Research Networking Technical WG Update

Marian Babik, Shawn McKee

HEPiX, March 2021

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

Research Networking Technical WG

- 95 members from ~ 50 organisations
- Making our network use visible (**marking**)
  - 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.
  - The proposed work here is to identify how we might label our traffic at the **packet** level to indicate which **experiment** and **activity** it is a part of.
- Shaping WAN data flows (**pacing**)
  - It remains a challenge for HEP storage endpoints to utilize the network efficiently and fully. Mainly focused on pacing the flows to match link capacity and avoid microburst problem; but also looking into new congestion control algorithms.
- **Orchestrating** the network to enable multi-site infrastructures
  - 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.

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.

Packet Marking Challenges

Aim at ALL significant R&E network users/science domains, not just HEP

- Required us to think broadly during design

Which IP fields we can use for marking ? How best to use the number of bits we can get?

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

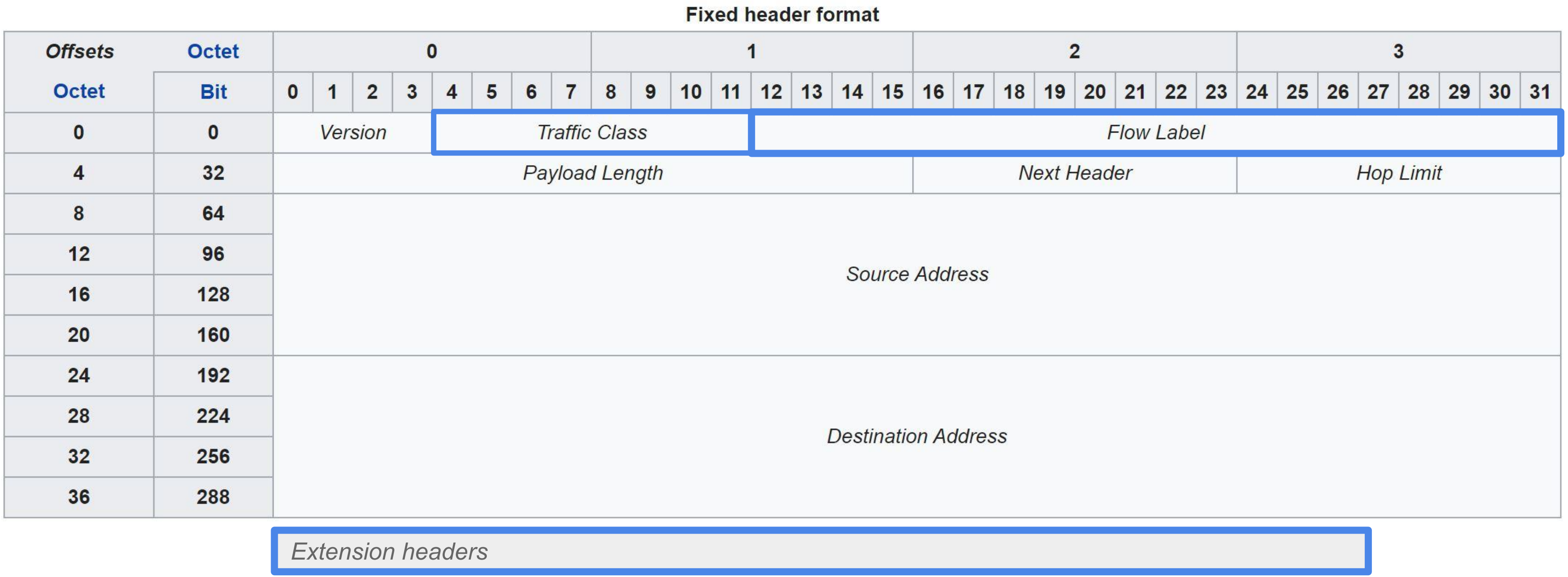
What technologies can we use in the Linux network stack ??

Can the bits easily consumed by hardware / software?

What can the network operators provide for accounting?

Packet Marking - IPv6

IPv6 candidates



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																																

Extension headers

For more details and discussion of various trade-offs please refer to the [Packet Marking Document]

Draft Packet Marking Scheme

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

An initial packet marking scheme is in a Google sheet.

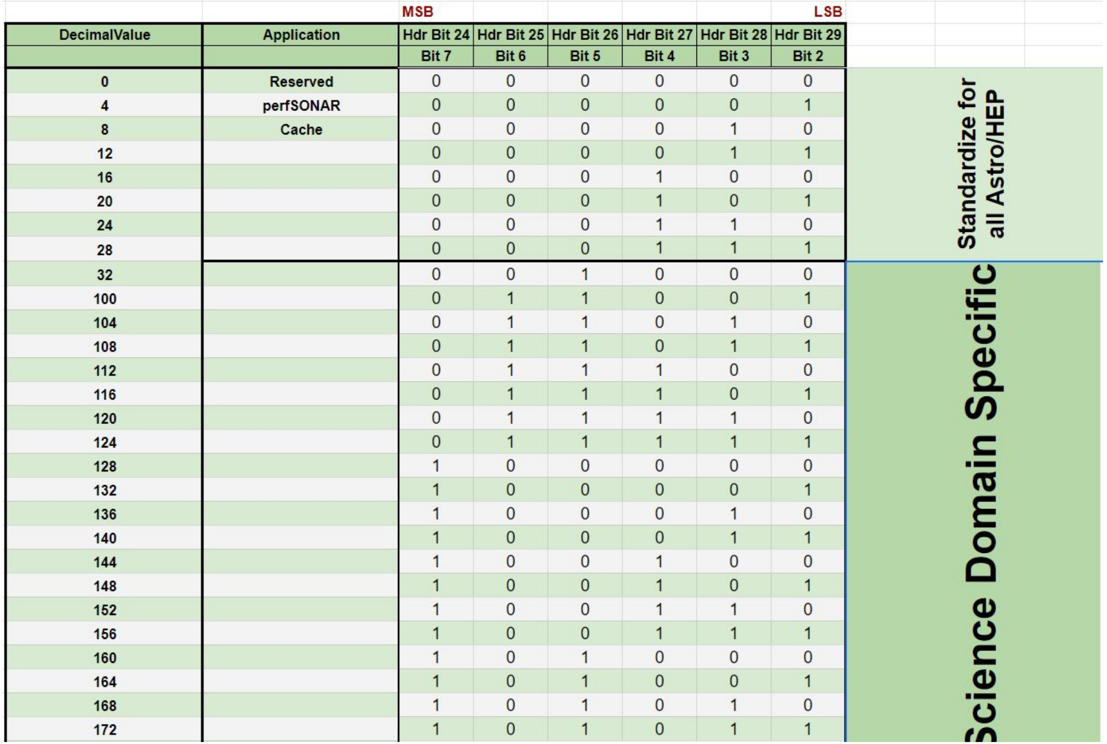
Application Marking Scheme

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.

