A Trial Deployment of a Reliable Network-Multicast Application across Internet2

Yuanlong Tan[†], Malathi Veeraraghavan[†], Hwajung Lee[‡], Steve Emmerson[§], Jack Davidson[†]
[†]University of Virginia, Charlottesville, VA, USA, {yt4xb,jwd}@virginia.edu
[‡]Radford Univeristy, Radford, VA, USA, hlee3@radford.edu
[§]University Corporation for Atmospheric Research, Boulder, CO, emmerson@ucar.edu

***Abstract*—A continuing trend in many scientific disciplines is the growth in the volume of data collected by scientific instruments and the desire to rapidly and efficiently distribute this data to the scientific community. Transferring these large data sets to a geographically distributed research community consumes significant network bandwidth. As both the data volume and number of subscribers grows, reliable network multicast is a promising approach to reduce the rate of growth of the bandwidth needed to support efficient data distribution. In prior work, we identified a need for reliable network multicast: scientists engaged in atmospheric research subscribing to meteorological file-streams. Specifically, the University Cooperation Atmospheric Research (UCAR) uses the Local Data Manager (LDM) to disseminate data. This work describes a trial deployment of a multicast-enabled LDM, in which eight university campuses are connected via corresponding regional Research-and-Education Networks (RENs) and Internet2. Using this deployment, we evaluated the new version of LDM, LDM7, which uses network multicast with a reliable transport protocol, and leverages Layer-2 (L2) multipoint Virtual LAN (VLAN/MPLS). A performance monitoring system was deployed to collect real-time performance of LDM7, which showed that our proof-of-concept prototype worked significantly better than the current production LDM, LDM6, in two ways: (i) LDM7 can distribute file streams faster than LDM6. With six subscribers, an almost 22-fold improvement was observed with LDM7 at 100 Mbps. And (ii) to achieve a similar performance, LDM7 significantly reduces the need for bandwidth, which reduced the bandwidth requirement by about 90% over LDM6 to achieve 20 Mbps average throughput across four subscribers.**

***Index Terms*—File-Stream Distribution; Software Defined Network; Multicast; Control-Plane Protocol**

I. Introduction

A continuing trend in many scientific disciplines is the growth in the volume of data collected by scientific instruments and the desire to rapidly and efficiently distribute this data to the scientific community. Transferring these large data sets to a geographically distributed research community consumes significant network bandwidth. For example, in Unidata's Internet Data Distribution (IDD) system, the University Corporation for Atmospheric Research (UCAR) uses an application, called Local Data Manager (LDM), to distribute 30 different types of meteorological data (e.g., surface observations, radar data, satellite imagery, wind profiler data, lightning data, and high-resolution computer-model output) to 574 hosts in 217 domains. Approximately 420,000 data products[1] comprising 50 gigabytes are generated each hour. The volume of data and number of subscribers have both been increasing. For example, the GOES-16 weather satellite that recently came online has an operational bandwidth that is 14 times greater than the previous-generation satellite.

A simultaneous occurrence of two events, an application top-down push for reliable network multicast, and a bottom-up technological advance in the form of Software-Defined Networks (SDN), led to the work presented here. The current LDM, LDM6, uses a separate unicast TCP connection from a publisher to each of its subscribers, which explains why both the sender computing power and network bandwidth requirements have been growing rapidly in the IDD system. While UCAR receives 50 GB/hr from various input sources, it transmits, on average, about 2.34 TB/hr from its sending compute cluster. A network multicast solution would alleviate the demand of both computing power and network bandwidth. Thus, LDM and IDD provide the *top-down application motivation* to revisit the use of network multicast.

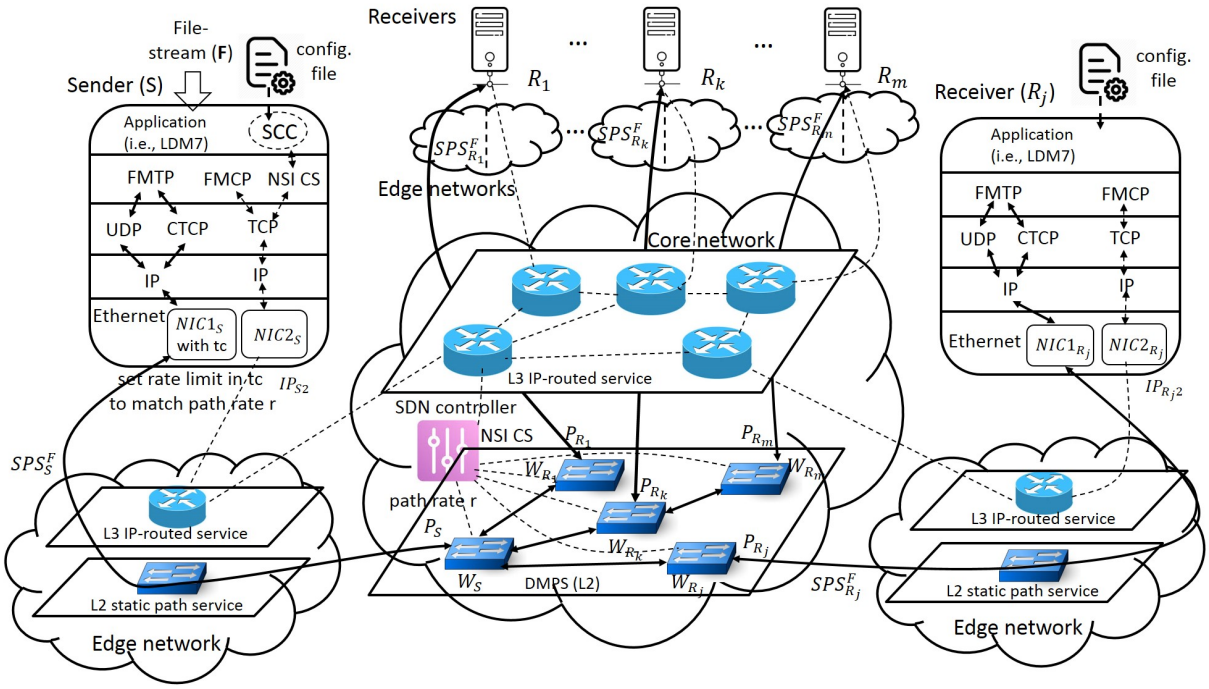
The main wide-area, inter-domain network multicast solution that was designed and implemented, but not broadly used, is IP multicast. Distributed routing solutions, with their associated protocols, were developed to support IP multicast. But the complexity of these solutions is one of the reasons cited for questioning its feasibility [1]. The use of centralized controllers in SDN greatly simplifies the control-plane actions required to configure forwarding table in switches for multicast flows. Thus, SDN provided us the *bottom-up technological-advance motivation* to revisit the use of network multicast.

In prior work [2], [3], we described a cross-layer Multicast-Push Unicast-Pull (MPUP) architecture for supporting reliable file-stream multicasting, which has the following features: (i) Layer-2 (L2) multipoint Virtual LAN (VLAN/MPLS) service, and (ii) File Multicast Transport Protocol (FMTP), a reliable transport-layer protocol for delivering file-streams, which uses UDP and Circuit TCP (CTCP) over the L2 network service. The proposed architecture was evaluated on the NSF GENI [4] and Chameleon testbeds [5].

In this paper, we describe a trial deployment of a network multicast solution that addresses the need to distribute large volumes of scientific data to subscribers reliably and efficiently. The deployment involved eight universities connected

[1]The terms "file" are "data product" are used interchangeably.

