FERMILAB-CONF-18-824-CCD-SCD

BigData Express: Toward Schedulable, Predictable, and High-Performance Data Transfer

Qiming Lu, Liang Zhang, S. Sasidharan, Wenji Wu, Phil DeMar
{qlu, liangz, sajith, wenji, demar}@fnal.gov
Fermilab

Chin Guok, John Macauley, Inder Monga
{chin, macauley,imonga}@es.net
ESnet

Se-young Yu, Jim Hao Chen, Joe Mambretti
{young.yu, jim-chen, j-mambretti}@northwestern.edu
Northwstern University

Jin Kim, Seo-Young Noh
{jkim, rsyoung}@kisti.re.kr
KISTI

Xi Yang, Tom Lehman
{maxyang, tlehman}@umd.edu
The University of Maryland

Gary Liu
qing.liu@njit.edu
NJIT

Abstract— **Big Data has emerged as a driving force for scientific discoveries. Large scientific instruments (e.g., colliders, and telescopes) generate exponentially increasing volumes of data. To enable scientific discovery, science data must be collected, indexed, archived, shared, and analyzed, typically in a widely distributed, highly collaborative manner. Data transfer is now an essential function for science discoveries, particularly within big data environments. Although significant improvements have been made in the area of bulk data transfer, the currently available data transfer tools and services can not successfully address the high-performance and time-constraint challenges of data transfer required by extreme-scale science applications for the following reasons: disjoint end-to-end data transfer loops, cross-interference between data transfers, and existing data transfer tools and services are oblivious to user requirements (deadline and QoS requirements). Fermilab has been working on the BigData Express project to address these problems. BigData Express seeks to provide a schedulable, predictable, and high-performance data transfer service for big data science. The BigData Express software is being deployed and evaluated at multiple research institutions, which include UMD, StarLight, FNAL, KISTI, KSTAR, SURFnet, Ciena, and other sites. Meanwhile, the BigData Express research team is collaborating with the StarLight International/National Communications Exchange Facility to deploy BigData Express at various research platforms, including Pacific Research Platform, National Research Platform, and Global Research Platform. It is envisioned that we are working toward building a high-performance data transfer federation for big data science.**

Keywords—big data, high-performance data transfer, DTN, SDN, co-scheduling, high-speed networking, performance

I. Introduction

Big data has emerged as a driving force for scientific discoveries [1]. Large scientific instruments (e.g., colliders, light sources, and telescopes) generate exponentially increasing volumes of data. Currently, Large Hadron Collider (LHC) experiments generate hundreds of petabytes of data per year. The aggregated amount of climate science data is expected to exceed 100 exabytes by 2020. To enable scientific discovery, science data must be collected, indexed, archived, shared, and analyzed, typically in a widely distributed, highly collaborative manner [2-7]. At present, computing facilities for large-scale science, such as ALCF, OLCF, and NERSC, offer the types of computing and storage resources needed to process and analyze science data. The efficient movement of science data from their sources into processing and storage facilities and ultimately on to user analysis is critical to the success of any such endeavor. Data transfer is now an essential function for science discoveries, particularly within big data environments.

Within the U.S. research communities, the emergence of distributed, extreme-scale science applications is generating significant challenges regarding data transfer [2-7]. We believe that the data transfer challenges of the extreme-scale era are characterized by two relevant dimensions:

- *High-performance challenges*. First, it is becoming critical to transfer data at the highest possible throughputs because the volumes of science data are growing exponentially. Second, the U.S. research communities are working toward deploying extreme-scale supercomputer facilities in support of extreme-scale science applications. To fully utilize these expensive computing facilities, ultra-high-throughput data transfer capabilities will be required to move data in or out of them.
- *Time-constraint challenges*. Scientific applications typically have explicit or implicit time constraints on data transfer. Based on the nature of these time constraints, data transfer tasks can be classified into three broad categories: *(a) Real-time, (b) deadline-bound, and (c) background data transfer*. For *real-time data transfer*, the data transfer task is on the critical path for the end-user experience or a real-time experimental control loop. Scientific applications, such as real-time data analysis, remote visualization, and real-time experimental control, are highly sensitive to data transfer delays. Even small increases in data transfer time can degrade the user experience or result in inaccurate scientific results. For *Deadline-bound data transfer*, the data transfer task is not on the critical path for the end-user experience or a real-time experimental control loop, but it does have an explicit deadline. For example, job startup and scratch storage space purge deadlines in supercomputer centers require deadline-bound data movement. for *Background data transfer*, the data transfer task has a long deadline or no explicit deadline. For example, replicating data from one data center to another data center for long-term storage is a background data transfer task.

This work is supported by the U.S. Department of Energy (DOE) ASCR Network Research Program

This manuscript has been authored by Fermi Research Alliance, LLC under Contract No. DE-AC02-07CH11359 with the U.S. Department of Energy, Office of Science, Office of High Energy Physics.

To date, several data transfer tools (e.g., GridFTP [8-9] and BBCP [10]) and services (e.g., the PhEDEx and Ruccio systems [11-12][36], the LIGO Data Replicator [13], and Globus Online [14]) have been developed to support science data movement. Advanced data transfer features, such as transfer resumption, partial transfer, third-party transfer, and security, have been implemented in these tools and services. There have also been numerous enhancements to speed up data transfer performance, including the following:

- Parallelism at all levels (e.g., multi-stream parallelism [8], multicore parallelism [15], and multi-path parallelism [16-19]) is widely implemented in bulk data movement and offers significant improvement in aggregate data transfer throughput.
- Science DMZ architectures [20] with dedicated high-performance Data Transfer Nodes (DTNs) have been widely deployed. The hardware devices, software, configurations, and policies of Science DMZ are structured and optimized for high-performance data transfer.
- The U. S. research communities are working toward deploying terabit networks in support of distributed extreme-scale data movement. Existing backbone networks are now based on ultra-scalable 100-gigabit technologies. Advanced virtual path services such as ESnet OSCARS [21] and Internet2 AL2S [22] has been developed.

Although significant improvements have been made in science data transfer capabilities, the currently available data transfer tools and services will not be able to successfully address the high-performance and time-constraint challenges of data transfer to support extreme-scale science applications for the following reasons:

(1) *Problem 1: Disjoint end-to-end data transfer loops*. In current data transfer frameworks, each entity in an end-to-end data transfer loop (e.g., DTN, LAN, WAN, and storage) is scheduled and managed locally, and their policies and mechanisms may act at odds with each other. Without end-to-end integration and coordination, this distributed resource management model may readily lead to resource contention or performance mismatch in the end-to-end loop. As a result, suboptimal (or even poor) performance would occur.