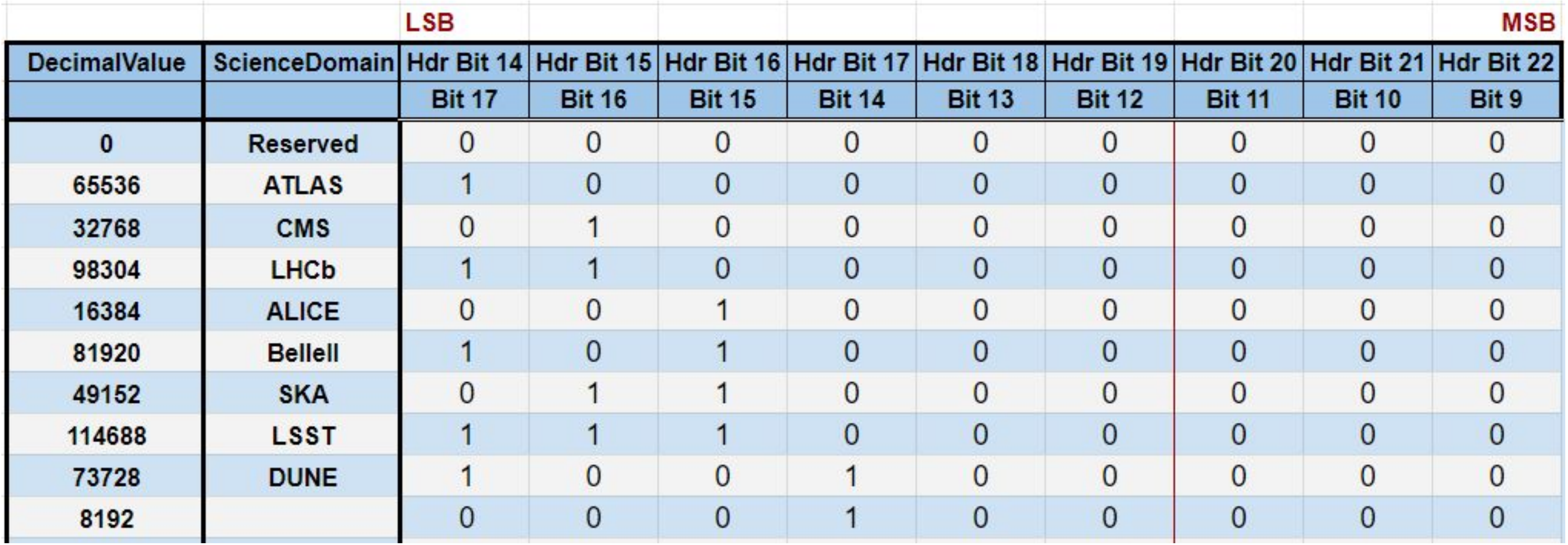


		MSB					LSB	
DecimalValue	Application	Hdr Bit 24	Hdr Bit 25	Hdr Bit 26	Hdr Bit 27	Hdr Bit 28	Hdr Bit 29	
		Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	
0	Reserved	0	0	0	0	0	0	Standardize for all Astro/HEP
4	perfSONAR	0	0	0	0	0	1	
8	Cache	0	0	0	0	1	0	
12		0	0	0	0	1	1	
16		0	0	0	1	0	0	
20		0	0	0	1	0	1	
24		0	0	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)

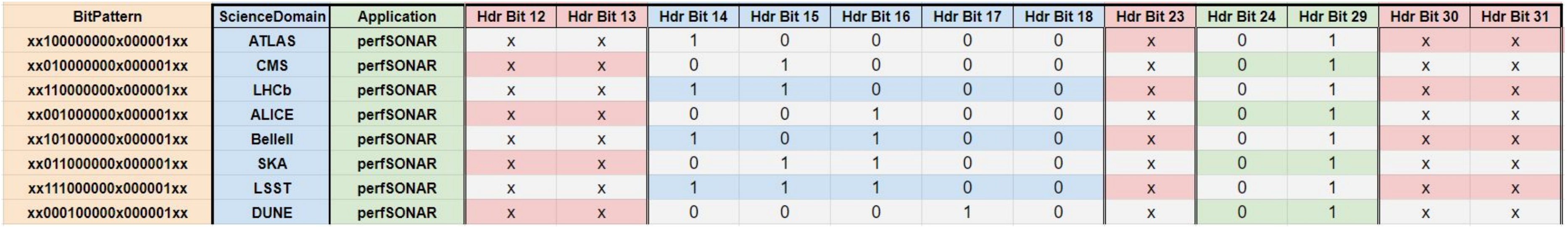


		LSB								MSB
DecimalValue	ScienceDomain	Hdr Bit 14	Hdr Bit 15	Hdr Bit 16	Hdr Bit 17	Hdr Bit 18	Hdr Bit 19	Hdr Bit 20	Hdr Bit 21	Hdr Bit 22
		Bit 17	Bit 16	Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9
0	Reserved	0	0	0	0	0	0	0	0	0
65536	ATLAS	1	0	0	0	0	0	0	0	0
32768	CMS	0	1	0	0	0	0	0	0	0
98304	LHCb	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

Summary: 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

Flow Label Implementation

- For now, focus on **IPv6 Flow Label** option and its implementation

- **Testing** flow label reachability in our R&E networks
- **Applications** - We need to enable packet marking in as many HEP applications as possible (storage/transfers systems/jobs):
  - Worked on creating prototypes to test the packet marking
  - Started to discuss possible implementation with storage providers
    - We have initial xrootd plan, describing the work needed
    - Eventually we want to engage with FTS, Rucio, dCache, STORM, HTTP (WebDav), GFAL2, and others

- **Networks** - Consuming / Utilizing the bits
  - Work with R&E networks and sites to try to capture and measure the marked traffic
  - Verify impact on traffic both WAN and LAN
  - Differentiate intentionally marked traffic vs standard flow-label use

Testing Flow Labels

- The first application used to test flow-label marking was perfSONAR Iperf3
  - We can 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.)

- IPv6 testing toolkit was enhanced by Fernando Gont to track flow labels
  - Developed as part of an effort to track IPv6 extension headers filtering
  - path6 - traceroute tool with full support of IPv6 extension headers
    - Following our request, the capability to track a particular flow label was introduced recently (howto)
  - Tim Chown has started an engagement with the **perfSONAR** developers to integrate the tool. This will provide an alternative to iperf3 which we can deploy

Flow Labels in Linux

Our primary aim was to investigate Linux kernel native API. It allows fine-grained control over how flow labels are assigned and used. We have developed as set of examples how to enable, set and clear flow labels as well as how to read both local and remote labels.

Flow labels reflection is a feature that enables TCP server side support without any further implementation. UDP works as expected using extended recvfrom/sendto calls.

Currently only tested on the latest kernel version, further tests are needed to better understand support across different versions as well as impact of existing optimisations used in storage/transfer systems.

- Tests across WAN and other existing network middleware (proxies, etc.)
- Tests different kernel versions/distribution support (C8)

XDP/eBPF/BPF-TC

eBPF - technology that can run sandboxed programs in the Linux kernel - programs can be connected in the kernel space via various hooks (connection tracking, traffic control, etc.)

XDP uses eBPF for fast packet filtering and forwarding. It executes directly in the receive handler of the driver (driver hook/space; before packet gets to Linux network stack; before socket buffer metadata is created).

