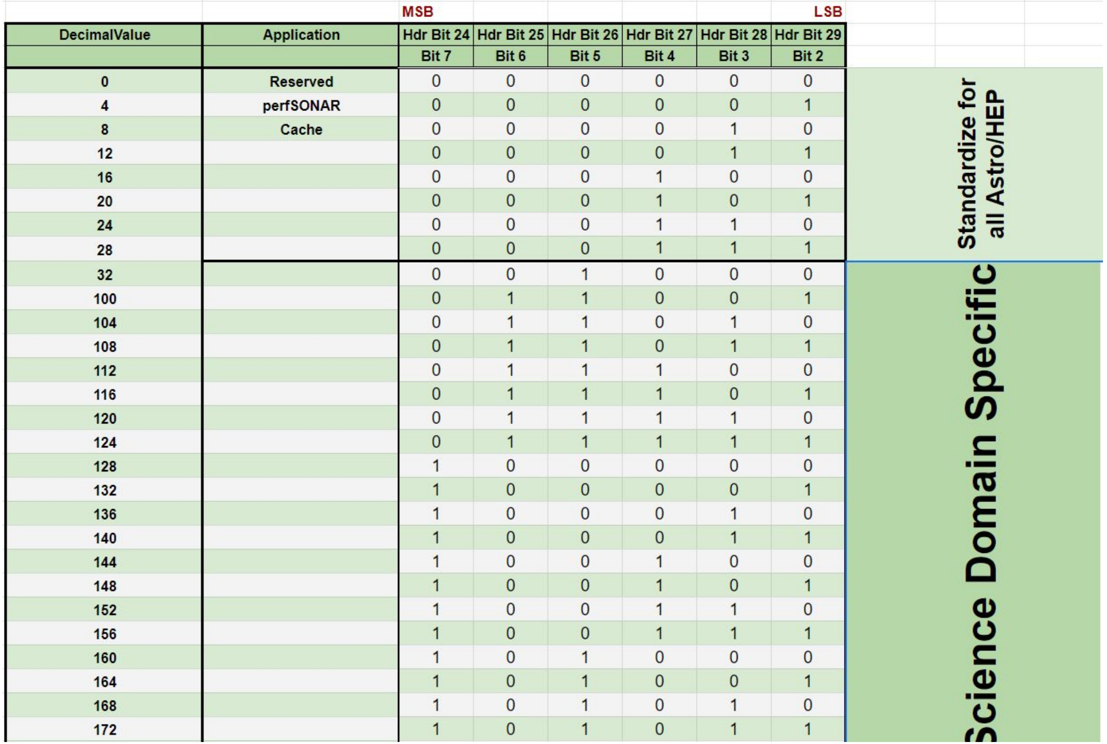
Application Marking Scheme

The 6 bits for Application are divided into two types: common across Science Domain (3 MSB = 0) and Science Domain specific

Note: some rows are hidden

We show the “**decimal value**” of the specific applications, assuming all the entropy bits are zero.

This makes it easy to add application+domain+entropy value to determine the final flow-label.