(2) *Problem 2: Cross-interference between data transfers.* A significant amount of cross-interference between data transfers can lead to contention for various resources (e.g., DTN, LAN, WAN, and storage), resulting in degraded performance. This can also lead to high variability in data transfer performance. Existing data transfer tools and services lack effective mechanisms to minimize cross-interference between data transfers.

(3) *Problem 3: Existing data transfer tools and services are oblivious to user (or user application) requirements (e.g., deadlines and QoS requirements).* Without deadline awareness, it is difficult to satisfy the time constraint requirement on data transfer.

(4) *Problem 4: Inefficiencies arise when existing data transfer tools are run on DTNs*. High-end DTNs are typically NUMA systems. However, existing data transfer tools are unable to fully exploit multicore hardware under the default OS support, especially on NUMA systems.

If these problems are not addressed appropriately, they will undermine the ability to support extreme-scale science in the coming years.

Fermilab has been working on the BigData Express project (http://bigdataexpress.fnal.gov) to address these problems. BigData Express seeks to provide a schedulable, predictable, and high-performance data transfer service for big data science. Essentially, *BigData Express* is a middleware data transfer service with the following key features:

- A data-transfer-centric architecture to seamlessly integrate and effectively coordinate the resources (e.g., DTNs, network, and storage resources) in an end-to-end data transfer loop. Within an end site (i.e., a data source or destination site), BigData Express schedules and manages the local resources (i.e., DTNs, LAN, and storage resources) for data transfer. Resources will be scheduled and assigned based on user requirements, task priorities, and resource status and usage policy information. Multiple DTNs can be provisioned to participate in a single data transfer task. BigData Express will directly schedule local network resources if local SDN capabilities are available. In addition, a distributed rate-based resource brokering mechanism is implemented to coordinate resource allocation across autonomous sites (i.e., data transfer source/destination sites and WANs). Finally, a distributed DTN matching mechanism has been implemented to coordinate and match heterogeneous DTNs at different sites to avoid DTN performance mismatch.
- A time-constraint-based scheduler to schedule data transfer tasks. By allowing user applications to inform the scheduler of their time constraints, the scheduler can prioritize requests from different applications to satisfy as many time constraints as possible. The scheduler acts in two modes: (1) in an event-driven mode, whenever a new data transfer request is submitted, or an old data transfer task is completed; and (2) in a periodic mode to reschedule and reassign resources periodically to adapt rapidly to changing run-time environments.
- An admission control mechanism to provide guaranteed resources for admitted data transfer tasks. A data transfer that cannot satisfy its time constraints without violating others will not be admitted.
- A distributed peer-to-to model for data transfer services, making it very flexible for the establishment of data transfer federations.
- A scalable software architecture. BigData Express makes use of MQTT [23] as message bus to support communication among its components.
- An extensible plugin framework to support different data transfer protocols, including mdtmFTP, GridFTP, and XrootD.

- An end-to-end data transfer model with fast provisioning of end-to-end network paths for guaranteed QoS. Specifically, the use of an SDN-enabled BigData-Express LANs and SDN-enabled WAN path services to reduce or eliminate network congestion.
- A high-performance data transfer engine. BigData Express adopts mdtmFTP as its default data transfer engine. mdtmFTP is specifically designed for optimization of data transfer performance on multicore systems (DTNs).

The BigData Express software is currently deployed and being evaluated at multiple research institutions, including UMD, StarLight, FNAL, KISTI, KSTAR, SURFnet, and Ciena. The BigData Express research team is collaborating with StarLight to deploy BigData Express on various research platforms, including Pacific Research Platform, National Research Platform, and Global Research Platform. It is envisioned that we are working toward building a high-performance data transfer federation for big data science.

The rest of paper is organized as follows. Section II presents background and related works. Section III discusses BigData Express design and implementation. Section IV discusses our initial evaluation of BigData Express. And Section VI concludes the paper.

II. Backgrounds and Related works

A. Data Transfer Tools and Services

Several data movement tools and technologies have been developed, such as TCP-based GridFTP [8-9], BBCP [10], and UDP-based UDT [24]. TCP-based tools are widely used in shared network environments. However, they typically encounter performance problems on high-speed networks because the TCP congestion control algorithm limits the efficiency of network resource utilization. There have been numerous efforts to scale TCP over high-bandwidth networks, such as FAST TCP [25], and CUBIC-TCP [26]. Alternatively, to overcome TCP's inefficiency on high-speed networks, UDP-based tools have been proposed as TCP replacements. These tools include Reliable Blast UDP and UDP-based data transport (UDT) [24]. Applications can benefit from selecting among the various available tools and technologies and adapting them to different networking environments. For example, in certain cases, exclusive access to the entire connection bandwidth could obviate the need for complex TCP mechanisms. Alternative transmission protocols, such as NACK-based UDT, that can make more efficient use of dedicated channels may provide a simpler, more efficient approach to data transfer.

In reality, many abnormal conditions may arise in bulk data transfer, including server failures, cut/dirty fibers, and line card malfunctions. Researchers have developed several data transfer services on tops of data transfer tools (e.g., GridFTP) to automate bulk data transfer. The High Energy Physics (HEP) community developed the PhEDEx and Ruccio systems to manage data movement for the LHC experiments [11-12][36]. The Laser Interferometer Gravitational Wave Observatory (LIGO) project developed the LIGO Data Replicator [13]. Argonne National Laboratory and the University of Chicago have developed Globus Online to manage fire-and-forget file transfers [14].

Private industry has also provided several data transfer services—Dropbox, Hightail, Akamai, and Windows Azure CDN. However, these services typically are not suitable for big science data transfer.

B. DTNs and The Science DMZ approach

A DTN is a computer system dedicated to the function of wide-area data transfer. Because of the scalability advantage of NUMA (non-uniform memory access) architecture, high-performance DTNs are typically NUMA-based and feature several nodes distributed across the system. Each node consists of a few cores, local memories, and/or I/O devices. A high-performance DTN is typically configured with one or multiple 10/25/40GE NICs today, with 50/100GE NICs on the horizon.

Science DMZ [20] refers to a specialized DTN deployment that is typically local to a site's network perimeter. The hardware devices, software, configuration, and policies of Science DMZ are structured and optimized for high-performance data transfer. DOE Leadership Computing Facilities and many university networks are now adopting the Science DMZ architecture to deploy DTNs.

C. Terabit networks

The U.S. Research and Education networks have deployed 100GE-based network infrastructure in place today to support the extreme-scale data movement of big science. Existing R&E network backbones are based on ultra-scalable 100-gigabit network technologies. Within the data center, server performance growth drives system 25/40/50/100GE connection deployments, with n x 100 GE uplinks in the LAN. The deployment of this very scalable network infrastructure at all levels provides the resource framework for meeting the high-performance and time-constraint data transfer challenges of big data science.