- **High-performance packet filtering, forwarding and manipulation (in place) is possible**
- **Can run as an independent service, tagging packets for all services**
  - Storage/transfer systems would only need to pass information on e.g. dst_addr/dst_port/tag
- **User space to kernel space communication can be used to configure the tagging**
  - Service can expose high level API to interact with
- **Easy integration with network namespaces, containers, etc.**

BPF-TC is another hook in traffic control (tc), which can work on egress but operates on socket buffers (TBD), which likely impacts its performance.

Current Plans and Schedule

- Continue to focus on **IPv6 Flow Label** option
- **Testing** in our R&E networks
  - Integration of path6 in perfSONAR to enable reachability testing
- **Applications** - enable packet marking in as many HEP applications as possible:
  - Improve existing examples for how to mark packets
    - Code examples for native Linux API now available
    - Working on initial code exercises with eBPF/XDP
    - Additional tests planned to understand coverage, limitations and support
  - We are currently targeting: perfSONAR, XRootD ASAP
    - We have initial xrootd plan, describing the work needed
    - Need broader engagement: FTS, Rucio, dCache, STORM, HTTP, etc.
- **Networks** - 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

Questions, Comments, Suggestions?

We have identified packet marking as important for WLCG and that work and the RNTWG parent group are a new focus area for the HEPiX working group.

We are interested in expanding membership.

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?

HEPiX NFV WG

- Research Networking Technical WG was formed to follow up on HEPiX NFV WG area on programmable networks
  - WG has a wider scope and involves not only sites, but also experiments, R&Es, storage and transfer systems and has wider scope than NFV WG and will therefore **replace it**
- The former HEPiX NFV WG had a second area on cloud native networking
  - We haven’t received any feedback on this since the last HEPiX and would therefore propose to merge this into the **Technology Watch WG**

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

Code : https://github.com/marian-babik/ipv6_flow_label

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)

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

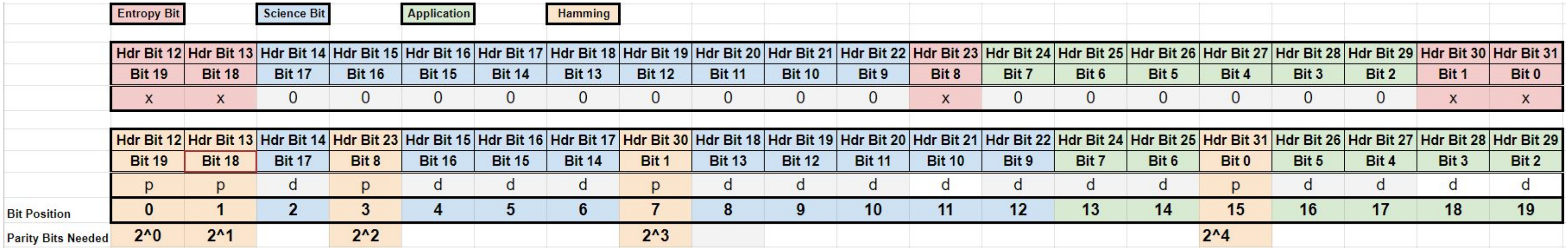
More details are in the notes available at:
https://docs.google.com/document/d/1yPYiI-dflyc00sbzWqjTYCrwFRDd5VFrugIcocbJGA0/edit#

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				

Bare Metal Switches + Network OS

Implement core switching/routing functions directly in Linux kernel and create an open source network OS that can run on bare metal switches.

This approach is followed by **Cumulus** and others
 - sometimes with their own equipment (using merchant packet switching silicon)

Free Range Routing (FRR) - IP routing suite for Linux that supports range of routing protocols BGP, OSPF, IS-IS (Linux Foundation project, formerly Quagga).

As this is a platform, different approaches are possible. One particularly interesting approach is to run **eBGP as the only control plane in DC** and use **EVPN/VXLAN** to integrate with compute.

Software switches

An alternative to bare metal/NOS is to have generic **software switch** running directly on the hypervisors/servers

- Good examples of this would be **OVS** and **Tungsten Fabric**

Tungsten/Contrail is a platform to build multi-stack networking DC:

- It has its own software switch running on the servers (vrouter), which uses EVPN/VXLAN to connect them*
- Supports different tunneling protocols (including MPLS)
- Native integration with physical equipment (via BGP or even netconf)

Supports both multi-stack and across-stack (OpenStack, VMware, K8s)
Using BGP/MPLS internally means it's easy to extend network to other DCs

*For comparison btw OVS and vrouter see OVS talk by Y. Yang

Network Disaggregation

- Breaks switch/router into hardware and software that can be purchased separately
  - Could have similar impact as server disaggregation had in the past
- This trend has started and is picking up pace due to the following reasons:
  - Clos topology requires small form factor switches with basic features in large quantities
  - Rise of the merchant packet switching silicon (now at the core in most switches)
  - Advances made in the manufacturing of the bare metal switches
  - Technical limitations/cost reduction in building up cloud-native DC with traditional equip.
- This had also significant impact on the progress of the open-source Network Operating Systems (NOS)
  - And in turn on developments in the Linux kernel networking stack and its extensions
- Created pressure for transition into **open environments** (ONIE, OCP, etc.)

Linux Foundation Networking

Additional projects that improve performance, provide alternative controllers, offer programmable off-loading capabilities, etc. are hosted by Linux Foundation

Intel's Data Plane Development Kit (DPDK) - accelerates packet processing workloads running on a wide variety of CPU architectures

P4 - programming language for packet processing - suitable for describing everything from high-performance forwarding ASICs to software switches.

DC Edge

- HEP sites usually require significant north-south bandwidth (btw. DCs)
  - Understanding how to effectively design the DC edge is therefore critical

- DC Edge is going to become very important for number of reasons
  - Advances and affordability of the **Data Centre Interconnect (DCI)** technologies - this includes both hardware-based and software-based approaches (SDN gateways)
  - **Cloud gateways** - connecting DC to the Virtual Private Clouds (VPCs) - extending the networks to one or multiple cloud providers and offer multi-Cloud approaches
  - **SDN gateways** (e.g. Tungsten gateway) offer a possibility to extend networks between DCs and also mix/match traffic to different VPNs - could be very interesting for federated approaches

Summary

- **Cloud native DC networking approaches offer ways to build scalable, robust and effective DC networks**
  - They have the potential to radically change the way how DC networking is build up and impact all the other areas
- **Network disaggregation** and open network environments are becoming mainstream at many cloud providers
- SENSE and BigData Express leading projects in **programmable networks** and data transfers, but non-OpenFlow approaches are also being investigated (NOTED)
- NFV WG report surveys the existing approaches and finalises Phase I
- LHCONE/LHCOPN meeting at CERN in January 2020 should provide decision point on Phase II
- We welcome additional contributions; contact us if you are interested!