978-1-6654-0454-9/20/$31.00 ©2020 IEEE
DOI 10.1109/INDIS51933.2020.00008



SCC: SDN Controller Client; FMTP: File Multicast Transport Protocol; FMCP: File Multicast Control Protocol;
NSI CS: Network Service Interface Connection Service; tc: traffic control; SPS: Static Path Segment;
L3: Layer-3 (IP header-based forwarding); L2: Layer-2 (VLAN/MPLS); DMPS: Dynamic Multipoint Path Service;

Fig. 1: Dynamic Reliable File-Stream Multicast (DRFSM) service architecture

via corresponding regional Research-and-Education Networks (RENs) and Internet2, the US-wide REN. To accommodate this multi-domain WAN usage, we developed a new Dynamic Reliable File-Stream Multicast (DRFSM) service architecture. This architecture requires a core network that supports Dynamic Multipoint Path Service (DMPS), which is available in some SDNs, e.g., Internet2.

This paper describes three contributions:

Contributions:

1) A design and implementation of a DRFSM service,
2) A design and implementation of a performance monitoring system and addition of metrics for a rigorous evaluation, and
3) A trial deployment of this modified LDM, called LDM7, involving eight university campuses with experimental results comparing the performance of LDM7 to LDM6, the current meteorological data distribution application.

This paper has the following organization: Section II describes the DRFSM architecture, reviews the transport-layer protocols, and describes the implementation of the DRFSM control-plane software and integration of these components with LDM7. Section III describes the design and implementation of a new LDM7 performance monitor and defines metrics used for our evaluation. Our trial deployment in a production WAN setting is described in Section IV. Section V presents results of our experimental evaluation on this deployment. Section VI provides background material and reviews recent related work, and Section VII concludes the paper.

II. Dynamic Reliable File-Stream Multicast Service

Section II-A illustrates and reviews the DRFSM architecture in a bottom-up manner. Section II-B describes our implementation of DRFSM control-plane and integration of these modules into LDM7.

A. Architectural description

Fig. 1 illustrates our proposed DRFSM service architecture. This architecture requires two types of network services, L2 path-based service for data-plane and Layer-3 (L3) IP-routed service for control-plane. The reason the architecture requires a *path-based network service* is that, in a multi-receiver context, sequenced delivery and rate guarantees simplify the key transport-layer functions of error control, flow control and congestion control. With sequenced delivery, a receiver can assume that a packet was dropped when it receives an out-of-sequence block, and then send a Negative ACKnowledgment (NACK) requesting block retransmission, which simplifies the error control. With rate guarantees on the paths from the sender to receivers, flow control and congestion control are handled by receivers agreeing to handle packets at the fixed rate of the multipoint network path in the control-plane path setup phase.

The rate is selected by the sender based on the characteristics of the file-stream and the application latency requirements.

Networks: Our architecture model allows for senders and receivers to be connected to their edge networks, which, in turn, are interconnected by a DMPS network. This more-complex model with multiple networks/domains is required, because the DRFSM service is proposed for WAN usage rather than for datacenter or enterprise network usage. The DRFSM service architecture assumes that:

- *Edge networks* offer:
  1) L3 IP-routed service, and
  2) L2 static path service, i.e., point-to-point path-based service with static provisioning capability,
- *Core network* offers:
  1) L3 IP-routed service, and
  2) DMPS, i.e., multipoint path service with dynamic control.

The L3 IP-routed service is used for exchanging control-plane messages, such as, subscription requests, and SDN controller signals. The L2 path-based service is used for disseminating scientific data. While Fig. 1 shows different routers and switches in the L3 network service and L2 network service, in practice, a single switch located in each Point-of-Presence (PoP) can provide both types of services. For example, most ISPs today deploy equipment, such as Juniper MX 960, that support both capabilities: (i) IP-forwarding, referred to as L3, for IP-routed service, and (ii) VLAN and MPLS forwarding, referred to as L2, for path-based service. To support dynamic provisioning, a SDN controller is required in the core network.

As shown in Fig. 1, Static Path Segments (SPSs) are provisioned from the sending and receiving hosts through edge networks to the DMPS core-network switch ports. The sender then uses its SDN Controller Clients (SCC) to send signaling messages to the core-network SDN controller to request the dynamic configuration of connecting/disconnecting SPSs in DMPS network. As shown in Fig. 1, a path rate r can be specified in the setup phase when the Network Service Interface-Connection Service (NSI-CS) signaling is used to send a request to the SDN controller. This model allows for the practical consideration that not all edge and core network providers will simultaneously start offering DMPS. Instead even if one core network offers DMPS, as illustrated in Fig. 1, end hosts can start using this service by leveraging the static path services of edge networks. For example, the U.S.-wide REN, Internet2, deploys a network that offers L3 service and DMPS with the Open Exchange Software Suit (OESS) SDN controller. Regional RENs in the U.S. offer both L3 IP-routed and static path services required of edge networks.

Network Interface Cards (NICs): Fig. 1 shows that sending and receiving hosts have two NICs: NIC1 connected to the path-service switch of the host's edge network, and NIC2 connected to the IP-forwarded router of the host's edge network. In practice, it is quite common for high-end servers to have two NICs, but it is also feasible to use just a single NIC connected to an edge-network switch, and provision multiple VLANs on that single NIC, with one VLAN configured to handle datagram-service packets, and the remaining VLANs configured to feed into different multipoint paths, one for each file-stream. Therefore, our model allows for both physically separate or logically separate NICs.

Given that DRFSM service is using a rate-guaranteed path inside the network for simplification of the transport-layer functions in the multi-receiver context, the rate at which packets are transmitted by the sender NIC should be limited to the rate of the multipoint path. The sender NIC1, as shown in Fig. 1, is configured using the Linux traffic control (`tc`) utility to send packets at rate r matched to the rate used for the multipoint path set up via the SDN controller. For example, the multipoint path and `tc` rate can be set to 500 Mbps when the NIC speed is 10 Gbps if 500 Mbps is sufficient to serve out files arriving into the sender for a particular file-stream.

Addressing, Routing and Configuration: Distributed intra- and inter-domain *routing protocols* are assumed to be deployed for the support of L3 service. For example, Open Shortest Path First (OSPF) [6] and Border Gateway Protocol (BGP) [7] allow the edge- and core-network IP routers to obtain address reachability information for NIC2, shown in Fig. 1, and create routing table entries based on their public IP addresses to enable IP-packet forwarding. For path-based networking services, no such distributed routing is assumed. Instead DRFSM assumes that publishers and subscribers should be configured with information required for routing and signaling.