D. ESnet OSCARS and Internet2 AL2S

ESnet has developed the OSCARS network reservation system [21] to reserve and provision multi-domain, high-bandwidth layer-2 circuits across WAN infrastructure. OSCARS can be used to obtain the customized WAN services needed to satisfy the application-specific requirements of large-scale science collaborations.

Internet2's AL2S [22] is a similar network service that provides researches and network engineers the ability to automatically provision dedicated circuits across network domains in support of bandwidth-intensive applications. AL2S leverages ESnet's OSCARS technology.

E. SDN and its promise

SDN is a network architecture that disentangles the control plane from data forwarding, thus allowing the network control to be directly programmable [27-28]. The promise of SDN is that it allows network resources to be managed and re-configured automatically and dynamically, offering immense performance advantages for network operations. In prior work, Hedera [29] designed a dynamic flow scheduling system that adaptively scheduled the switching fabric to reduce traffic collisions. Similarly, Wang et al. [30] used OpenFlow [31] to install wildcard packet-handling rules to balance the loads at each server replica while achieving considerably lower processing overhead. Recently, SWAN [32] improved the link utilization of inter-data-center networks by orchestrating traffic and re-configuring the data plane to match the current traffic demands. Efforts have also been made to reduce the costs associated with fine-grained control in OpenFlow [31]. The success of OpenFlow-based SDN has been demonstrated by Google's B4, a private WAN connecting Google's data centers across the world [33].

F. The CILogon Service

CILogon (*https://cilogon.org*) provides a federated X.509 certification authority for secure access to cyberinfrastructure [34]. The CILogon service is implemented by a web application, with a back-end MyProxy CA that uses InCommon (SAML) for authentication. Users authenticate to CILogon via the SAML protocol using their home institution credentials. The InCommon federation publishes public keys for identity providers (i.e., campuses) and service providers (i.e., CILogon) so they can trust each other. CILogon takes the user information (name, email, unique ID) from the SAML assertion issued by the campus, asks the MyProxy CA to issue a certificate containing that information, and delivers the certificate to the user. CILogon provides multiple interfaces for issuing certificates: web browser, command-line, and OAuth/OIDC. Via the OIDC interface, CILogon can issue JSON ID tokens instead of or in addition to X.509 certificates.

III. BigData Express Design and Implementation

The design of BigData Express follows three high-level principles: parallelism, integration, and cooperation. BigData Express is designed to support three types of data transfers: real-time data transfer, deadline-bound data transfer, and background data transfer.

A. System Design and Architecture

BigData Express will typically run in a data center, such as a DOE Leadership Computing Facility. As illustrated in Figure 1, a typical site will feature a dedicated cluster of high-performance DTNs, an SDN-enabled LAN, and a large-scale storage system.

- The dedicated DTNs are deployed using the Science DMZ architecture. A high-performance DTN is typically a NUMA-based multicore system with multiple NICs configured. Data transfer tool runs on each DTN. mdtmFTP [15] is BigData Express' default data transfer engine.

- The BigData-Express LAN is a network slice dedicated to the DTNs for bulk data transfer. It consists of either physical or virtualized SDN-enabled switches or routers connecting the local DTNs to a GW—a gateway router or switch that connects to external networks. Multiple GWs may exist for a large LAN.

- At Leadership Computing Facilities, the storage system is typically a shared parallel file system (e.g., Lustre File System). All DTNs access this shared storage via a high bandwidth and well-connected Infiniband interconnect.

BigData Express optionally requires an on-demand site-to-site WAN connection service to provide the path(s) between source and destination sites. Normally, the WAN service supports guaranteed bandwidth and designated time slot reservations. ESnet and Internet2 currently are capable of providing such a WAN service via OSCARs and AL2S, respectively. This requirement is necessary for BigData Express to establish end-to-end network paths with guaranteed QoS to support real-time and deadline-bound data transfer. Otherwise, BigData Express would provide best-effort data transfer.

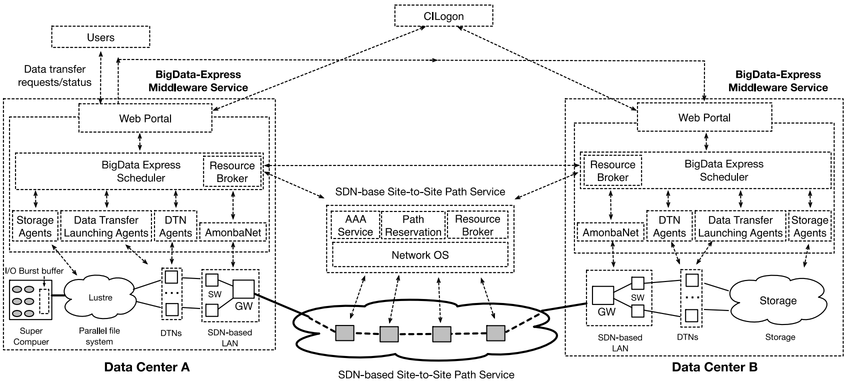

Figure 1 BigData Express architecture

BigData Express adopts a distributed, peer-to-peer model. A logically centralized BigData Express scheduler coordinates all activities at each BigData Express site. This BigData Express scheduler manages and schedules local resources (DTNs, storage, and the BigData Express LAN) through agents (DTN agents, storage agents, and AmoebaNet). Each type of resource may require one or multiple agents. The scheduler communicates with agents through a MQTT-based message bus (Figure 2). This architecture offers flexibility, robustness, and scalability. BigData Express Schedulers located at different sites negotiates and collaborates to execute data transfer tasks. They execute a distributed rate-based resource brokering mechanism to coordinate resource allocation across autonomous sites.

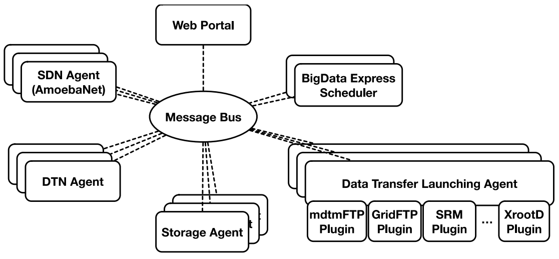

Figure 2 A scalable system architecture

Web Portal allows users and applications to access BigData Express services. For a data transfer task, the following information will be conveyed to BigData Express via Web Portal: the X.509 certificates of the task submitter, the paths and filenames of the data source, the paths of the data destination, the task deadline, and the QoS requirements. BigData Express uses this information to schedule and broker resources for the data transfer task, then launch data transfers. Web portal also allows users to browse file folders, check the data transfer status, or monitor the system/site status.