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

WG Report: https://docs.google.com/document/d/1w7XUPxE23DJXn--j-M3KvXIfXHUnYgsVUhBpKFjyjUQ/edit#

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

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

Tungsten Fabric architectural overview: https://tungstenfabric.github.io/website/Tungsten-Fabric-Architecture.html

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

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

Cloud native Data Centre (book) - https://www.amazon.com/Cloud-Native-Data-Center-Networking-Architecture/dp/1492045608/ref=sr_1_2?keywords=cloud+native+data+center+networking&qid=1568122189&s=gateway&sr=8-2

MPLS in the SDN Era (book) - https://www.amazon.com/MPLS-SDN-Era-Interoperable-Scenarios/dp/149190545X/ref=sr_1_1?keywords=MPLS+in+the+SDN&qid=1568122219&s=gateway&sr=8-1

Backup slides

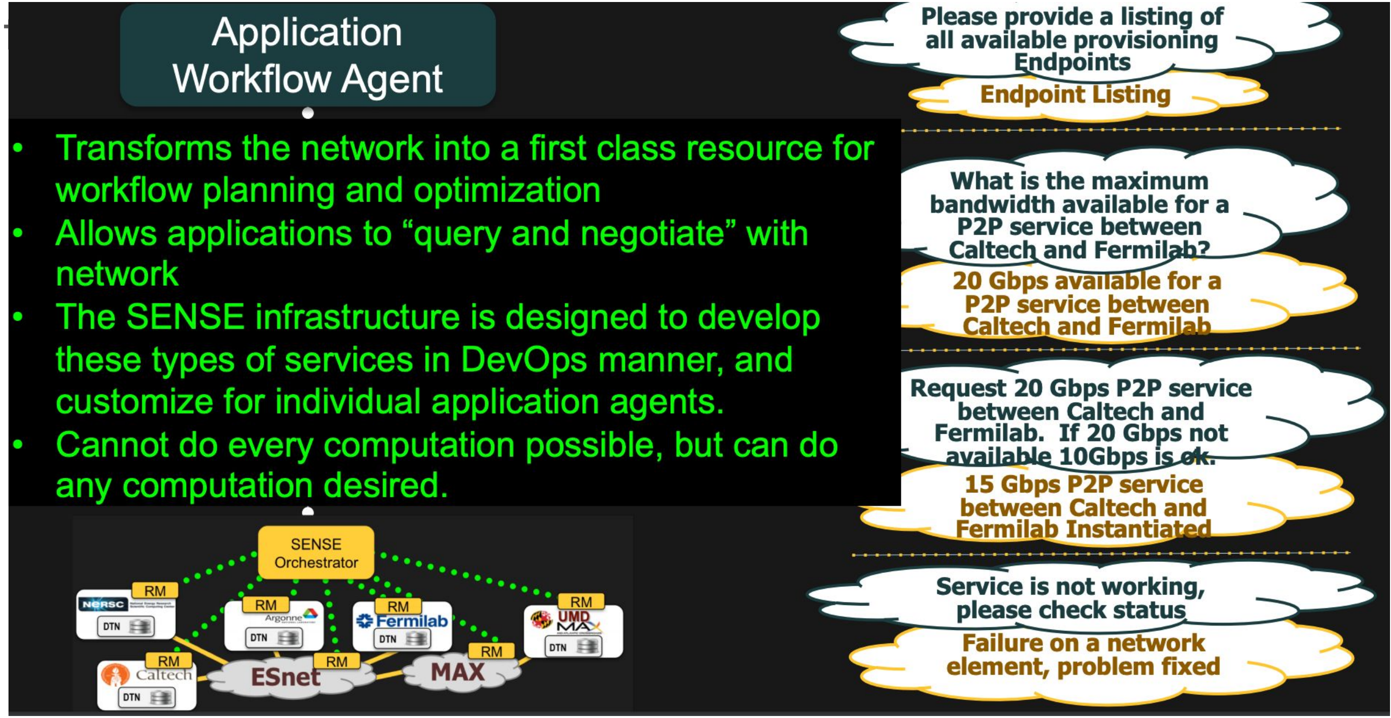
Use Cases

Data centre networking offering standard cloud compute services

- Native support for multi-stack
  - Connecting and integrating multiple orchestration stacks like k8s, OpenStack, etc.
  - Networking and security across legacy, virtualized and containerized applications
- Network support across-stack
  - Networking and security across legacy, virtualized and containerized applications
- Native support for multi-cloud
  - Extending DC networks to Commercial Clouds and creating federated services spanning DCs
- Multi-tenancy/isolation
  - Support for application/experiment level networking (e.g., MultiONE presentation earlier)
- Network automation
- Security and observability
  - Multistack and across-stack policy control, visibility and analytics

SENSE

SDN for End-to-end Networked Science at the Exascale (SENSE) - U.S. DOE
t nd



Application Workflow Agent

- Transforms the network into a first class resource for workflow planning and optimization
- Allows applications to “query and negotiate” with network
- The SENSE infrastructure is designed to develop these types of services in DevOps manner, and customize for individual application agents.
- Cannot do every computation possible, but can do any computation desired.

Please provide a listing of all available provisioning Endpoints

Endpoint Listing

What is the maximum bandwidth available for a P2P service between Caltech and Fermilab?

20 Gbps available for a P2P service between Caltech and Fermilab

Request 20 Gbps P2P service between Caltech and Fermilab. If 20 Gbps not available 10Gbps is ok.

15 Gbps P2P service between Caltech and Fermilab Instantiated

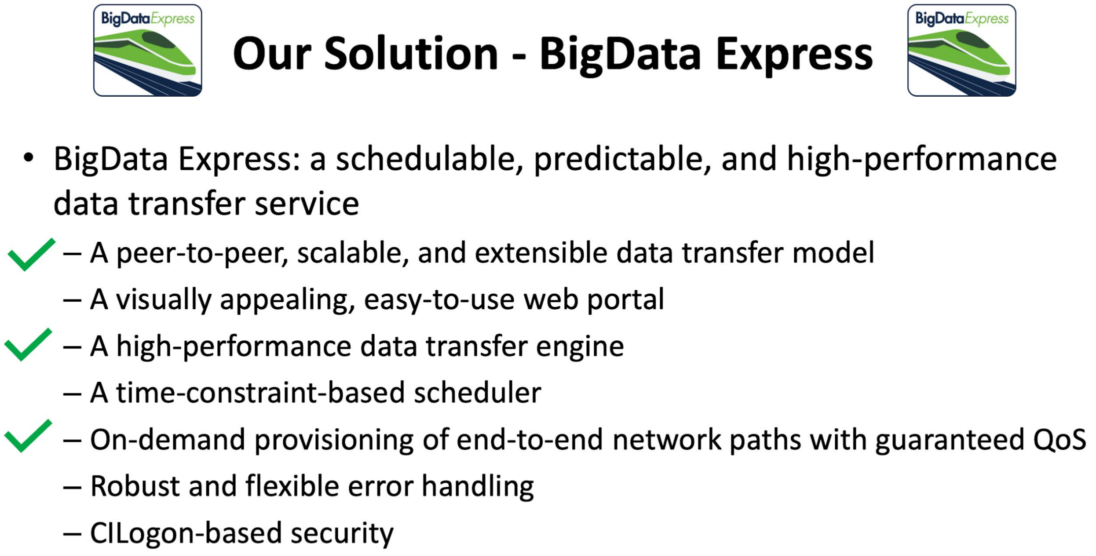
Service is not working, please check status