TABLE I: Configuration file entries for file-stream **F**

Sender S	
(W_S, P_S)	a core-network switch and port to which the sender's SPSs are provisioned
$NIC1_S$	an identifier of the NIC1 at the sender
$SPS_S^{\mathbf{F}}$	an identifier of the SPS provisioned between the sender and the DMPS core network for **F**
$r^{\mathbf{F}}$	a multipoint-path rate and `tc` sending rate for **F**
$(IP_{mr}^{\mathbf{F}}, N_{mr}^{\mathbf{F}})$	a multi-receiver private IP address and UDP port number used for multicast of **F**
$(IP_{NIC1_S}^{\mathbf{F}}, M^{\mathbf{F}})$	a 32-bit private unicast IP address and netmask assigned to the SPS connected logical interface associated with sender's NIC1 for **F**
Receiver $R_j, 1 \leq j \leq m$	
(W_{R_j}, P_{R_j})	a core-network switch and port to which R_j's SPSs are provisioned
$NIC1_{R_j}$	an identifier of the NIC used for static path service at receiver R_j
$SPS_{R_j}^{\mathbf{F}}$	an identifier of SPS provisioned between R_j and the DMPS core network for **F**
$(IP_{S2}^{\mathbf{F}}, N_S^{\mathbf{F}})$	a public IP address assigned to NIC2 of the file-stream sender, along with a TCP port number

Table I shows the parameters used by DRFSM for performing dynamic multipoint path control operations for one file-stream. Some of these parameters are per-host (sender/receiver), while others are per-file-stream. The per-host parameters are: (i) the DMPS core-network switch and port to which SPSs are provisioned from each host's NIC1, and (ii) an identifier for the NIC used for static path service (first two rows of Table I for the sender and receiver). Per-file-stream parameters are: (i) an SPS identifier, (ii) a multipoint path/sending rate, and (iii) an IP address, transport-

layer port number, and netmask (remaining rows). Most of the parameters listed in Table I are illustrated in Fig. 1.

SPSs are provisioned from the sender to its DMPS core-network switch port for each file-stream that it multicasts, and SPSs are provisioned from each receiver to its DMPS core-network switch port for each of its subscribed file-streams. Examples of SPS are VLANs. For each file-stream, the sender configuration file has a rate-setting for the multipoint path, which is also used by `tc` as the packet sending rate.

Three types of IP addresses are used: (i) public IP addresses are assigned to NIC2 on each host, and these addresses are used for control-plane message exchanges such as subscription requests, (ii) private multi-receiver IP addresses are used as the destination IP address in datagrams that are multicast from the sender to all receivers on a multipoint path for a particular file-stream. While this address is not used for packet forwarding within the core or edge networks (packet forwarding in the path-based service is on L2 headers such as VLAN ID and MPLS label), this multi-receiver IP address needs to be configured in all receivers to accept IP packets sent via multicast because a UDP/IP socket is used by the sender; and (iii) private unicast IP addresses are assigned to each SPS connected logical interface associated with NIC1 of each host; these addresses are required for unicast retransmissions of packets lost by a receiver.

As seen in Table I, at sender, the second and third types of IP addresses are stored in the configuration file. The private unicast IP address is associated with a netmask so that the sender can assign IP addresses within the same subnet to the SPSs connected logical interfaces at all the receivers on the multipoint path. At receiver, for each file-stream, it requires the IP address of the corresponding sender's NIC2 public IP address, and TCP-port number for sending a subscription request. Other IP addresses required at the receiver are communicated via signaling from the sender.

Transport protocols: A reliable multicast transport protocol, File Multicast Transport Protocol (FMTP) [8], is used in the MPUP architecture. As illustrated in Fig. 1, FMTP uses UDP for multicast. The UDP datagrams are sent to a multicast IP address that is configured for the multipoint VLAN interface on NIC1 of the sender and all receivers. This multicast IP address does not need to be a Class-D IPv4 address because this address is only used at the hosts; all the transit switches perform packet forwarding on L2-header fields. For each file, FMTP sends a Begining-of-Product (BOP) message via L2 multicast to all receivers. Next, the FMTP sender divides the file into blocks large enough to fit in UDP datagrams, and multicasts these packets. Finally, FMTP multicasts an End-of-Product (EOP) message to all receivers.

Each FMTP receiver checks the FMTP packet header and detects missing blocks; since sequenced delivery is guaranteed on the L2 paths of the multipoint VLAN, missing blocks are detected from the block sequence number. Unicast Circuit TCP (CTCP) [9] connections, sent over the L2 paths from the sender to the receiver, are used for retransmissions. CTCP, designed for dedicated circuits, simply drops the congestion control functionality of TCP since bandwidth resources are reserved on circuits. The error and flow control functions of TCP are retained as packet losses from receiver-buffer overflows can occur. CTCP sends multiple segments without waiting for ACKs, limited only by the receiver's flow-window size. A unicast private IP address is assigned to the VLAN interface at NIC1 on the sender and at each receiver. This private IP address is used for the CTCP connections. If one or more data blocks are missing, the FMTP receiver sends a retransmission request to the FMTP sender, which retransmits the requested data blocks to just the requesting receiver. When all blocks are received correctly, the FTMP receiver signals the sender. A product index is carried in the FMTP header so that if a whole product is dropped, the receiver can request retransmission of all blocks of the missing product.

The FMTP sender sets a retransmission timer for each file. When this timer expires, the FMTP sender stops serving all pending and new retransmission requests and sends back rejections. This timer is required to prevent slow receivers from reducing multicast throughput for all other receivers. The presence of the FMTP sender retransmission timer necessitates an application-layer backstop mechanism so that applications with reliability requirements that are more stringent than achievable with the FMTP service can deliver missed products. For example, the LDM7 application uses an *LDM6-backstop mechanism*, in which LDM6 uses a TCP connection, established through the L3 IP-routed network, to send products that could not be delivered fully via FMTP.

B. Implementation

Three control-plane modules: (i) OESS client, (ii) File Multicast Control Protocol (FMCP), and (iii) `vlanUtil`, were implemented, and integrated into LDM7.

The *OESS client* serves as the SCC in our DRFSM service implementation. Specifically, the programmatic interface specified for the OESS server was used to implement this OESS client in Python. The OESS client implements the NSI CS client signaling modules to generate Provision Request, Edit Request and Release Request, and respond with ACKs to Reply messages. The SPS identifiers used in this implementation are 12-bit VLAN IDs defined in the IEEE 802.1Q standard.

The *FMCP* module implements the signaling between sender and receivers in the DRFSM service architecture to add or drop subscriptions.

The `vlanUtil` program executes the `ip (1)` utility to statically configure virtual interfaces (associated with VLANs) and assign private IP addresses to the virtual interfaces on NIC1 of the sender ($IP^{\mathbf{F}}_{NIC1_S}$). This program also executes `tc` to limit the sending rate and sender-buffer size at the $SPS^{\mathbf{F}}_S$ connected virtual interface on sender NIC1 ($NIC1_S$), using a combination of Hierarchical Token Bucket (HTB) and Bytes First In First Out (BFIFO) queueing disciplines. Two queues are defined, one for the UDP multicast packets, and the second for the FMTP block retransmissions on unicast CTCP connections. Bandwidth borrowing between these queues is allowed, which reduces the total required bandwidth. The

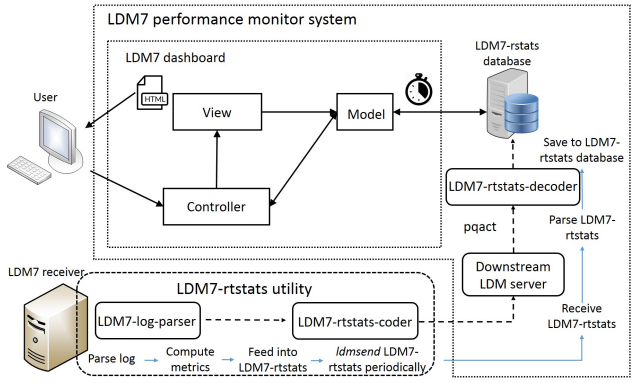


Fig. 2: LDM7 monitoring architecture

possibility of packet losses stemming from the use of too high a rate is avoided by setting the `ceil` parameter, which limits the maximum rate used to serve packets from a class. On the receiving side, the `vlanUtil` utility executes the `ip` command to set the private IP address received in the SR FMCP message to set up or down the virtual interface corresponding to the SPS (VLAN) on NIC1 ($IP^{\mathbf{F}}_{NIC1_{R_j}}$).