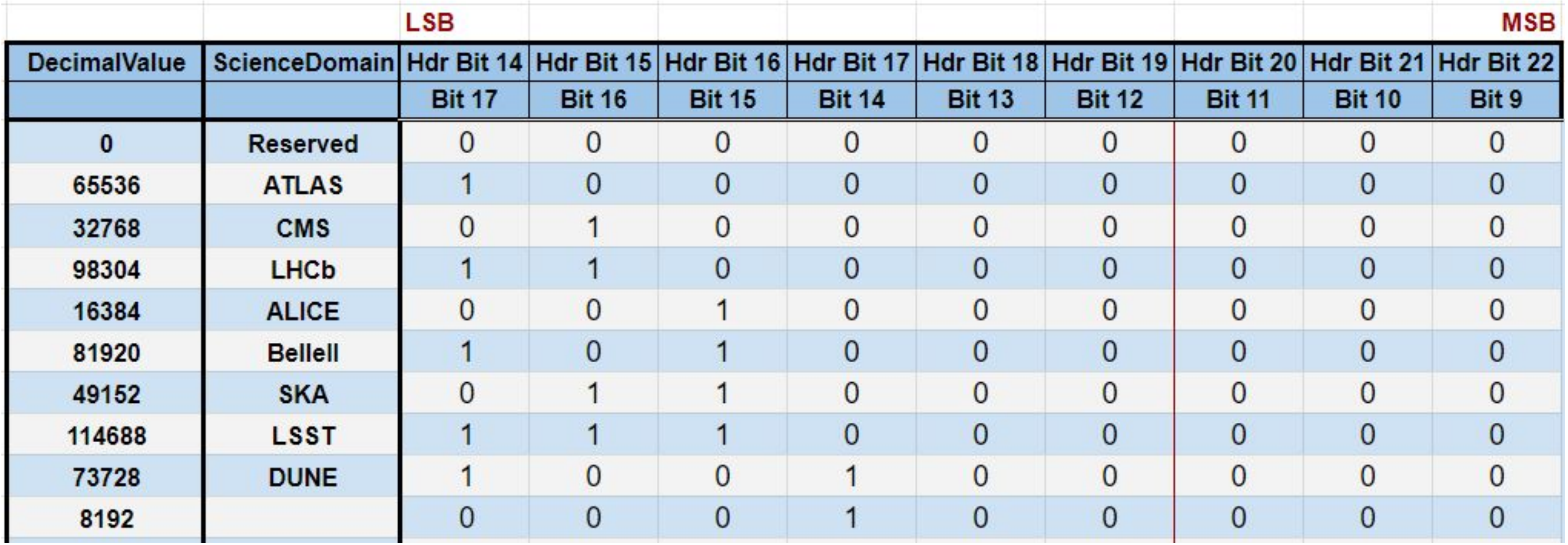


		MSB					LSB	
DecimalValue	Application	Hdr Bit 24	Hdr Bit 25	Hdr Bit 26	Hdr Bit 27	Hdr Bit 28	Hdr Bit 29	
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	
0	Reserved	0	0	0	0	0	0	Standardize for all Astro/HEP
4	perfSONAR	0	0	0	0	0	1	
8	Cache	0	0	0	0	1	0	
12		0	0	0	0	1	1	
16		0	0	0	1	0	0	
20		0	0	0	1	0	1	
24		0	0	0	1	1	0	
28		0	0	0	1	1	1	
32		0	0	1	0	0	0	Science Domain Specific
100		0	1	1	0	0	1	
104		0	1	1	0	1	0	
108		0	1	1	0	1	1	
112		0	1	1	1	0	0	
116		0	1	1	1	0	1	
120		0	1	1	1	1	0	
124		0	1	1	1	1	1	
128		1	0	0	0	0	0	
132		1	0	0	0	0	1	
136		1	0	0	0	1	0	
140		1	0	0	0	1	1	
144		1	0	0	1	0	0	
148		1	0	0	1	0	1	
152		1	0	0	1	1	0	
156		1	0	0	1	1	1	
160		1	0	1	0	0	0	
164		1	0	1	0	0	1	
168		1	0	1	0	1	0	
172		1	0	1	0	1	1	

Science Domain Marking

The 9 bits assigned for Science Domain are in reverse bit-order

to keep the currently reserved (non-Astro/HEP) bits closest to the entropy bit, in case we need to adjust later. (Bits 11-9 != 0 are Non-Astro/HEP)



		LSB								MSB
DecimalValue	ScienceDomain	Hdr Bit 14	Hdr Bit 15	Hdr Bit 16	Hdr Bit 17	Hdr Bit 18	Hdr Bit 19	Hdr Bit 20	Hdr Bit 21	Hdr Bit 22
		Bit 17	Bit 16	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9
0	Reserved	0	0	0	0	0	0	0	0	0
65536	ATLAS	1	0	0	0	0	0	0	0	0
32768	CMS	0	1	0	0	0	0	0	0	0
98304	LHCb	1	1	0	0	0	0	0	0	0
16384	ALICE	0	0	1	0	0	0	0	0	0
81920	BelleII	1	0	1	0	0	0	0	0	0
49152	SKA	0	1	1	0	0	0	0	0	0
114688	LSST	1	1	1	0	0	0	0	0	0
73728	DUNE	1	0	0	1	0	0	0	0	0
8192		0	0	0	1	0	0	0	0	0

Introduction / Overview

The LHCOPN/LHCONE meeting at CERN a year ago, brought in the LHC/HEP experiments who described their networking needs, interests and use-cases.

The experiments reinforced what the HEPiX NFV phase I report suggested were useful areas to focus effort upon:

- Making our network use visible (Packet Marking)
- Shaping WAN data flows (Traffic Shaping)
- Orchestrating the network (Network Orchestration)

In response we formed the Research Networking Technical Working group with three sub-groups focused on the above areas.

Today we are providing an update on our activities and plans focused primarily on the Packet Marking effort.

Research Networking Technical WG

Charter:

https://docs.google.com/document/d/1I4U5dpH556kCnoIHzyRpBI74IPc0gpgAG3VPUp98Io0/edit#

Mailing list:

http://cern.ch/simba3/SelfSubscription.aspx?groupName=net-wg

Members (90 as of today, in no particular order):