Failure on a network element, problem fixed

Source: http://conferences.computer.org/scw/2018/#!/toc/3; https://indico.cern.ch/event/795430/

BigData Express

U.S DOE funded; FNAL, ESNet, StarLight, KISTI, Univ. of Maryland, ORNL



Our Solution - BigData Express

- BigData Express: a schedulable, predictable, and high-performance data transfer service
  - ✓ – A peer-to-peer, scalable, and extensible data transfer model
  - – A visually appealing, easy-to-use web portal
  - ✓ – A high-performance data transfer engine
  - – A time-constraint-based scheduler
  - ✓ – On-demand provisioning of end-to-end network paths with guaranteed QoS
  - – Robust and flexible error handling
  - – CILogon-based security

Existing projects also in ATLAS (OVS btw AGLT2/MWT2/KIT), SDN aspects also in NSF-funded **SLATE**, **OSIRIS** and CERN's **NOTED** project

Source: http://conferences.computer.org/scw/2018/#!/toc/3; https://indico.cern.ch/event/777367/

Network Virtualisation - OpenFlow

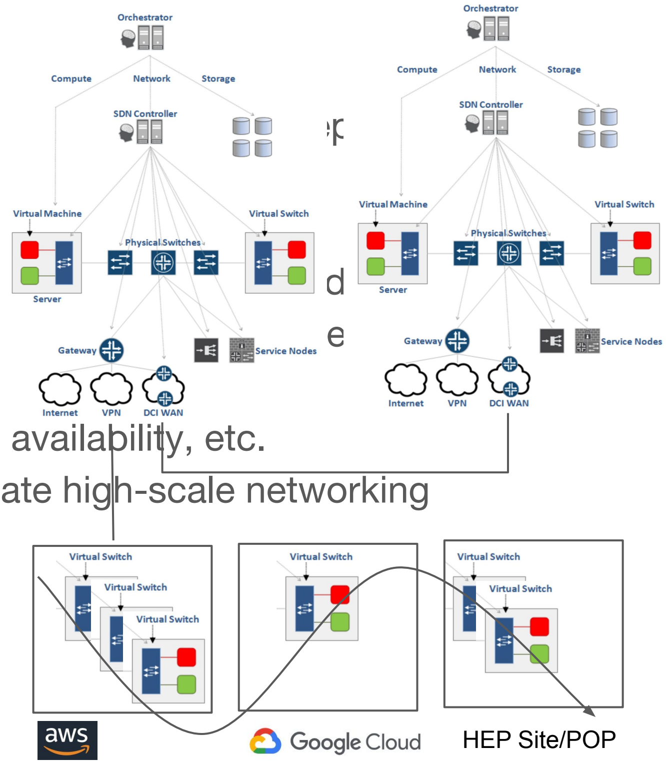
- OpenFlow started with an influential paper and became a movement in networking R&D
  - The core idea is to use flow tables (available in most packet switching silicon) and use OpenFlow protocol to remotely program the tables (from a centralised controller)
- OpenVSwitch (OVS) is an open source implementation of pure OpenFlow software switch
  - Native controller to program it is Open Virtual Network (OVN) (but others can be used as well)
  - Data plane can use VXLAN, GRE, Geneve; control plane is OpenFlow or native OVSDB
  - Controller supports integration with OpenStack and K8s
- OpenFlow protocol has been updated several times to address its shortcomings and overall didn’t live up to its expectations
  - However there are existing production deployments (Google)
  - OpenFlow as such has proven to be very useful in other areas (WAN use cases)
  - Flow tables are still core part of some key network functions (ACLs, NAT, etc.)

SmartNICs

- Now offered from multiple vendors - goal is to maximise capacity while providing full programmability for virtual switching and routing, tunnelling (VXLAN, MPLS), ACLs and security groups, etc.
- Three approaches are being followed:
  - FPGA based - good performance, but difficult to program, workload specific optimisation
  - ASIC based - best price/performance, easy to program but extensibility limited to pre-defined capabilities
  - SOC based - good price/performance, easily programmable, highest flexibility
- Datapath programmability (tutorial)
  - Application level - OpenVSwitch, Tungsten vRouter, etc.
  - Packet movement infrastructure (part of data path) - BPF (Berkeley Packet Filter)/eBPF
  - Full description of data path - P4 language
- FPGA-based SmartNICs broadly deployed in Microsoft Azure
- Tungest Fabric 5.1 release plans to support smartNICs
- Good overview provided in ACM SIGARCH article

DC Edge - Multi-Cloud - DCI/Remote Compute

- SDN-based DC enables other interesting options
- **Data Center Interconnect (DCI)**
  - SDN services spanning multiple physical sites, each site
  - Agnostic to the Virtual Infrastructure Manager (Orchestrat
- **Remote Compute**
  - Single SDN deployment extending its services to remote extend VPNs/VMs to another site without running a dedic
- **Service chaining (NFV)**
  - Steering traffic between VPNs/VMs according to a policy, availability, etc.
- All the options are complementary and can be combined to create high-scale networking combining 100s or even 1000s of sites.



Networking Challenges

- Capacity/share for data intensive sciences
  - No issues wrt available technology, however
  - What if N more HEP-scale science domains start competing for the same resources ?
- Remote data access proliferating in the current DDM design
  - Promoted as a way to solve challenges within experiment’s DDM
  - Different patterns of network usage emerging
    - Moving from large streams to a mix of large and small frequent event streams
- Integration of Commercial Clouds
  - Impact on funding, usage policies, security, etc.
- Technology evolution
  - Software Defined Networking (SDN)/Network Functions Virtualisation (NFV)

Technology Impact

- Increased importance to oversee network capacities
  - Past and anticipated network usage by the experiments, including details on future workflows
- New technologies will make it easier to transfer vast amounts of data
  - HEP quite likely no longer the only domain that will need high throughput
- Sharing the future capacity will require greater interaction with networks
  - While unclear on what technologies will become mainstream (see later), we know that software will play a major role in the networks of the future
  - We have an opportunity here
- It’s already clear that software will play major role in networks in the mid-term
- Important to understand how we can design, test and develop systems that could enter existing production workflows
  - **While at the same time changing something as fundamental as the network that all sites and experiments rely upon**
  - We need to engage sites, experiments and (N)REN(s) in this effort

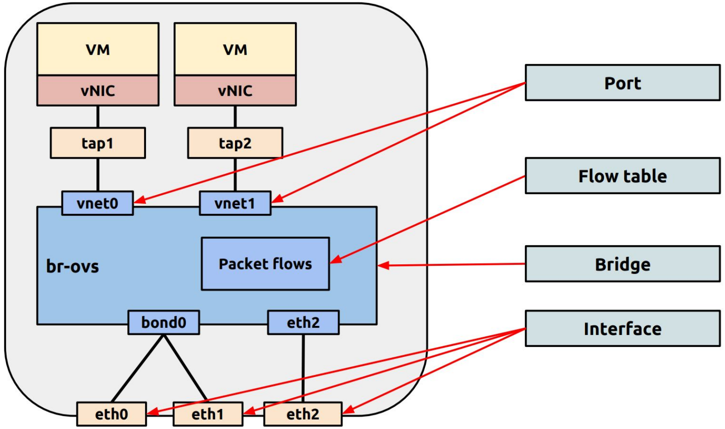
Software Defined Networks (SDN)