LDM7 was modified to integrate these three control-plane modules. The FMCP module was integrated into the original LDM6 subscription protocol. New LDM7 code was implemented to trigger OESS client actions to generate NSI CS messages based on received FMCP messages. Finally, LDM7 code invoked functions in the `vlanUtil` to configure virtual interfaces, IP addresses and `tc` (at sender only) based on the parameter values sent/received in the FMCP messages.

III. LDM7 Performance Monitoring System

A performance monitoring system was designed and implemented to collect statistics on an LDM7 data-distribution network, and offer LDM7 administrators and users a Graphical User Interface (GUI) for visualization. This section describes the monitoring-system software architecture (in Section III-A), and the performance metrics collected (in Section III-B).

A. Architecture

An *LDM7-rtstats utility* is executed at each LDM7 server to collect metrics and push the data in a *LDM7-rtstats feed* (an LDM term used for file-streams) to a centralized *LDM7 performance monitor*. The LDM7 performance monitor runs a downstream LDM7 server to receive these feeds, parses the feeds using an rtstats-decoder, stores information into an *LDM7-rtstats database*, and offers visualization services for user querying through a *dashboard*.

In the LDM7-rtstats feed, an upstream LDM7 server sends the rate of the multipoint path for each feed, while downstream LDM7 servers send performance metrics computed from their LDM7 log files. The log files include per-product information such as arrival time, latency incurred in delivering the product (servers run NTP to synchronize clocks), size, mode of delivery: FMTP-only or with LDM6-backstop, number of FMTP block retransmissions, and feedtype.

We define three file types as follows: (i) *backstop-needed files* are data products that required LDM6-backstop retransmissions; (ii) *Multicast-itself-sufficient* files are data products that were successfully received by FMTP without any FMTP block retransmissions, i.e., these files were received fully in the multicast phase; and (iii) *FMTP-retx-needed* files are data products that were received by FMTP but required one or more FMTP block retransmissions. We use the additional term *FMTP-received files* to include both FMTP-retx-needed and Multicast-itself-sufficient files.

Fig. 2 shows the components of the LDM7 performance monitor: (i) Downstream LDM server, (ii) pqact utility, (iii) LDM7-rtstats-decoder, (iv) LDM7-rtstats database, and (v) dashboard. The Downstream LDM server can be an LDM6 or LDM7 server since in our current deployment, it is the only receiver of the LDM7-rtstats feed. A utility called `pqact`, which allows for an administrator to set the name of an executable program to handle all products of a feed, is used to invoke our LDM7-rtstats-decoder program when LDM7-rtstats feed products are received. The per-τ interval metrics received in the feed are saved in the LDM7-rtstats database. This database uses MongoDB, a NoSQL database. The LDM7 dashboard uses the model-view-controller architecture, in which the model interacts with the LDM7-rtstats database, the controller receives user input and passes it to the model, and the view creates the representation of the requested data for the user.

B. Metrics

Metric: Throughput: Per-file throughput is defined as file size divided by file latency. For average throughput, we compute the harmonic-mean (which is more appropriate than arithmetic mean when averaging rates [10]) of the per-file throughput of all FMTP-received, all multicast-itself-sufficient, and all FMTP-rext-needed files in time interval $(t-\tau, t)$ as follows:

$$\mathbb{T}_t^{fmtp} = \frac{\sum_{i \in N'_t} S_i}{\sum_{i \in N'_t} L_i} = \frac{\mathbb{S}'_t}{\mathbb{L}'_t} \tag{1}$$

$$\mathbb{T}_t^{mc} = \frac{\sum_{i \in N''_t} S_i}{\sum_{i \in N''_t} L_i} = \frac{\mathbb{S}''_t}{\mathbb{L}''_t} \qquad \mathbb{T}_t^{retx} = \frac{\mathbb{S}'_t - \mathbb{S}''_t}{\mathbb{L}' - \mathbb{L}''_t} \tag{2}$$

where S_i is file size of file i, L_i is file latency, N'_t is the number of all FMTP-received files, N''_t is the number of all multicast-itself-sufficient files, $\mathbb{S}'_t$ and $\mathbb{S}''_t$ are the cumulative size of all FMTP-received files and all multicast-itself-sufficient files, respectively, and $\mathbb{L}'_t$ and $\mathbb{L}''_t$ are the corresponding cumulative latencies in time interval $(t-\tau, t)$. The LDM7-rtstats feed products contain cumulative file sizes and cumulative latencies that were computed on a per-minute basis, i.e, τ was set to 1 minute. Per-hour (or longer-duration) throughput values can be computed from the per-minute cumulative sizes and cumulative latencies.

Metric: FMTP File Delivery Ratio (FFDR): These metrics characterize the success of file delivery via FMTP. We define

successful delivery of file i by FMTP if all blocks of file i were received via multicast alone or via multicast with one or more FMTP block retransmissions (the FMTP sender resends blocks for only those requests that are received before the expiry of a retransmission timer). However, with the LDM6-backstop mechanism, LDM7 ensures successful delivery of all files to all receivers as long as receivers request files within the specified duration for which files are served by the LDM6-backstop mechanism (typically 1 hour).

FFDR captures the extent to which FMTP was successful in delivering files without the LDM6-backstop mechanism. There are two measures for FFDR, which are file-count-based ($\mathbb{F}_t^{count}$) and sized-based ($\mathbb{F}_t^{size}$):

$$\mathbb{F}_t^{count} = \frac{N_t'}{N_t} * 100\% \qquad \mathbb{F}_t^{size} = \frac{\mathbb{S}_t'}{\sum_{i \in N_t} S_i} * 100\% \quad (3)$$

where N_t is the number of all LDM7-received files (backstop-needed files plus FMTP-received files) in the interval $(t-\tau, t)$. As with throughput, per-minute numerators and denominator values are sent to allow the rtstats-decoder of the performance monitor to compute FFDR over longer time durations.

Metric: Multicast Packet Loss Rate (MPLR): Packet loss rate is measured as a percentage of packets lost with respect to packets sent. Thus, we define MPLR ($\mathbb{L}_t^{mc}$) to quantify the multicast packet loss by measuring the requested FMTP block retransmissions as follow:

$$\mathbb{L}_t^{mc} = \frac{B_t * 1448}{\mathbb{S}_t'} * 100\% \quad (4)$$

where B_t is the number of FMTP block retransmissions in the interval $(t-\tau, t)$, 1448 bytes is the size of the FMTP-packet payload (FTMP, UDP, CTCP and IP headers are 12, 8, 20, and 20 bytes, respectively; to avoid a multicast block from requiring two CTCP segments in case of retransmissions, the FMTP-UDP multicast blocks carry 1448 bytes of payload). As with throughput, per-min numerators and denominator values are sent to allow the rtstats-decoder of the performance monitor to compute MPLR over longer time durations.