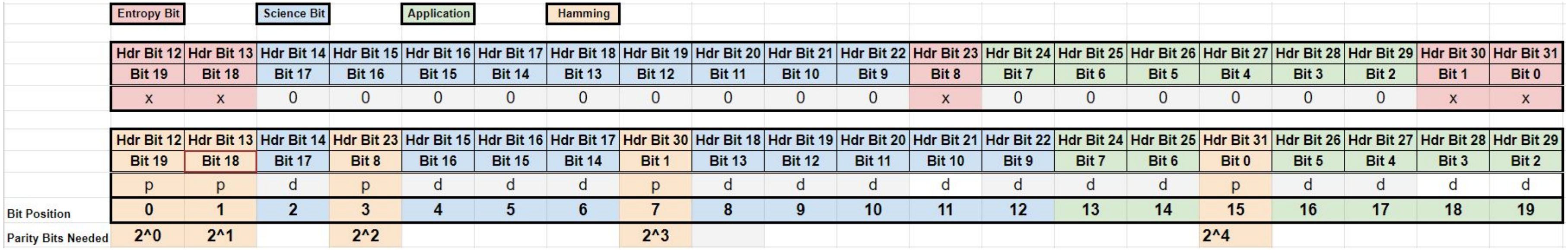
Christian Todorov (Internet2) Frank Burstein (BNL) Richard Carlson (DOE) Marcos Schwarz (RNP) Susanne Naegele Jackson (FAU)
Alexander Germain (OHSU) Casey Russell (CANREN) Chris Robb (GlobalNOC/IU) Dale Carder (ESnet) Doug Southworth (IU)
Eli Dart (ESNet) Eric Brown (VT) Evgeniy Kuznetsov (JINR) Ezra Kissel (ESnet) Fatema Bannat Wala (LBL) Joseph Breen (UTAH)
James Blessing (Jisc) James Deaton (Great Plains Network) Jason Lomonaco (Internet2) Jerome Bernier (IN2P3) Jerry Sobieski
Ji Li (BNL) Joel Mambretti (Northwestern) Karl Newell (Internet2) Li Wang (IHEP) Mariam Kiran (ESnet) Mark Lukasczyk (BNL)
Matt Zekauskas (Internet2) Michal Hazlinsky (Cesnet) Mingshan Xia (IHEP) Paul Acosta (MIT) Paul Howell (Internet2)
Paul Ruth (RENCI) Pieter de Boer (SURFnet) Roman Lapacz (PSNC) Sri N () Stefano Zani (CNAF) Tamer Nadeem (VCU)
Tim Chown (Jisc) Tom Lehman (ESnet) Vincenzo Capone (GEANT) Wenji Wu (FNAL) Xi Yang (ESnet) Chin Guok (ESnet)
Tony Cass (CERN) Eric Lancon (BNL) James Letts (UCSD) Harvey Newman (Caltech) Duncan Rand (Jisc)
Edoardo Martelli (CERN) Shawn McKee (Univ. of Michigan) Simone Campana (CERN) Andrew Hanushevsky (SLAC)
Marian Babik (CERN) James William Walder () Petr Vokac () Alexandr Zaytsev (BNL) Raul Cardoso Lopes () Mario Lassnig (CERN)
Han-Wei Yen () Wei Yang (Stanford) Edward Karavakis (CERN) Tristan Suerink (Nikhef) Garhan Attebury (UNL) Pavlo Svirin ()
Shan Zeng (IHEP) Jin Kim (KISTI) Richard Cziva (ESnet) Phil Demar (FNAL) Justas Balcas (Caltech) Bruno Hoeft (FZK)

Packet Marking Validity Option

One concern expressed during our discussions was “pollution” of our results from packets that use the flow-label to provide entropy.

We can minimize this by calculating a Hamming code, using our 5 entropy bits to create parity bits. This maximizes the distance (bit-wise) between valid flow-labels for our marking use-case/

The table below shows how to rearrange the bits for this:



Entropy Bit | Science Bit | Application | Hamming

	Hdr Bit 12	Hdr Bit 13	Hdr Bit 14	Hdr Bit 15	Hdr Bit 16	Hdr Bit 17	Hdr Bit 18	Hdr Bit 19	Hdr Bit 20	Hdr Bit 21	Hdr Bit 22	Hdr Bit 23	Hdr Bit 24	Hdr Bit 25	Hdr Bit 26	Hdr Bit 27	Hdr Bit 28	Hdr Bit 29	Hdr Bit 30	Hdr Bit 31
	Bit 19	Bit 18	Bit 17	Bit 16	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
	x	x	0	0	0	0	0	0	0	0	0	x	0	0	0	0	0	0	x	x

	Hdr Bit 12	Hdr Bit 13	Hdr Bit 14	Hdr Bit 23	Hdr Bit 15	Hdr Bit 16	Hdr Bit 17	Hdr Bit 30	Hdr Bit 18	Hdr Bit 19	Hdr Bit 20	Hdr Bit 21	Hdr Bit 22	Hdr Bit 24	Hdr Bit 25	Hdr Bit 31	Hdr Bit 26	Hdr Bit 27	Hdr Bit 28	Hdr Bit 29
	Bit 19	Bit 18	Bit 17	Bit 8	Bit 16	Bit 15	Bit 14	Bit 1	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 7	Bit 6	Bit 0	Bit 5	Bit 4	Bit 3	Bit 2
	p	p	d	p	d	d	d	p	d	d	d	d	d	d	d	p	d	d	d	d
Bit Position	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Parity Bits Needed	2^0	2^1		2^2				2^3								2^4				