- Software Defined Networking (SDN) are a set of new technologies enabling the following use cases:
  - **Automated service delivery** - providing on-demand network services (bandwidth scheduling, dynamic VPN)
  - **Clouds/NFV** - agile service delivery on cloud infrastructures usually delivered via Network Functions Virtualisation (NFV) - underlays are usually Cloud Compute Technologies, i.e. OpenStack/Kubernetes/Docker
  - **Network Resource Optimisation (NRO)** - dynamically optimising the network based on its load and state. Optimising the network using near real-time traffic, topology and equipment. This is the core area for improving end-to-end transfers and provide potential backend technology for DataLakes
  - **Visibility and Control** - improve our insights into existing network and provide ways for smarter monitoring and control
- Many different point-to-point efforts and successes reported within LHCOPN/LHCONE
  - **Primary challenge is getting end-to-end!**
- While it's still unclear which technologies will become mainstream, it's already clear that software will play major role in networks in the mid-term
  - Massive network automation is possible - in production and at large-scale
- HEPiX SDN/NFV Working Group was formed to bring together sites, experiments, (N)RENs and engage them in testing, deploying and evaluating network virtualization technologies

Software Switches

Open vSwitch (OVS) - open source multilayer virtual switch supporting standard interfaces and protocols:

- OpenFlow, STP 802.1d, RSTP,
- Advanced Control, Forwarding, Tunneling
- Primarily motivated to enable VM-to-VM networking, but grew to become the core component in most of the existing open source cloud networking solutions



Runs as any other standard Linux app - user-level controller with kernel-level datapath including HW off-loading (recent) and acceleration (Intel DPDK)

Enables massive network automation …

Open vSwitch Features

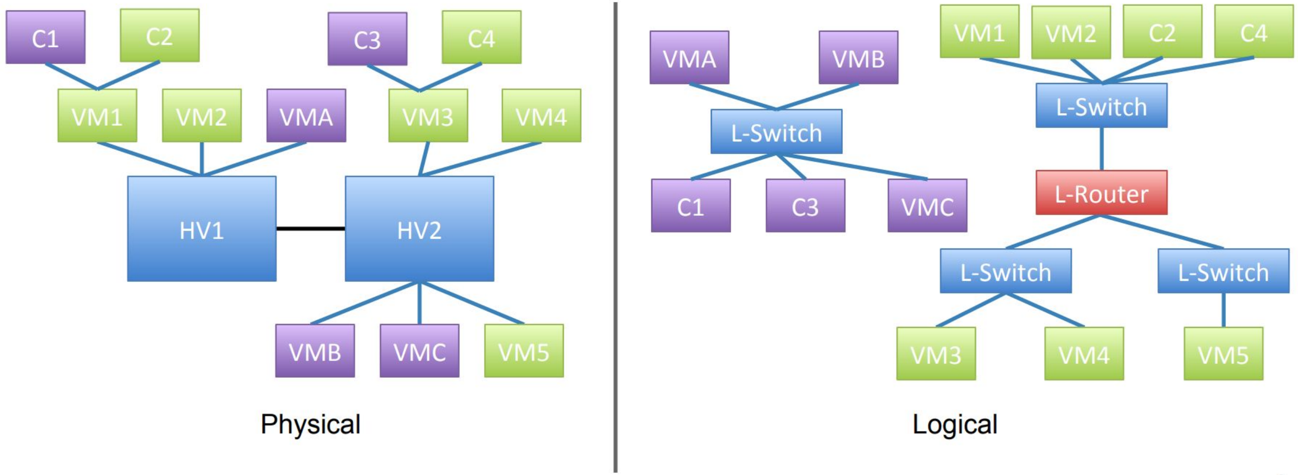
- Visibility into inter-VM communication via NetFlow, sFlow(R), IPFIX, SPAN, RSPAN, and GRE-tunneled mirrors
- LACP (IEEE 802.1AX-2008)
- Standard 802.1Q VLAN model with trunking
- Multicast snooping
- IETF Auto-Attach SPBM and rudimentary required LLDP support
- BFD and 802.1ag link monitoring
- STP (IEEE 802.1D-1998) and RSTP (IEEE 802.1D-2004)
- Fine-grained QoS control
- Support for HFSC qdisc
- Per VM interface traffic policing
- NIC bonding with source-MAC load balancing, active backup, and L4 hashing
- OpenFlow protocol support (including many extensions for virtualization)
- IPv6 support
- Multiple tunneling protocols (GRE, VXLAN, STT, and Geneve, with IPsec support)
- Remote configuration protocol with C and Python bindings
- Kernel and user-space forwarding engine options
- Multi-table forwarding pipeline with flow-caching engine
- Forwarding layer abstraction to ease porting to new software and hardware platforms

Controllers - Open DayLight

- Modular open platform for customizing and automating networks of any size and scale. Core use cases include:
  - **Cloud and NFV** - service delivery on cloud infrastructure in either the enterprise or service provider environment
  - **Network Resource Optimisation** - Dynamically optimizing the network based on load and state; support for variety of southbound protocols (OpenFlow, OVSDB, NETCONF, BGP-LS)
  - Automated Service Delivery - Providing on-demand services that may be controlled by the end user or the service provider, e.g. on-demand bandwidth scheduling, dynamic VPN
  - Visibility and Control - Centralized administration of the network and/or multiple controllers.
- Core component in number of open networking frameworks
  - ONAP, OPNFV, OpenStack, etc.
- Integrated or embedded in more than 50 vendor solutions and apps
- ODL is just one of many controllers that are available:
  - OpenContrail, ONOS, MidoNet, Ryu, etc.

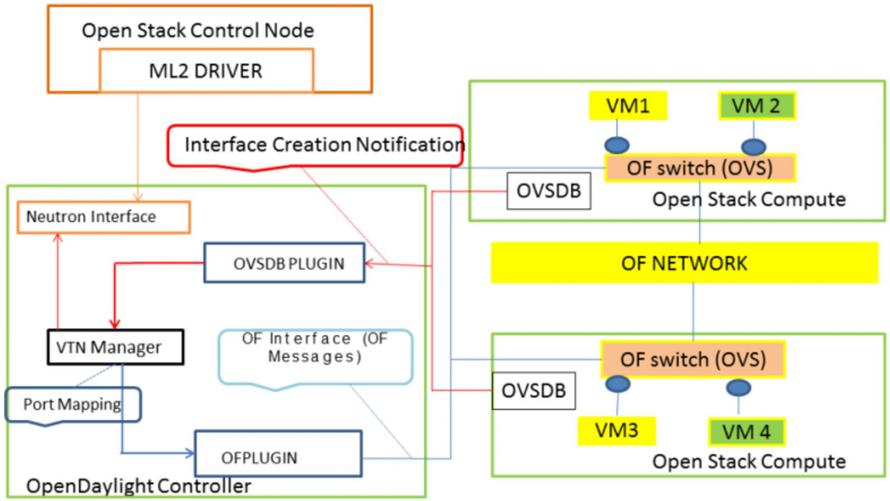
Controllers - Open Virtual Network (OVN)