IV. Trial Deployment Over Multi-Domain SDN

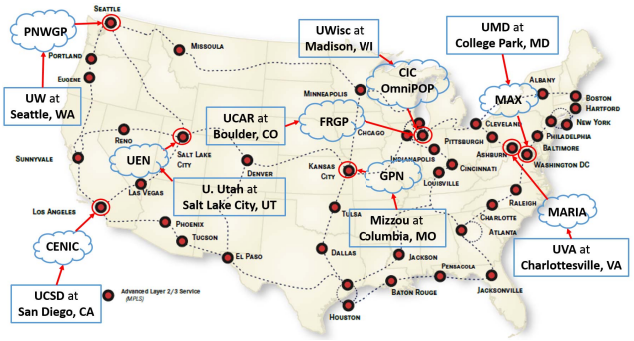


Fig. 3: Trial deployment of LDM7 across Internet2

Our trial deployment tests LDM7 in a WAN multi-domain setting to distribute the real-time meteorology data collected by the Unidata IDD project. While Internet2 offers Advanced Layer-2 Services (AL2S) with dynamic path provisioning along with L3 IP-routed service, the regional RENs only offer static VLAN service and L3 IP-routed service. The LDM7 application requires dynamic provisioning because receivers add and delete subscriptions to different feeds, and each feed should have its own multipoint VLAN for bandwidth and receiver processing efficiency. Therefore, our trial deployment used the dynamic provisioning capability of Internet2's AL2S in conjunction with statically provisioned VLAN segments across university campus networks and their regional RENs.

As shown in Fig. 3, we deployed LDM7 at eight university campuses. Most of the regional RENs are connected to the closest Internet2 PoP, e.g., the Virginia (UVA) regional REN, MARIA, is connected to the Internet2 switch at Ashburn, VA. However, one exception is that the Colorado based FRGP regional REN, which serves the University Cooperation for Atmospheric Research (UCAR), is connected via a 100 GE link to the Starlight Internet2 PoP in Chicago, for legacy reasons. This makes the Univeristy of Wisconsin (UWisc) LDM7 server the closest in terms of Round-Trip Time (RTT) to UCAR LDM7 server.

On each campus, the deployment effort consisted of: (i) deploying a server, (ii) installing and configuring LDM7 on the server, and (iii) provisioning VLANs (static path segments) across campus networks and regional RENs (edge networks in DRFSM service architecture). The last task required a campus network administrator and a corresponding regional REN operator to first agree on a common set of VLAN IDentifiers (IDs) and then provision these VLANs on every switch on the path from the campus LDM7 server all the way through the edge network to the Internet2 switch/port to which the regional REN is connected.

Combining static and dynamically provisioned VLANs: Fig. 4 shows an installation of DRFSM architecture on our LDM7 trial deployment, with a sender at UCAR, and receivers at the University of Virginia (UVA) and University of Maryland (UMD). MAX is the regional REN provider for UMD. VLAN segments are manually provisioned from the LDM7 servers in the three campuses to their corresponding Internet2 router/switch ports, e.g., VLAN III from UVA. This

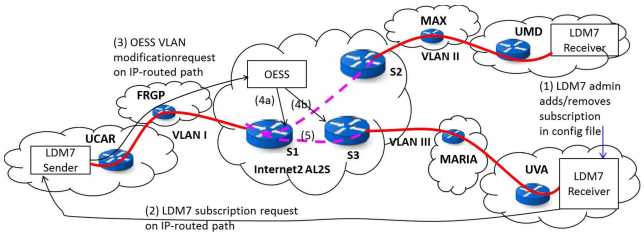


Fig. 4: An installation of DRFSM architecture on the LDM7 trial deployment; black arrows: control-plane messages; red lines: provisioned VLAN segments; magenta dashed lines: dynamic MPLS paths

step required our research team to communicate with network administrators at the various campuses, and in turn these administrators needed to communicate with their regional REN administrators to agree on a common set of VLAN IDentifiers (IDs) and get these VLANs provisioned across the campus networks and the regional RENs. We also asked each of our campus collaborators to authorize our Internet2 AL2S OESS working groups to make requests for connections to their VLANs on the Internet2 switch port connected to their regional-RENs.

Fig. 4 also illustrates how Internet2's dynamic L2 service capability is leveraged for our LDM7 trial deployment. Assume that VLAN I had been previously connected to VLAN II via an Internet2 AL2S MPLS path represented by the magenta dashed line between switches S1 and S2 to receive feed F1. This configuration would have occurred when the LDM7 receiver at MAX requested a new feed from the UCAR LDM7 server, at which point the latter would have signaled the Internet2 AL2S OESS server requesting the connection of VLAN I from FRGP's Internet2 AL2S switch port to VLAN II on MAX's Internet2 AL2S switch port.

Now consider the steps required when the LDM7 administrator at UVA adds a subscription to feed F1 in the LDM configuration file, step (1) in Fig. 4. Step (2) shows the LDM7 subscription request being sent on the IP-routed path. The LDM7 sender checks if the UVA receiver is authorized to receive feed F1, and if it is authorized, the Upstream LDM7 process instructs the OESS client at the sender to send a VLAN-modification request to the Internet2 AL2S OESS server, which is executed in (3). In (4a) and (4b), the OESS server performs book-keeping operations and then if the modification request can be accommodated, the OESS servers sends commands to the MPLS switches to reconfigure their forwarding tables. In the example shown in Fig. 4, switch S1 will add an entry to forward packets received with a particular MPLS label from VLAN I from its FRPG port to its link to S3, in addition to its entry for forwarding the same packets to its link to switch S2. Similarly, switch S3 will add an entry to forward packets received on its link from S1, with a specified MPLS label, to its port to MARIA on VLAN III. Step (5) illustrates that data starts flowing on the newly established MPLS-VLAN segment from switch S1 to the UVA LDM7 server.

V. Evaluation

Section V-A describes the experiments executed on the trial deployment. Section V-B presents our results.

A. Execution

Each of the servers deployed at the eight university campuses has at least 64 GiB RAM, 500 GB disk space and two ordinary network interfaces. A 1Gbps Ethernet (GbE) NIC connects to the general-purpose campus network for L3 IP-routed services, and a 10GbE NIC connects to a switch through which VLANs are provisioned to the nearest Internet2 switch port via the campus network and regional REN.

Input parameters that influence output metrics include feedtypes, VLAN/sending rate, and publisher/subscriber buffer sizes. We chose the NGRID feed (numerical model output from NOAA), and a specific publisher and set of subscribers, but varied the VLAN/sending rate and the set of receivers. The sender buffer size was set to a large enough value (600 MB) to prevent packet drops at the sender, and `tc` statistics confirmed that no packets were dropped due to sender-buffer overflows.

We ran three sets of experiments. The goal of the first set of experiments was to evaluate the newly modified LDM7 and performance monitoring system. Then a further investigation was executed with a second set of experiments to analyze the performance of LDM7 on the trial deployment. Last, we ran the third set of experiments to compare the performance of LDM7 with LDM6 on the trial deployment.

Experiment Set 1: The NGRID feed distribution was started on the UCAR LDM7 publisher. The UWisc, UVA, UMD, University of Utah (Utah), University of Washington (UWash), and University of California at San Diego (UCSD) LDM7 subscribers sent NGRID subscription requests to UCAR to join the multicast group. We limited the VLAN/sending rate at 40 Mbps in this set of experiments. The reason for rate-limiting at the sender is explained in Section II.

Experiment Set 2: We collected one-hour NGRID products, 03:00 to 04:00 UTC on Jul. 12, 2020. Then we created one-hour file-streams that follows the traffic pattern of the collected NGRID products (using the utility `pqinsert`) and replayed the data repeatedly in multiple experiments to avoid differences in incoming file-stream product sizes or inter-arrival times from influencing results.