DTN agents collect and report the DTN configuration and status. They also assign and configure DTNs for data transfer tasks as requested by the BigData Express scheduler.

AmoebaNet [35] keeps track of the BigData Express LAN topology and traffic status with the aid of SDN controllers. As requested by the BigData Express scheduler, AmoebaNet programs local networks at run-time to provide custom network services.

Storage agents keep track of local storage systems usage, provide information regarding storage resource availability and status to the scheduler, and execute storage assignments.

Data Transfer Launching Agents initiate data transfer jobs as requested by the BigData Express scheduler. Typically, Data Transfer Launching Agents launch 3rd party data transfers between DTNs using X.509 certificates on behalf of users. Data transfer launching agent features an extensible plugin framework that is capable of supporting different data transfer protocols, such as mdtmFTP, GridFTP, and XrootD.

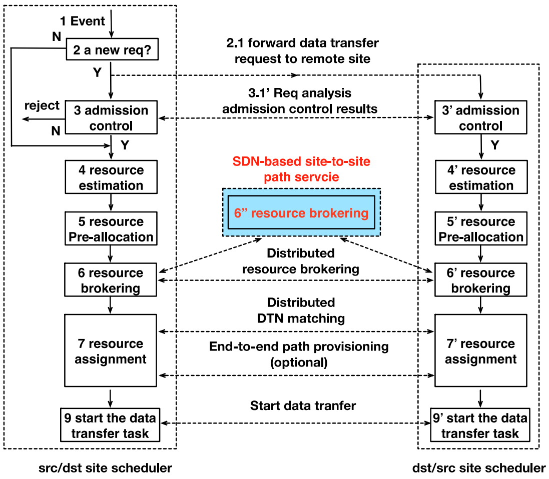
1 Event
N
2 a new req?
Y
reject
3 admission control
N
Y
4 resource estimation
5 resource Pre-allocation
6 resource brokering
7 resource assignment
9 start the data transfer task
src/dst site scheduler
2.1 forward data transfer request to remote site
3.1' Req analysis admission control results
SDN-based site-to-site path servcie
6'' resource brokering
Distributed resource brokering
Distributed DTN matching
End-to-end path provisioning (optional)
Start data tranfer
3' admission control
Y
4' resource estimation
5' resource Pre-allocation
6' resource brokering
7' resource assignment
9' start the data transfer task
dst/src site scheduler

Figure 3 The scheduler operations

The BigData Express scheduler implements a time-constraint-based scheduling mechanism to schedule resources for data transfer tasks. Each resource is estimated, calculated, and converted into a rate that can be apportioned into data transfer tasks. The scheduler assigns rates for data transfer tasks in the following order of priority: real-time data transfer tasks → deadline-bound data transfer tasks → background data transfer tasks. Rates will be assigned to real-time data transfer tasks on an as-needed basis. The scheduler schedules and assigns resources for data transfer tasks in two modes: (a) in an event-driven mode, when a data transfer request arrives, or a data transfer task is completed; and (b) in a periodic mode to periodically reschedule and reassign resources for data transfer tasks to adapt rapidly to changing run-time environments. For extreme-scale data movement, a data transfer task may take days, or even longer. Except for real-time data transfer tasks, it would be naïve to make a one-shot reservation for a particular data transfer task throughout its duration because run-time environments (e.g., traffic load, network, and storage conditions) will change with time. Therefore, the scheduler runs in a periodic mode to reschedule and reassign resources for a data transfer task as it progresses based on the deadline and the remaining data size. On either an event-driven or periodic basis, the scheduler performs the following tasks (Figure 3):

- Admission control. A data transfer job that cannot satisfy its time constraints without violating others will not be admitted.
- Resource estimation and calculation. Estimating and calculating the local site resources that can be assigned to data transfer tasks.
- Resource pre-allocation. Implementing a time-constraint-based resource allocation mechanism to pre-allocate the local site resources—in terms of rates—to data transfer tasks.
- Resource brokering. Implementing the resource brokering mechanism to coordinate rate pre-allocation across sites for a particular data transfer task, as well as determining the coordinated end-to-end data transfer rate for the task.
- Resource assignment. Assigning the local site resources to data transfer tasks based on their coordinated end-to-end data transfer rates, and establishing end-to-end paths between DTNs if required.

When the scheduler reshuffles resources for data transfer tasks, deadline-bound and background data transfer tasks may be temporarily suspended and then later resumed.

A BigData Express site can also allocate some resources to allow data transfer with non-BigData Express sites, but in a best effort manner.

B. *A High-performance Data Transfer Engine*

mdtmFTP is BigData Express' default data transfer engine. It offers high-performance data transfer capabilities.

mdtmFTP achieves high performance through several key mechanisms. First, mdtmFTP adopts a pipelined I/O centric design. A data transfer task is carried out in a pipelined manner across multiple cores. Dedicated I/O threads are spawned to perform network and disk I/O operations in parallel. Second, mdtmFTP utilizes the MDTM middleware services to make optimal use of the underlying multicore system. Finally, mdtmFTP implements a large virtual file mechanism to address the Lots of Small Files (LOSF) problem. Evaluations have shown that mdtmFTP achieves higher performance than data transfer tools such as GridFTP, FDT, and BBCP.

mdtmFTP supports third-party data transfer. It also supports GSI-based security. Figure 4 illustrates a BigData Express data transfer example. A Data Transfer Launching Agent launches a third-party data transfer between two DTNs using X.509 certificates.

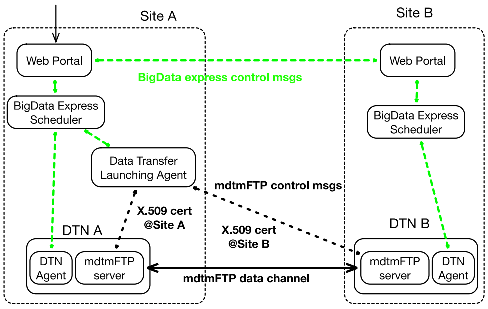

Figure 4 BigData Express launches data transfer jobs

C. On-Demand Provisioning of End-to-End Network Paths with Guaranteed QoS

BigData Express intelligently programs network at run-time to suit data transfer requirements. It dynamically provisions end-to-end network paths with guaranteed QoS between DTNs. An end-to-end network path typically consists of LAN and WAN segments. In BigData Express end-to-end data transfer model, LAN segments are provisioned and guaranteed by AmoebaNet, while WAN segments are provisioned through on-demand WAN path services such as ESnet OSCARS, or Internet2 AL2S to provide paths between the data source and destination sites.