- Open source logical networking for OVS
- Provides L2/L3 networking
  - Logical Switches; L2/L3/L4 ACLs
  - Logical Routers, Security Groups
  - Multiple Tunnel overlays (Geneve, VXLAN)
  - Top-of-rack-based & software-based physical-to-logical gateways



Cloud Compute - OpenStack Networking

- Cloud stresses networks like never before
  - Massive scale, Multi-tenancy/high density, VM mobility
- OpenStack Neutron offers a plugin technology to enable different (SDN) networking approaches - brings all previously mentioned techs together



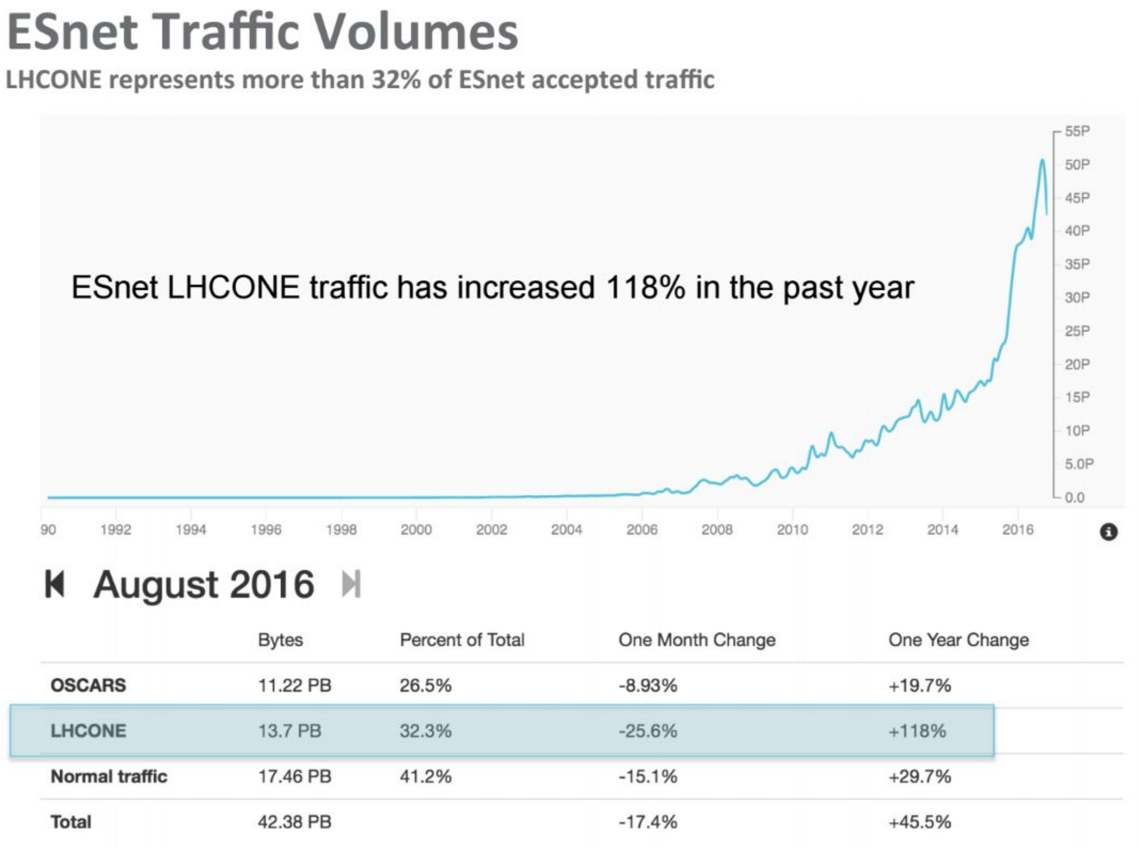
ML2 driver is what makes controllers pluggable, so you can easily replace Neutron controller with OpenDaylight, OVN, etc.

Both generic and vendor-specific plugins are available

Cumulus Linux

- Alternative to OVS - uses separate apps/kernel functions to program different functionality such as STP/RSTP (mstpd), VXLAN (ifupdown2), VLAN (native linux bridge) etc.
- It does contain OVS to enable integration with controllers:
  - VMware NSX, Midokura Midonet, etc.
- Unlike OVS, Cumulus Linux is not an app, but a distribution, which is certified to run on bare metal switches
  - The list of supported HW is at (https://cumulusnetworks.com/products/hardware-compatibility-list/)
  - Mainly Broadcom Tomahawk, Trident2/+, Helix4 and Mellanox Spectrum ASICs
- Otherwise runs like standard Linux, which means compute and network "speak the same language"
  - E.g. automation with Ansible, Puppet, Chef, etc.

R&E Traffic Growth Last Year



ESnet Traffic Volumes

LHCONE represents more than 32% of ESnet accepted traffic

ESnet LHCONE traffic has increased 118% in the past year

August 2016

	Bytes	Percent of Total	One Month Change	One Year Change
OSCARS	11.22 PB	26.5%	-8.93%	+19.7%
LHCONE	13.7 PB	32.3%	-25.6%	+118%
Normal traffic	17.46 PB	41.2%	-15.1%	+29.7%
Total	42.38 PB		-17.4%	+45.5%

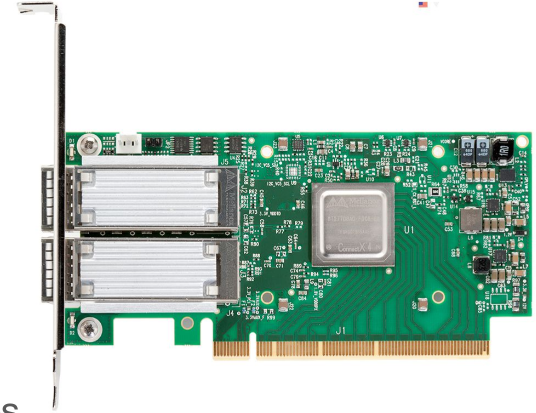
Slide from Michael O'Connor, LHCONE operations update

In general, ESNet sees overall traffic grow at factor 10 every 4 years. Recent LHC traffic appears to match this trend.

GEANT reported LHCONE peaks of over 100Gbps with traffic increase of 65% in the last year.

This has caused stresses on the available network capacity due to the LHC performing better than expected, but the situation is unlikely to improve in the long-term.