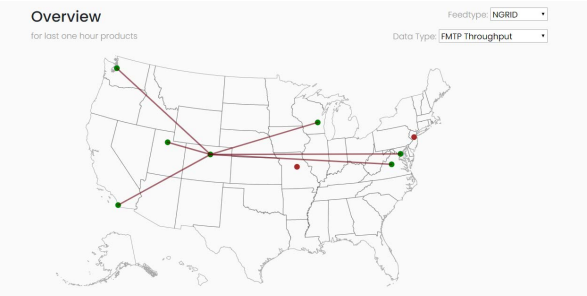
Experiment Set 3: LDM7 with LDM6 were compared when they had same configuration of VLAN/sending rates and sets of subscribers using the created one-hour file-stream.

B. Evaluation

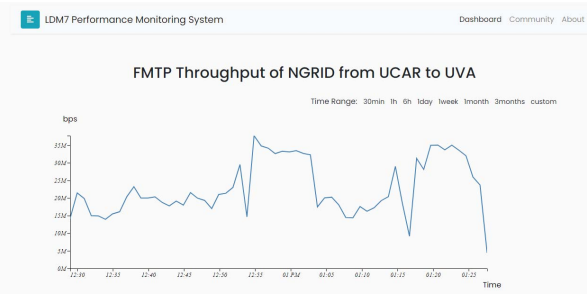
1) LDM7 performance monitoring system: Fig. 5a shows the LDM7 performance monitoring system dashboard for Experiment Set 1. The dashboard offers users a GUI to specify parameters such as the time range, feedtype, and metric of interest. The dashboard implementation is based on D3.js. Every publisher/subscriber that joins the meteorological data distribution is geo-located on a U.S. map. Publisher/subscribers are color coded as follows:

- red: Subscriber is unavailable, i.e., a product was last received on the ldm7-rtstats feed more than 1 hour ago;
- yellow: Less active, i.e., a product was last received on the ldm7-rtstats feed less than 1 hour but more than 10 minutes ago;
- green: Active, i.e., a product was last received on the ldm7-rtstats feed less than 10 minutes ago.

In Fig. 5a, there are six subscribers receiving real-time NGRID feed from the publisher, UCAR. The link between two servers, such as the publisher (UCAR) and the UVA LDM7 subscriber indicates the logical connection between them. A



(a) LDM7 performance monitoring system dashboard



(b) 1-hour throughput of NGRID from UCAR to UVA

Fig. 5: LDM7 Performance Monitoring Dashboard

time-series plot with per-minute information for the link is shown in Fig. 5b. It shows the throughput of NGRID feed from UCAR to UVA in one hour.

2) LDM7 Metrics: Table II shows the metrics, defined in Section III-B, when the multicast group contains one publisher, UCAR, and six subscribers, UMD, UWisc, UWash, UCSD, UVA, and Utah.

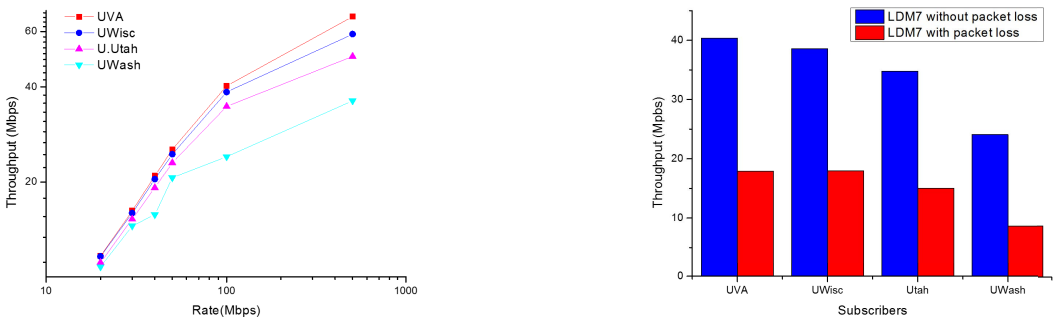
First, we observe that the overall DRFSM solution worked well on the trial deployment. We observe from the Average throughput rows that different subscribers achieved different values, due to different propagation delays. Since This is due to differences in product-latency, whose components are: (i) processing delays, (ii) sender-buffering delays incurred because of the sender `tc` rate limiting used in LDM7, (iii) emission (transmission) delays, (iv) propagation delays (roughly half of RTT), (v) switch/router packet queueing delays, and (vi) retransmission delays. Processing delays are typically negligible when compared to RTT in the WAN setting. All subscribers have the same, small sender-buffering delays because' the six subscribers request the NGRID feed from the same publisher. One-way propagation delays are high in this WAN setting (e.g., 20 ms between UCAR and UVA). Given the high link capacities (10 Gbps or higher) in the production RENs on which our trial was deployed, switch/router packet queueing delays should be small. Our observation in Fig. 6a confirms this conclusion.

Fig. 6a shows the average throughput for Experiment set 2. We replayed NGRID feed from UCAR to the four subscribers, UWisc, UVA, Utah, and UWash with varying VLAN/sending rates among 20, 30, 40, 50, 100, 500 Mbps. Generally, as the VLAN/sending rate increases, the throughput of each subscriber increases. But different subscribers achieved different throughput at the same VLAN/sending rate. UVA has the highest average throughput, while UWash has the lowest. The most reasonable explanation is their different RTTs (along the path from the publisher to each subscriber).

Second, regarding reliability, we used two FFDRs and MPLR defined in Section III-B to evaluate the reliable capability of LDM7. Such as 100% of file-count-based FFDR and 100% of size-based FFDR indicate that reliable mechanism worked well and all files are delivered by FMTP (data-plane solution) without requiring the LDM6-backstop mechanism. FMTP uses a TCP retransmission mechanism to deliver the blocks that are not able to transmit over multicast UDP. Multicast Packet Loss Rate (MPLR) row in Table II shows the ratio of lost multicast packets, which also presents the ratio of block retransmissions. This retransmission would incur significant delays in the production WAN setting. Fig. 6b shows the impact of packet loss on LDM7. With artificially injected packet loss. All 45702 products were delivered successfully to the four subscribers. One interesting finding in Fig. 6b is that, although all subscribers suffered throughput degradation, different subscribers performed differently. For exmaple, the throughput of UVA LDM7 decreases from 40.37 Mbps to 17.90 Mbps (55.66%), while the of UWash LDM7 decreases from 24.07 Mbps to 8.67 Mbps (63.98%).