AmoebaNet applies SDN technologies to provide "Application-aware" network service services in the local network environment. It offers several capabilities to support BigData Express operations. To support network programmability, AmoebaNet provides a rich set of network programming primitives to allow BigData Express to program the local area network at run-time. To support QoS guarantees, AmoebaNet provides two classes of services, *priority* and *best-effort*. Priority traffic flows are typically specified with designated rates or bandwidth. AmoebaNet uses QoS queues to differentiate priority and best-effort traffic at each SDN switch. Priority traffic is transmitted first, but metered to enforce rate control. In addition, AmoebaNet supports QoS-based routing and path selection. Finally, AmoebaNet supports fine-grained control of network traffic.

WAN QoS can be provisioned and guaranteed by utilizing ESnet OSCARS, or Internet2 AL2S to reserve bandwidths between Service Termination Points (STPs), where AmoebaNet services end.

Typically, AmoebaNet gateways (GWs) are either logically, or physically connected to WAN STPs. VLAN popping, pushing, and/or swapping operations are performed at AmoebaNet gateways to concatenate WAN and LAN segments.

As illustrated in Figure 5, BigData Express typically performs the following operations to provision an end-to-end network path:

1) Estimate and calculate the DTN-to-DTN traffic matrix, and the related QoS requirements (e.g. throughput, delay).
2) Negotiate and broker network resources to determine the end-to-end rate for the path.
3) Call ESnet OSCARS or Internet2 AL2S circuit service to set up a site-to-site WAN path.
4) Call AmoebaNet at each site to program and configure the LAN paths.
5) Send PING traffic to verify a contiguous end-to-end network path has been successfully established.

A large data transfer job typically involves many DTNs, and a corresponding large number of data flows. To avoid the necessity of establishing many WAN paths between the source and destination sites, multiple LAN segments can be multiplexed/de-multiplexed to/from a single WAN path, which in turn is configured to support the aggregated bandwidth of its component paths. This strategy helps to reduce burden on WAN path services.

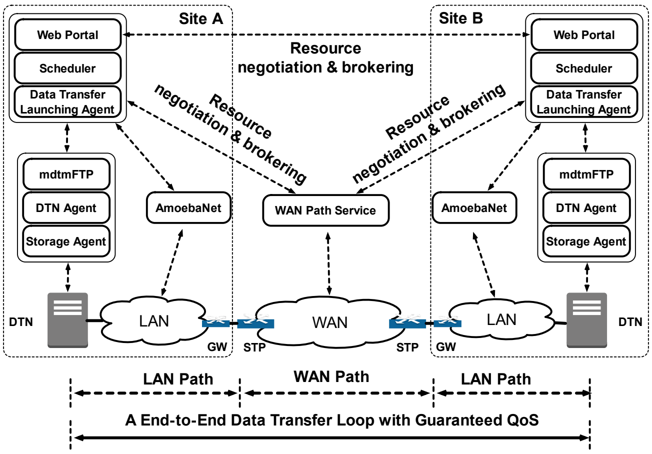
Site A
Site B
Web Portal
Scheduler
Data Transfer Launching Agent
Resource negotiation & brokering
Resource negotiation & brokering
Resource negotiation & brokering
Web Portal
Scheduler
Data Transfer Launching Agent
mdtmFTP
DTN Agent
Storage Agent
AmoebaNet
WAN Path Service
AmoebaNet
mdtmFTP
DTN Agent
Storage Agent
DTN
LAN
WAN
LAN
DTN
GW
STP
STP
GW
LAN Path
WAN Path
LAN Path
A End-to-End Data Transfer Loop with Guaranteed QoS

Figure 5 Provisioning of end-to-end path with guaranteed QoS

D. Secuirty

BigData Express runs in secure environments. At each site, BigData Express systems run in trusted security zones protected by security appliances. All DTNs are secured by using X.509 certificates. All BigData Express sits use a common single-point sign-on service (CILogon) to obtain X.509 certificates for secure access to DTNs. In addition, each site publishes its public key so that different sites can establish trust. Communication channels between two sites are secured by HTTPS.

Users are authenticated and authorized to access BigData Express services. From a user's perspective, BigData Express provides two layers of security:

(1) A user must first use his/her username and password to login to a particular BigData Express web portal. Once login is successful, a user can manage data transfer tasks (submission, cancellation, and monitoring), or monitor the system/site status.
(2) Within a logged-in web portal, a user must further login to data transfer source and/or destination site(s) to obtain X.509 certificates for secure access to local DTNs. Once authenticated locally, the user can browse files, and/or launch data transfer tasks. With CILogon issued X.509 certificates, the BigData Express scheduler will request Data Transfer Launching Agents to launch data transfer tasks on behalf of the user.

E. Error handling

Faults are inevitable in BigData Express due to the scale and complexity of the system. BigData Express handles failures through redundancy and retries. For critical components, multiple instances will be launched to improve system reliability. A failed operation will be retried multiple times until the maximum retry limit is reached. When a failure can not be recovered, the event will be recorded and system administrator will be alerted.

All transferred data will be checksummed and validated. Checksum errors result in retransmission. Typically, a large data set is split into multiple smaller blocks. Only the block(s) with errors are retransferred.

IV. EXPERIMENTS

In this section, we conduct two experiments to evaluate BigData Express. First, we demonstrate BigData Express's high-performance data transfer capabilites through use of mdtmFTP, its default data transfer engine. Second, we run BigData Express transfers across a trans-Pacific SDN path to demonstrate its key features and capabilities, including time-constraint-based scheduling of data transfer tasks and on-demand provisioning of end-to-end paths with guaranteed QoS.

A. High-performance Data Transfer

We evaluated mdtmFTP locally at StarLight using high-performance DTNs. This testbed focuses on high performance data plane experiments, providing sufficient computing/IO resources. The topology of the testbed is shown in Figure 6. Two high-performance DTNs were connected to a 100GE switch. Their configurations are listed below:

	DTN1	DTN2
Hardware	2 x NUMA nodes 28 cores (Intel E5-2683 v3)	2 x NUMA nodes 24 cores (Intel E5-2687W v4)
	64GB MEM	24GB MEM
	1 x 100GE Mellanox ConnectX-4	1 x 100GE Mellanox ConnectX-4
	8 x NVMe Drives	8 x NVMe Drives
OS	Linux 4.4	Linux 4.4

Table 1 DTN configuration

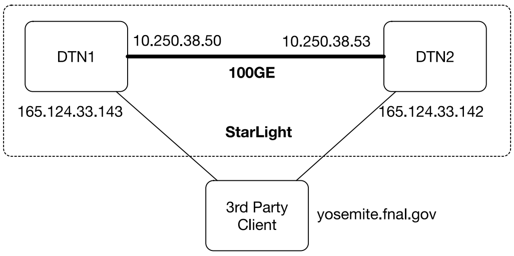

Figure 6 mdtmFTP Evaluation

Each DTN is equipped with eight NVMe drives, attached to a PCI-Express Gen 3x16 slot. The NIC in each DTN is attached to another x16 slot in the same NUMA node, forming a logical unit for high-performance transfer.