WAN vs LAN capacity



- Historically WAN capacity has not always had a stable relationship compared to data-centre
  - In recent history WAN technologies grew rapidly and for a while outpaced LAN or even local computing bus capacities
  - Today 100Gbps WAN links are the typical high-performance network speed, but LANs are also getting in the same range
    - List price for 100Gbit dual port card is ~ $1000, but significant discounts can be found (as low as $400), list price for 16 port 100Gbit switch is $9000
- Today it is easy to over-subscribe WAN links
  - in terms of $ of local hardware at many sites
- Will WAN be able to keep up ? **Likely yes,** however:
  - We did benefit from the fact that 100Gbit was deployed on time for Run2, might not be the case for Run3 and 4
  - By 2020 800 Gbps waves likely available, but at significant cost since those can be only deployed at proportionally shorter distances

Improving Our Use of the Network

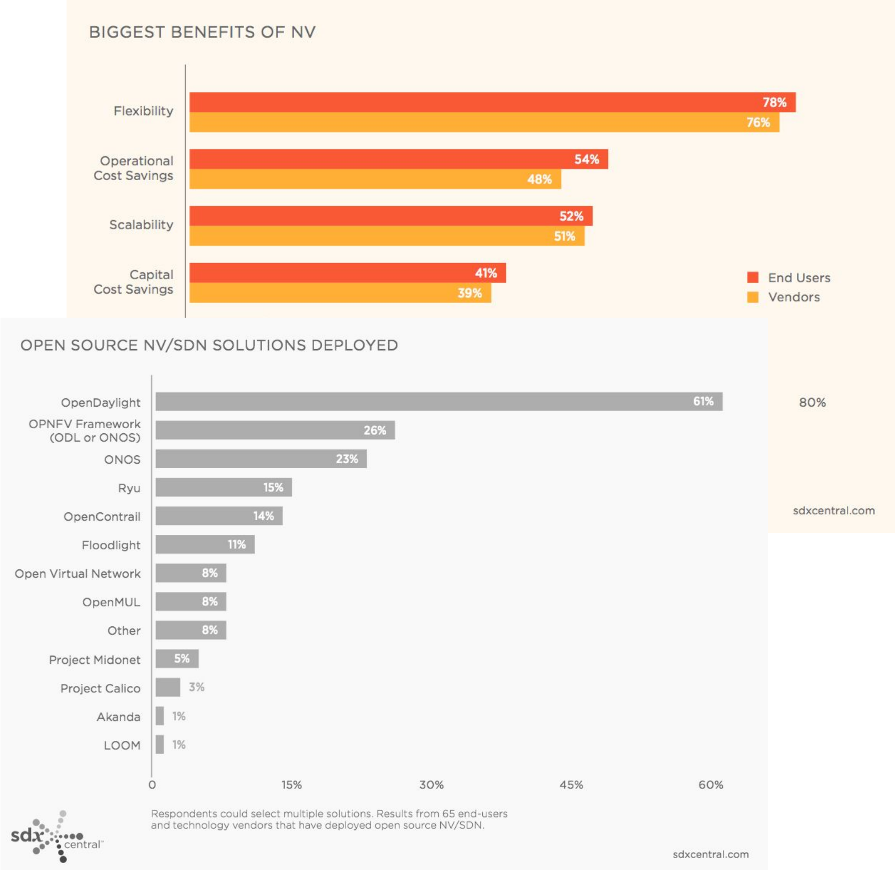
- TCP more stable in CC7, throughput ramp ups much quicker
  - Detailed report available from Brian Tierney/ESNet
- Fair Queueing Scheduler (FQ) available from kernel 3.11+
  - Even more stable, works better with small buffers
  - Pacing and shaping of traffic reliably to 32Gbps
- Best single flow tests show TCP LAN at 79Gbps, WAN (RTT 92ms) at 49Gbps
  - IPv6 slightly faster on the WAN, slightly slower on the LAN
- **In summary: new enhancements make tuning easier in general**
  - But some previous “tricks” no longer apply
- New TCP congestion algorithm (TCP BBR) from Google
  - Google reports factor 2-4 performance improvement on path with 1% loss (100ms RTT)
  - Early testing from ESNet less conclusive and questions need answering

R&E Networking

- R&E network providers have long been working closely with HEP community
  - HEP has been representative of the future data intensive science domains
  - Often serving as testbed environment for early prototypes
- Big data analytics requiring high throughput no longer limited to HEP
  - SKA (Square Kilometer Array) plans to operate at data volumes 200x current LHC scale
  - Besides Astronomy there are MANY science domains anticipating data scales beyond LHC, cf. ESRFI 2016 roadmap
- **What if N more HEP-scale science domains start competing for the same network resources ?**
  - Will HEP continue to enjoy “unlimited” bandwidth and prioritised attention or will we need to compete for the networks with other data intensive science domains ?
  - Will there be **AstroONE**, **BioONE**, etc., soon ?

Tech Trends: Software Defined Networks (SDN)

- SDN is a set of technologies offering solutions for many of the future challenges
  - ○ Current links can handle ~ 6x more traffic if we could avoid peaks and be more efficient
  - ○ SDN driven by commercial efforts
- Many different point-to-point efforts and successes reported within LHCOPN/LHCONE
  - ○ **Primary challenge is getting end-to-end!**
- While it's still unclear which technologies will become mainstream, it's already clear that software will play major role in networks in the mid-term
  - ○ Will experiments have effort to engage in the existing SDN testbeds to determine what impact it will have on their data management and operations ?



Respondents could select multiple solutions. Results from 65 end-users and technology vendors that have deployed open source NV/SDN.

Tech Trends: SD-WAN

- Large Network as a Service providers include several well established CSPs such as Amazon, Rackspace, AT&T, Telefonica, etc.
- Recently more niche NaaS providers have appeared offering SD-WAN solutions
  - Aryaka, Cloudgenix, Pertino, VeloCloud, etc.
  - Their offering is currently limited and not suitable for high throughput, but evolving fast
- SD-WAN market is estimated to grow to $6 billion in 2020 (sdxcentral)
- Will low cost WAN become available in a similar manner we are now buying cloud compute and storage services ?
  - Unlikely, our networks are shared, not easy to separate just LHC traffic
  - Transit within major cloud providers such as Amazon currently not possible and unlikely in the future, limited by regional business model - but great opportunity for NRENs

Tech Trends: Containers

- Recently there has been a strong interest in the container-based systems such as Docker
  - They offer a way to deploy and run distributed applications
  - Containers are lightweight - many of them can run on a single VM or physical host with shared OS
  - Greater portability since application is written to container interface not OS
- Obviously networking is a major limitation to containerization
  - Network virtualization, network programmability and separation between data and control plane are essential
  - Tools such as Flocker or Rancher can be used to create virtual overlay networks to connect containers across hosts and over larger networks (data centers, WAN)
- Containers have great potential to become disruptive in accelerating **SDN** and **merging LAN and WAN**
  - But clearly campus SDNs and WAN SDNs will evolve at different pace

Network Operations

- Deployment of perfSONARs at all WLCG sites made it possible for us to see and debug end-to-end network problems
  - OSG is gathering global perfSONAR data and making it available to WLCG and others
- A group focusing on helping sites and experiments with network issues using perfSONAR was formed - WLCG Network Throughput
  - Reports of non-performing links are actually quite common (almost on a weekly basis)
  - Most of the end-to-end issues are due to faulty switches or mis-configurations at sites
  - Some cases also due to link saturation (recently in LHCOPN) or issues at NRENs
- Recent network analytics of LHCOPN/LHCONE perfSONAR data also point out some very interesting facts:
  - Packet loss greater than 2% for a period of 3 hours on almost 5% of all LHCONE links
- Network telemetry (real-time network link usage) likely to become available in the mid-term (but likely not from all NRENs at the same time)
- ~~It is increasingly important to focus on site-based network operations~~