TABLE II: Experiment Set 1: Statistics for files received by the UVA, UWash, UMD, UWisc and UCSD LDM7 receivers; t is the start time of the experiment and τ is the whole duration

Subscribers	UVA	UMD	UWisc	UWash	UCSD	Utah
Number of FMTP-received files	45642	45642	45642	45642	45642	45642
File-count-based FFDR $\mathbb{F}_t^{count}$	100%	100%	100%	100%	100%	100%
Size-based FFDR $\mathbb{F}_t^{size}$	100%	100%	100%	100%	100%	100%
Number of files that needed FMTP retransmissions	3	3	3	6	4	64
Number of FMTP block retransmissions	21	21	21	34	24	529
Multicast Packet Loss Rate (MPLR) $\mathbb{L}_t^{mc}$	3.5e-4%	3.5e-4%	3.5e-4%	5.6e-4%	4.0e-4%	8.8e-3%
Average throughput of FMTP-received files (Mbps) $\mathbb{T}_t^{fmtp}$	20.92	21.08	20.43	13.81	18.03	19.17
Average throughput of multicast-itself-sufficient files (Mbps) $\mathbb{T}_t^{mc}$	20.93	21.08	20.44	13.83	18.04	19.31
Average throughput of FMTP-retx-needed files (Mbps) $\mathbb{T}_t^{retx}$	0.36	0.35	0.33	0.11	0.24	0.23



(a) LDM7 Throughput varying VLAN/sending rate

(b) LDM7 Throughput w/o packet loss at 100 Mbps

Fig. 6: Experiment Set 2 Results

3) Performance of LDM7 and LDM6 over trial deployment: The goal of this subsection were two-fold: (i) compare the average throughput of LDM6 and LDM7 under same VLAN/sending rates, and (ii) compare the VLAN/sending rates for LDM6 and LDM7 to achieve the same throughput. The main parameters varied in this experiment (Experimental Set 3) were the VLAN/sending rate and set of subscribers.

First, we replayed NGRID feed from UCAR to four subcribers, UVA, UWisc, Utah, and UWash with varying VLAN/sending rate among 50, 100, 200, 300, 400, 500 Mbps via LDM6 and LDM7. The Fig. 7 presents the results we collected from the trial deployment. Generally, the throughput of LDM6 is less than the throughput of LDM7. And there is a clear gap between the average throughput of LDM7 across the four subscribers and the average throughput of LDM6. As the VLAN/sending rate increases, the differences between LDM7 throughput and LDM6 throughput increases. For example, at the VLAN/sending rate of 50 Mbps, the different value is 21.88 Mbps, while at 100 Mbps, the value increases to 31.62 Mbps. But, it stabilizes around 31 Mbps when the VLAN/sending rate reaches 100 Mbps. Furthermore, Fig. 7

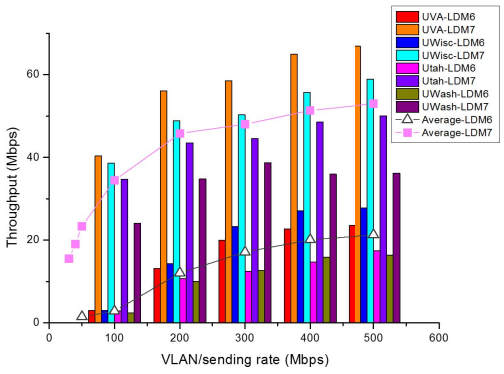


Fig. 7: Average throughput of LDM6 and LDM7 with four subscribers varying VLAN/sending rate at UVA

shows that there is an around 90% bandwidth saving for LDM7 (sending at 40 Mbps), compared to LDM6 (sending at 400 Mbps), to achieve a 20 Mbps average throughput with four subscribers. This verifies that LDM7 significantly reduces the need for bandwidth, and it even be better when the number of subscribers is larger.

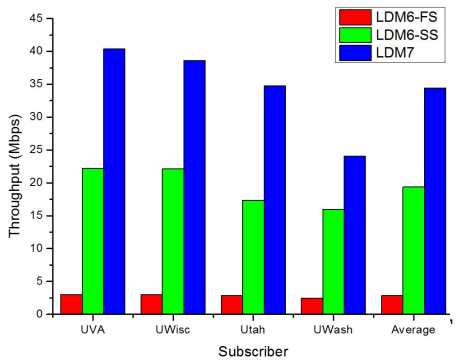
Second, we take a close look at the VLAN/sending rate of 100 Mbps. Fig. 8a shows the throughput comparison of LDM6 and LDM7 at 100 Mbps. LDM6-FS (four subscribers) shows throughput of LDM6 with four subscribers, while LDM6-SS (single subscriber) means the throughput of LDM6 with single subscriber at 100 Mbps. The first finding in Fig. 8a is LDM7 achieved a 77.6% higher average throughput than LDM6, even there is only one subscriber (LDM6-SS). This improvement may come from the difference between TCP and UDP. Another observation is that the throughput of LDM6-FS has an almost seven-fold improvement to the throughput of LDM6-SS. One possible explanation is a larger sender-buffer queuing delay with multiple copies in LDM6-FS. Moreover, Fig. 8b presents the impact of number of subscribers on the performance of LDM7 and LDM6 at 100 Mbps. With the number of subscriber increases, the throughput of LDM7 varies in a small ranges, while the throughput of LDM6 decreases a lot. Especially, at UVA, with six subscribers, the throughput of LDM7 (40.12 Mbps) has an almost 22-fold improvement to the throughput of LDM6 (1.79 Mbps).

VI. Related Work

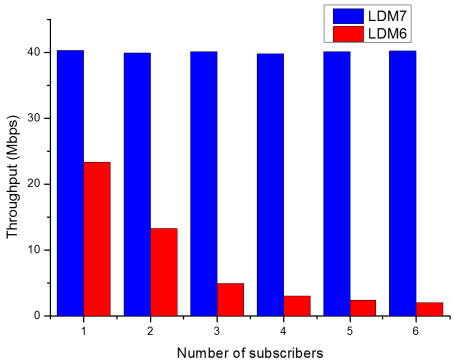
Section VI-A offers the reader background on dynamic Layer-2 (L2) path service. Section VI-B reviews recent related work.

A. Dynamic path-based service

To support dynamic L2-path service, first, control-plane protocols have been specified and standardized. These include Inter-Domain Controller Protocol (IDCP) [11] and the Open Grid Forum Network Service Interface Connection Services (NSI CS) version 2.0 [12]. Both protocols support inter-domain signaling for advance reservation and provisioning of



(a) Throughput at various subscribers



(b) Throughput varying number of subscribers observed at UVA

Fig. 8: Experiment Set 3 Results: Throughput comparison of LDM6 and LDM7 at the VLAN/sending rate of 100 Mbps

rate-guaranteed dynamic L2 paths. Second, there are many SDN controllers [13], of which Internet2 deployed an Open Exchange Software Suite (OESS) server as the controller for its Advanced L2 Service (AL2S) offering. The OESS server has both a GUI and programmatic interface for users to request paths between Internet2 router/switch ports. The technology used for L2 paths in the current deployment is MultiProtocol Label Switching (MPLS). User working group identifiers are used for access control allowing a regional REN to authorize a remote user group to connect to particular Virtual LANs (VLAN) on its Internet2 switch port. Rate and duration can be specified in the request for an L2 path, and multipoint VLAN/MPLS virtual topologies are supported.

B. Recent related work

Solutions have been proposed to leverage SDN techniques to provide efficient and well-managed network-multicast services. Many of these solutions [14]–[16] aim to find optimal trees. These SDN-based multicast advances focused on the control-plane problem of finding the best multicast topologies but not on the data-plane aspects.

Failures can affect the quality of real-time multicasting services. Some solutions have investigated methods to make multicasting reliable, such as Multicast TCP (MCTCP) [17], and ECast [18]. MCTCP is designed for small-multicast groups, and eCast requires a sub-tree of the multicast tree to be established to multicast retransmissions. Neither of these solutions work in a WAN context.