Each NVMe drive is formatted with the Ext4 file system and mounted to a folder "/data/nvme-*x*" in both DTNs. "*x*" ranges from 1 to 8. In DTN1, a 300GB file is created at each "/data/nvme-*x*" folder for data transfer.

Tools	Streams	Block size	Concurrency	TCP/IP parameter
GridFTP	-p 8	8MB	-cc 4	System config
mdtmfTP	-p 8	8MB	N/A	System config

Table 2 Testing configuration

Scenario	Data transfer
1-job	1. DTN1:/data/nvme-1/300GB.txt → DTN2:/data/nvme-1/
2-jobs (parallel)	1. DTN1:/data/nvme-1/300GB.txt → DTN2:/data/nvme-1/
	2. DTN1:/data/nvme-2/300GB.txt → DTN2:/data/nvme-2/
4-jobs (parallel)	1. DTN1:/data/nvme-1/300GB.txt → DTN2:/data/nvme-1/
	2. DTN1:/data/nvme-2/300GB.txt → DTN2:/data/nvme-2/
	3. DTN1:/data/nvme-3/300GB.txt → DTN2:/data/nvme-3/
	4. DTN1:/data/nvme-4/300GB.txt → DTN2:/data/nvme-4/
8-jobs (parallel)	1. DTN1:/data/nvme-1/300GB.txt → DTN2:/data/nvme-1/
	2. DTN1:/data/nvme-2/300GB.txt → DTN2:/data/nvme-2/
	3. DTN1:/data/nvme-3/300GB.txt → DTN2:/data/nvme-3/
	4. DTN1:/data/nvme-4/300GB.txt → DTN2:/data/nvme-4/
	5. DTN1:/data/nvme-5/300GB.txt → DTN2:/data/nvme-5/
	6. DTN1:/data/nvme-6/300GB.txt → DTN2:/data/nvme-6/
	7. DTN1:/data/nvme-7/300GB.txt → DTN2:/data/nvme-7/
	8. DTN1:/data/nvme-8/300GB.txt → DTN2:/data/nvme-8/

Table 3 Data transfer scenarios

In our evaluation, mdtmFTP was compared with GridFTP. For fair comparisons, all the tools were configured with the same parameters—I/O block size and the number of parallel streams (Table 2). We launched third party data transfers. A remote client located at Fermilab initiated the transfer task(s) but the data was transferred from DTN1 to DTN2. Four data transfer scenarios were evaluated, as listed in Table 3. Throughput was used as the performance metric. Each scenario was run multiple times, and the average was calculated.

The results are listed in Table 4. It can be seen that mdtmFTP achieved roughly twice the throughput of GridFTP in the 1-job and 2-jobs scenarios. Even when the number of parallel jobs increased to 8, mdtmFTP was still approximately 50% faster than GridFTP. Because the NVMe drives and the NIC were installed in the same NUMA node at each DTN, we noticed that both DTNs' system buses were close to saturation in the 8-jobs scenario, which made it difficult to further increase performance.

We also varied paramters in the evaluation. We noticed that following results:

- When the number of parallel streams was larger than 4, the performance did not change much for both tools.
- The *concurrency* parameter for GridFTP was difficut to configure. When the paramter was set to a small number (e.g. 1 or 2), the throughput was poor due to lack of thread parallelism in the 1-job and 2-jobs secnarios. However, when the parameter was to set a larger nubmer (e.g, 6 or 8), too many threads would be created in the 8-jobs scenarios, leading to significant performance degradation.

The evaluation showed that mdmFTP achieves significantly better performance than GridFTP.

	1-job	2-jobs	4-jobs	8-jobs
GridFTP	6.2Gbps	12.24Gbps	20.35Gbps	28.32Gbps
mdtmFTP	13.27Gbps	23.80Gbps	28.35Gbps	43.94Gbps

Table 4 mdtmFTP vs. GridFTP

B. *Field Evaluation of BigData Express*

In this experiment, we evaluated BigData Express cross a trans-Pacific SDN path (Figure 7) to demonstrate its key features and capabilities, including time-constraint-based scheduling of data transfer tasks and on-demand provisioning of end-to-end paths with guaranteed QoS.

The testbed consists of two administratively independent sites – FNAL site and KISTI site, with a dedicated layer-2 WAN circuit that connects the sites.

FNAL site:

- DTNs: bde1, bde2, and bde3. Each DTN was equipped with an Intel NVMe drive that is formated with the Ext4 file system, and a 40GE Mellanox NIC.
- SDN switches: Pica8 P5101 (running PicOS)
- An ONOS-based SDN controller

KISTI site:

- DTNs: dtn2 and dtn3. Each DTN was equipped with an Intel NVMe drive that is formatted with the Ext4 file system, and a 10GE NIC.
- SDN switches: HP Z91000 (running PicOS).
- An ONOS-based SDN controller

The trans-Pacific path was provided by a dedicated layer-2 WAN circuit, which runs across ESnet, StarLight, and KREONET. In particular, the ESnet segment, an OSCARS circuit, can be dynamically set up and torn down using ESnet NSI circuit services.

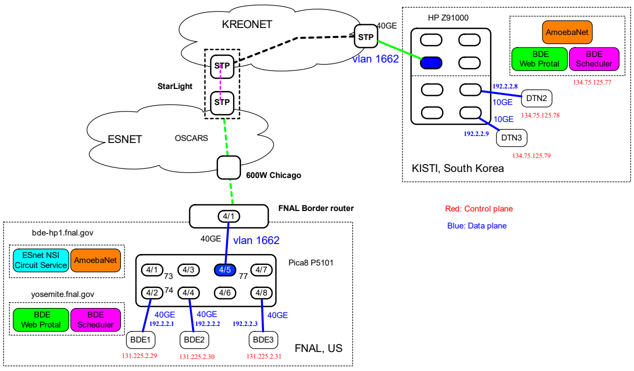

Figure 7 A Cross-Pacific SDN testbed

BigData Express software ran at both sites. Users can access BigData Express services from either https://yosemite.fnal.gov:5000 (BigData Express Web Portal @FNAL), or https://134.75.125.77:2888/ (BigData Express Web Portal @KISTI), respectively.

