Update from RNTWG - Packet Marking Subgroup

Shawn McKee, Marian Babik

on behalf of the RNTWG

HEPiX IPv6 Working Group - Virtual F2F Meeting

September 29, 2020

Introduction / Overview

The LHCOPN/LHCONE meeting at CERN in January, 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)

Review: 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 experiment's planning is including networking to a much greater degree than before
  - HL-LHC computing review: DOMA, dedicated networking section.
  - ESnet Planning and Case Studies: detailing operations, needs, use-case and future plans.
  - Broad realization that network challenges are going to be critical to prepare for HL-LHC
- **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

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.
- We started with a Kickoff meeting for the whole RNTWG in April and then moved to Packet Marking sub-group mtgs.

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.

RNTWG Meetings Since Spring

- 21 Apr - Kickoff meeting - presented charter
- 04 June - Created draft documents and shared Drive
  - Started working on packet marking; had a long discussion on it during the meeting
  - Agreed that forwarding decisions and policing bits is out of scope for this work
  - **Decided: we focus on IPv6 and if possible backport to IPv4**
  - Initiated discussion on possible approaches in IPv6 packet marking
    - Flow labels; Extension headers; IPv6 addressing
- 30 June - More in-depth discussion related to IPv6 packet marking
  - Looked at Linux kernel IPv6 implementation status
    - Agreed to go ahead with IPv6 labels and come up with concrete proposal (and shim prototype) as well as to look further at the other (IPv6 marking) options to better understand the status of their implementation and how they would match our use cases
    - We briefly discussed how/where to capture activities and how to make them available
- Two days ago we had another Packet Marking sub-group meeting

Packet Marking Meeting - 14 Sep 20

The meeting focus was on the Draft Packet Marking Bit Definition (more on this later)

- We proceed trying to use the **IPv6 flow label** (20bits, IPv6 header field)
- Proposal is to use 9 bits for science domain and 6 bits for activity, leaving 5 bits for flow entropy and/or consistency

While there are other options we will continue to explore (Using an **IPv6 Hop-by-Hop**, **Destination Option**, **Using IPv6 addressing**) we have chosen the **Flow-Label** to make quick progress because

- It is supported in the standard linux kernels (CentOS7+) via setsockopt calls.
- Network devices and flow monitoring tools support extracting it in most cases

For now we are proceeding with the source of truth being a Google spreadsheet but we may want to consider developing a service to maintain and provide access to the label definitions.

- This is true even if the marking changes from using Flow-Label or changes size

More details are in the notes available at:
https://docs.google.com/document/d/1yPYiI-dflyc00sbzWqjTYCrwFRDd5VFrugIcocbJGA0/edit#

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?

- Need to standardize bits and publish and **maintain**!!

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?

Packet Marking - IPv6 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																																

Draft Packet Marking Scheme

We have drafted an initial packet marking scheme 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

Packet Marking Scheme

We can combine the previous two tables for Science Domain and Application, along with **5 entropy bits** to produce the 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

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				

Current Plans and Schedule

- For now, focus on **IPv6 Flow Label** option
- Initial bit use definitions: **DONE**
  - Next steps: engage with the science domains to flesh out common and domain specific application markings
- **Applications** - We need to enable packet marking in as many HEP applications as possible
  - We are targeting: perfSONAR, XRootD
    - We have initial xrootd plan, describing the work needed
    - Eventually we need to engage with FTS, Rucio, dCache, STORM, HTTP (WebDav) and others
- Consuming / Utilizing the bits
  - Work with R&E networks and sites to try 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

perfSONAR Enhancements

- The first application we used to test flow-label marking was perfSONAR Iperf3 traffic
  - **PWA** was able to centrally configure `--flow-label` for IPv6 **Iperf3** tests
  - Labels were manually verified via `tcpdump` at the destination
- 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?
- 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**)

Questions, Comments, Suggestions?

We have identified packet marking as important for WLCG

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

Want to be involved?

We really need a broad range of expertise involved: network programming, standardization experience, experiment software expertise, storage software expertise, NRENs, documentation experience, monitoring, accounting, etc.

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/
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/
GEANT Automation, Orchestration and Virtualisation (link)
Cloud Native Data Centre Networking (book)
MPLS in the SDN Era (book)

Backup slides

Pacing/Shaping WAN data flows

It remains a challenge for HEP storage endpoints to utilize the network efficiently and fully.

- An area of potential interest to the experiments is traffic shaping/pacing.
  - Without traffic 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

- OpenStack and Kubernetes are being leveraged to create very dynamic infrastructures to meet a range of needs.
  - Critical for these technologies is a level of automation for the required networking using both software defined networking and network function virtualization.
  - For HL-LHC, important to find tools, technologies and improved workflows that may help bridge the anticipated gap between the resources we can afford and what will actually be required
- The ways in which we may organize our computing and storage resources will need to evolve.
- Data Lakes, federated or distributed Kubernetes and multi-site resource orchestration will certainly benefit (or require) some level of WAN network orchestration to be effective.
  - We would suggest a sequence of limited scope proof-of-principle activities in this area would be beneficial for all our stakeholders.

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 - 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?

High Level Notes

What is useful? Feasible? Possible?

The idea of marking, shaping and orchestration are steps in order of assumed difficulty and time-to-implement

Marking and shaping/pacing **must happen on the source**

Orchestration is much more feasible once marking is in place