More recent papers include the following. Desmouceaux et al. [19] proposed a solution that requires all routers to implement a Bit-Indexed Explicit Replication (B.I.E.R) shim layer, which is an expensive modification for WAN deployment. Multicasting solutions for SDN based data center networks include Multicast Routing for Data Centers (MCDC) [20], ATHENA [21], and Datacast [22], but these are not readily extendible for WAN deployment. The DCCast solution [23] is proposed for inter-data-center multicasts, and is only evaluated with synthetic traffic through simulations, i.e., practical deployment considerations of addressing, routing and transport-layer protocols are not considered.

VII. Conclusions

This work demonstrated a trial deployment of L2 path-based network multicast solution and IP-routed service. First, our experiments showed that file-delivery ratios and throughput metrics achieved in this DRFSM implementation met LDM7 application requirements. Second, the LDM7 performance dashboard with key LDM7 performance metrics worked well. Third, we compared LDM7 with LDM6 on the trial deployment, which showed an almost 22-fold throughput improvement at the VLAN/sending rate of 100 Mbps with six subscribers and 90% bandwidth savings from path-based network multicast solution to achieve a 20 Mbps average throughput across four subscribers. While our current design handled these gaps in multipoint path availability well, which is within requirements for the LDM/IDD application that we tested, a new design is needed to alleviate the impact of DRFSM control-plane overheads.

Acknowledgment

We thank NSF for grant 1659174 and our campus partners for supporting the trial: University of Wisconsin, University of Maryland, University of California San Diego, University of Washington, University of Utah, University of Missouri, and Rutgers University, and their regional RENs, Internet2 and GRNOC.

References

[1] S. Ratnasamy, A. Ermolinskiy, and S. Shenker, "Revisiting IP Multicast," in *Proceedings of the 2006 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, ser. SIGCOMM '06. New York, NY, USA: ACM, 2006, pp. 15–26. [Online]. Available: http://doi.acm.org/10.1145/1159913.1159917

[2] S. Chen, X. Ji, M. Veeraraghavan, S. Emmerson, J. Slezak, and S. G. Decker, "A cross-layer Multicast-Push Unicast-Pull (MPUP) architecture for reliable file-stream distribution," in *2016 IEEE 40th COMPSAC*, vol. 1, June 2016, pp. 535–544.

[3] Y. Tan, S. Chen, S. Emmerson, Y. Zhang, and M. Veeraraghavan, "Advances in Reliable File-Stream Multicasting over Multi-Domain Software Defined Networks (SDN)," in *2019 28th International Conference on Computer Communication and Networks (ICCCN)*, July 2019, pp. 1–11.

[4] M. Berman, J. S. Chase, L. Landweber, A. Nakao, M. Ott, D. Raychaudhuri, R. Ricci, and I. Seskar, "GENI: A federated testbed for innovative network experiments," *Computer Networks*, vol. 61, pp. 5–23, 2014.

[5] K. Keahey, P. Riteau, D. Stanzione, T. Cockerill, J. Mambretti, P. Rad, and P. Ruth, "Chameleon: a scalable production testbed for computer science research," in *Contemporary High Performance Computing: From Petascale toward Exascale*, 1st ed., ser. Chapman & Hall/CRC Computational Science, J. Vetter, Ed. Boca Raton, FL: CRC Press, 2018, vol. 3, ch. 5.

[6] J. Moy, "OSPF Version 2," IETF, RFC 2328, Apr. 1998. [Online]. Available: http://tools.ietf.org/rfc/rfc2328.txt

[7] Y. Rekhter, T. Li, and S. Hares, "A Border Gateway Protocol 4 (BGP-4)," IETF, RFC 4271, Jan. 2006. [Online]. Available: http://tools.ietf.org/rfc/rfc4271.txt

[8] J. Li, M. Veeraraghavan, S. Emmerson, and R. Russell, "File multicast transport protocol (FMTP)," in *IEEE CCGrid, SCRAMBL workshop*, May 2015, pp. 1037–1046.

[9] A. Mudambi, X. Zheng, and M. Veeraraghavan, "A transport protocol for dedicated end-to-end circuits," in *IEEE ICC*, vol. 1, June 2006, pp. 18–23.

[10] B. Jacob and T. Mudge, "Notes on calculating computer performance," *University of Michigan, Tech. Rep. CSE-TR-231-95*, March 1995.

[11] A. Lake, J. Vollbrecht, A. Brown *et al.*, "Inter-domain Controller (IDC) Protocol Specification," 2008.

[12] G. Roberts, T. Kudoh, I. Monga, J. Sobieski, J. MacAuley, and C. Guok, "NSI Connection Service v2.0," *GFD. 212*, pp. 1–119, 2014.

[13] D. Kreutz, F. M. V. Ramos, P. E. Veríssimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, Jan 2015.

[14] J.-P. Sheu, C.-W. Chang, and Y.-C. Chang, "Efficient multicast algorithms for scalable video coding in software-defined networking," in *2015 IEEE 26th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*. IEEE, 2015, pp. 2089–2093.

[15] Y.-D. Lin, Y.-C. Lai, H.-Y. Teng, C.-C. Liao, and Y.-C. Kao, "Scalable multicasting with multiple shared trees in software defined networking," *Journal of Network and Computer Applications*, vol. 78, pp. 125–133, 2017.

[16] P. Yu, R. Wu, H. Zhou, H. Yu, Y. Chen, and H. Zhong, "Multicast routing tree for sequenced packet transmission in software-defined networks," in *Proceedings of the 8th Asia-Pacific Symposium on Internetware*. ACM, 2016, pp. 27–35.

[17] T. Zhu, F. Wang, Y. Hua, D. Feng, Y. Wan, Q. Shi, and Y. Xie, "MCTCP: Congestion-aware and robust multicast TCP in software-defined networks," in *2016 IEEE/ACM 24th International Symposium on Quality of Service (IWQoS)*. IEEE, 2016, pp. 1–10.

[18] X. Zhang, M. Yang, L. Wang, and M. Sun, "An OpenFlow-enabled elastic loss recovery solution for reliable multicast," *IEEE Systems Journal*, vol. 12, no. 2, pp. 1945–1956, 2018.

[19] Y. Desmouceaux, T. H. Clausen, J. A. C. Fuertes, and W. M. Townsley, "Reliable multicast with B.I.E.R." *Journal of Communications and Networks*, vol. 20, no. 2, pp. 182–197, April 2018.

[20] S. Shukla, P. Ranjan, and K. Singh, "MCDC: Multicast routing leveraging SDN for Data Center networks," in *2016 6th International Conference - Cloud System and Big Data Engineering (Confluence)*, Jan 2016, pp. 585–590.

[21] K. Mahajan, D. Sharma, and V. Mann, "Athena: Reliable multicast for group communication in SDN-based data centers," in *2017 9th International Conference on Communication Systems and Networks (COMSNETS)*, Jan 2017, pp. 174–181.

[22] J. Cao, C. Guo, G. Lu, Y. Xiong, Y. Zheng, Y. Zhang, Y. Zhu, C. Chen, and Y. Tian, "Datacast: A scalable and efficient reliable group data delivery service for data centers," *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 12, pp. 2632–2645, December 2013.

[23] M. Noormohammadpour, C. S. Raghavendra, S. Rao, and S. Kandula, "DCCast: Efficient Point to Multipoint Transfers Across Datacenters," in *9th USENIX Workshop on Hot Topics in Cloud Computing (HotCloud 17)*. Santa Clara, CA: USENIX Association, Jul. 2017. [Online]. Available: https://www.usenix.org/conference/hotcloud17/program/presentation/noormohammadpour