In the evaluation, three data transfer tasks were submitted at https://yosemite.fnal.gov:5000:

- Task 1, which is to move a 300GB data set from dtn3@KISTI to bde2@FNAL. The task was submitted at 60th second, with a deadline of 1800 seconds.
- Task 2, which is to move a 200GB data set from dtn2@KISTI to bde1@FNAL. The task was submitted at 200th second, with a deadline of 300 seconds.
- Task 3, which is to move a 800GB data from dtn3@KISTI to bde1@FNAL. It was submitted at time 0 to provide background best-effort traffic, without an explicit deadline.

The evaluation results are illustrated in Figure 8. Task 3 was submitted at time 0 as background traffic. Because there were no other data transfer tasks that competed for resources between 0s and 60s, Task 3's transfer rate reached as high as ~5Gpbs.

Task 1 was submitted @60s. Based on the resource availability and the job priority, Task 1 was admitted and launched with an initial rate of 4Gbps. Because the BigData Express scheduler runs in a periodic mode to reschedule and reassign resources for data transfer tasks to adapt rapidly to changing run-time environments, Task 3 was throttled to a lower rate of 2Gbps @80s while Task 1's rate was increased to ~6Gpbs @95s. Our results verify that BigData Express prioritizes deadline-bound data transfer task(s) over best-effort data transfer task(s).

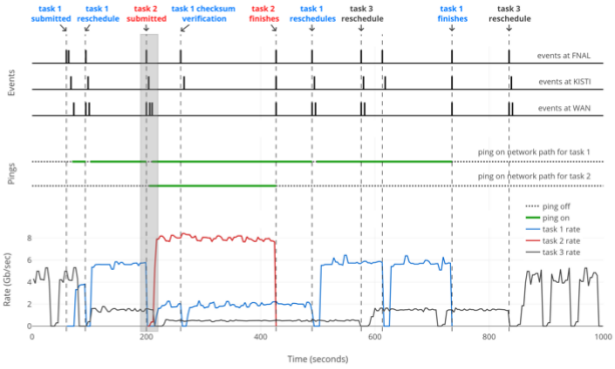

Figure 8 BigData Express data transfer evaluation

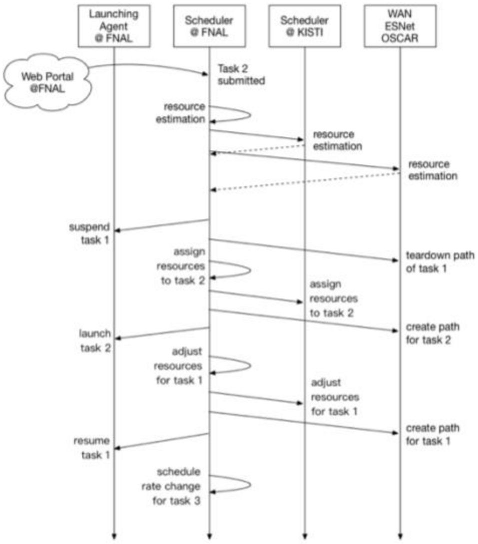

Figure 9 Event sequence diagram @period [200s, 220s]

Task 2 was submitted @200s, which was to move a 200 GB data set from KIST to FNAL within 300 seconds. Therefore, it required a minimum rate of ~5.4Gbps to meet the deadline. From Figure 8, it can be seen that the transfer rate for Task 2 was assigned to as high as ~8Gbps, while the rates for Task 1 and 3 were lowered to ~2Gbps and ~0.5Gbps, respectively. After Task 2 was completed, the rates for Task 1 and 3 were adjusted to higher rates again. Because BigData Express prioritizes requests from different users to satisfy as many time constraints as possible, both Task 1 and 2 met their deadlines. Figure 9 illustrates an event sequence diagram of BigData Express for the period from 200s to 220s when Task 2 was submitted.

An end-to-end network path is dynamically established for each data transfer job. In this experiment, each end-to-end network path ran across three domains – FNAL, ESnet-KREONET WAN, and KISTI. Therefore, each end-to-end network path consists of a LAN segment at FNAL, a WAN segment at ESnet-KREONET, and another LAN segment at KISTI. We used the spoke-hub distribution model to set up these paths. A single layer-2 point-to-point circuit with an aggregated bandwidth of 20Gbps was established between sites FNAL and KISTI. AmoebaNet was responsible for setting up the local LAN path segments between DTNs and gateway at FNAL and KISTI, respectively. All inter-site traffic is multiplexed/de-multiplexed to/from a single point-to-point layer 2 circuit. Figure 8 also shows when the network paths for Task 1 and 2 were established and pingable.

The following scripts illustrate some of BigData Express commands. List 1 illustrates the NSI commands for setting up the ESnet WAN path. List 2 shows an AmoebaNet command for set up a LAN segment at FNAL.

```
onsa reserveprovision \
-g 4b2beff5-11d8-4453-b255-b8a277e4e351 \
-d es.net:2013::star-cr5:6_1_1:+?vlan=1662 \
-s es.net:2013::chic-cr5:3_2_1:+?vlan=1662 \
-b 20000 \
-a 2018-08-9T20:00:00 \
-e 2019-07-30T20:20:20 \
-u https://nsi-aggr-west.es.net:443/nsi-v2/ConnectionServiceProvider \
-p es.net:2013:nsa:nsi-aggr-west \
-r es.net:2013:nsa:nsi-requester \
-h 131.225.2.17 \
-o 8443 \
-l ./wenji.crt \
-k ./wenji.key \
-i ./etc/ssl/certs/ \
-x
```

List 1 The script for setting up the ESnet WAN path

```
{
   "cmd" : "reserve_request",
   "dtns" :
   {
      "dstId" : "a4:bf:01:47:e9:dd",
      "dstIp" : "192.2.2.8",
      "dstMac" : "ec:0d:9a:17:45:c0",
      "oscarsVlanId" : "1662",
      "rate" : "8000.000000",
      "routeType" : "h2g",
      "srcId" : "0c:c4:7a:ab:63:7e",
      "srcIp" : "192.2.2.1",
      "srcMac" : "68:05:ca:2e:77:18",
      "trafficType" : "1",
      "vlanId" : "2"
   },
   "end" : "2018-12-30 16:35",
   "start" : "direct"
}
```

List 2 An AmoebaNet command for setting up an LAN segment@FNAL

V. Conclusion

The emergence of distributed, extreme-scale science applications is generating significant challenges regarding data transfer. The data transfer challenges of the extreme-scale era are typically characterized by two relevant dimensions: *high-performance challenges* and *time-constraint challenges.* In this paper we have shown how the BigData Express project addresses these challenges. BigData Express seeks to provide a schedulable, predictable, and high-performance data transfer service for big data science.

The BigData Express software is being deployed and evaluated at multiple research institutions, which include UMD, StarLight, FNAL, KISTI, KSTAR, SURFnet, and Ciena. Meanwhile, the BigData Express research team is collaborating with the StarLight International/national Communications Exchange Facility to deploy BigData Express at various research platforms, including Pacific Research Platform, National Research Platform, and Global Research Platform. It is envisioned that we are working toward building a high-performance data transfer service federation for big data science.

References

[1] "Synergistic Challenges in Data-Intensive Science and Exascale Computing", DOE ASCR Data Subcommittee Report 2013.

[2] Eli Dart, Mary Hester, Jason Zurawski, "Basic Energy Sciences Network Requirements Review - Final Report 2014", ESnet Network Requirements Review, September 2014, LBNL 6998E

[3] Eli Dart, Mary Hester, Jason Zurawski, "Fusion Energy Sciences Network Requirements Review - Final Report 2014", ESnet Network Requirements Review, August 2014, LBNL 6975E

[4] Eli Dart, Mary Hester, Jason Zurawski, Editors, "High Energy Physics and Nuclear Physics Network Requirements - Final Report", ESnet Network Requirements Workshop, August 2013, LBNL 6642E

[5] Eli Dart, Brian Tierney, Editors, "Biological and Environmental Research Network Requirements Workshop, November 2012 - Final Report"", November 29, 2012, LBNL LBNL-6395E

[6] David Asner, Eli Dart, and Takanori Hara, "Belle-II Experiment Network Requirements", October 2012, LBNL LBNL-6268E

[7] Eli Dart, Brian Tierney, editors, "Advanced Scientific Computing Research Network Requirements Review, October 2012 - Final Report", ESnet Network Requirements Review, October 4, 2012, LBNL LBNL-6109E

[8] W. Allcock, J. Bester, J. Bresnahan, A. Chervenak, L. Liming, and S. Tuecke, "GridFTP: Protocol Extension to FTP for the Grid," Grid Forum Internet-Draft, Mar. 2001.

[9] B. Allcock, J. Bresnahan, R. Kettimuthu, M. Link, C. Dumitrescu, I. Raicu and I. Foster, "The Globus Striped GridFTP Framework and Server," SC'2005, 2005.

[10] BBCP, http://www.slac.stanford.edu/~abh/bbcp/

[11] T. A. Barrass, et al., "Software Agents in Data and Workflow Management," in Computing in High Energy and Nuclear Physics (CHEP) 2004, Interlaken, Switzerland, 2004.

[12] J. Rehn, T. Barrass, D. Bonacorsi, J. Hernandez, I. Semeniouk, L. Tuura, and Y. Wu, "PhEDEx high-throughput data transfer management system," in Computing in High Energy and Nuclear Physics (CHEP) 2006, Mumbai, India, 2006.

[13] http://www.lsc-group.phys.uwm.edu/LDR/

[14] https://www.globus.org/

[15] Liang Zhang, Wenji Wu, Phil DeMar, Eric Pouyoul: mdtmFTP and its evaluation on ESNET SDN testbed. Future Generation Comp. Syst. 79: 199-204 (2018)

[16] Han, Huaizhong, et al. "Multi-path tcp: a joint congestion control and routing scheme to exploit path diversity in the internet." IEEE/ACM Transactions on Networking (TON) 14.6 (2006): 1260-1271.

[17] Wang, Bing, et al. "Application-layer multipath data transfer via TCP: schemes and performance tradeoffs." Performance Evaluation 64.9 (2007): 965-977.

[18] Iyengar, Janardhan R., Paul D. Amer, and Randall Stewart. “Concurrent multipath transfer using SCTP multihoming over independent end-to-end paths.” Networking, IEEE/ACM Transactions on 14.5 (2006): 951-964.

[19] Gunter, Dan, et al. “Exploiting network parallelism for improving data transfer performance.” High Performance Computing, Networking, Storage and Analysis (SCC), 2012 SC Companion:. IEEE, 2012.

[20] Dart, Eli, et al. “The science dmz: A network design pattern for data-intensive science.” Scientific Programming 22.2 (2014): 173-185.

[21] Guok, Chin, ESnet Network Engineer, and David Robertson. “ESnet On-Demand Secure Circuits and Advance Reservation System (OSCARS).” Internet2 Joint Techs Workshop, Salt Lake City, Utah. 2005.

[22] https://www.internet2.edu/products-services/advanced-networking/layer-2-services/

[23] http://mqtt.org

[24] Y. Gu, and R. Grossman, “UDT: UDP-based Data Transfer for High-Speed Wide Area Networks,” Computer Networks (Elsevier). Volume 51, Issue 7. May 2007.

[25] C. Jin, D.X. Wei, and S.H. Low, “FAST TCP: Motivation, Architecture, Algorithms, Performance,” in Proc. IEEE Infocom, 2004.

[26] Injong Rhee and Lisong Xu, CUBIC: A New TCP-Friendly High-Speed TCP Variants. PFLDnet 2005, Feb. 2005, Lyon, France.

[27] McKeown, Nick. “Software-defined networking.” INFOCOM keynote talk 17.2 (2009): 30-32.

[28] Shenker, Scott, et al. “The future of networking, and the past of protocols,” Open Networking Summit (2011).

[29] Al-Fares, Mohammad, et al. “Hedera: Dynamic Flow Scheduling for Data Center Networks.” NSDI. Vol. 10. 2010.

[30] Wang, Richard, Dana Butnariu, and Jennifer Rexford. “OpenFlow-based server load balancing gone wild.” (2011).

[31] McKeown, Nick, et al. “OpenFlow: enabling innovation in campus networks.” ACM SIGCOMM Computer Communication Review 38.2 (2008): 69-74.

[32] Hong, Chi-Yao, et al. “Achieving high utilization with software-driven WAN.” ACM SIGCOMM Computer Communication Review. Vol. 43. No. 4. ACM, 2013.

[33] Jain, Sushant, et al. “B4: Experience with a globally-deployed software defined WAN.” ACM SIGCOMM Computer Communication Review. Vol. 43. No. 4. ACM, 2013.

[34] Jim Basney, Terry Fleury, and Jeff Gaynor, "CILogon: A Federated X.509 Certification Authority for CyberInfrastructure Logon," Concurrency and Computation: Practice and Experience, Volume 26, Issue 13, pages 2225-2239, September 2014.

[35] S. A. R. Shah, W. Wu, Q. Lu, L. Zhang, S. Sasidharan, P. DeMar, C. Guok, J. Macauley, E. Pouyoul, J. Kim, S. Noh: AmoebaNet: An SDN-enabled network service for big data science. J. Network and Computer Applications 119: 70-82 (2018).

[36] https://rucio.cern.ch/