Cluster Computing (2022) 25:2931–2952
https://doi.org/10.1007/s10586-021-03519-6

High-performance reliable network-multicast over a trial deployment

Yuanlong Tan[1] · Malathi Veeraraghavan[1] · Hwajung Lee[2] · Steven Emmerson[3] · Jack W. Davidson[1]

Received: 13 May 2021 / Revised: 9 October 2021 / Accepted: 9 December 2021 / Published online: 4 February 2022

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. As both the data volume and number of subscribers grows, a reliable network multicast is a promising approach to alleviate the demand for the bandwidth needed to support efficient data distribution to multiple, geographically-distributed, research communities. In prior work, we identified the need for a reliable network multicast: scientists engaged in atmospheric research subscribing to meteorological file-streams. An application called Local Data Manager (LDM) is used to disseminate meteorological data to hundreds of subscribers. This paper presents a high-performance, reliable network multicast solution, Dynamic Reliable File-Stream Multicast Service (DRFSM), and describes a trial deployment comprising eight university campuses connected via Research-and-Education Networks (RENs) and Internet2 and a DRFSM-enabled LDM (LDM7). Using this deployment, we evaluated the DRFSM architecture, which uses network multicast with a reliable transport protocol, and leverages Layer-2 (L2) multipoint Virtual LAN (VLAN/MPLS). A performance monitoring system was developed to collect the real-time performance of LDM7. The measurements showed that our proof-of-concept prototype worked significantly better than the current production LDM (LDM6) in two ways. First, LDM7 distributes data faster than LDM6. With six subscribers and a 100 Mbps bandwidth limit setting, an almost 22-fold improvement in delivery time was observed with LDM7. Second, LDM7 significantly reduces the bandwidth requirement needed to deliver data to subscribers. LDM7 needed 90% less bandwidth than LDM6 to achieve a 20 Mbps average throughput across four subscribers.

Keywords File-stream distribution · Software-defined network · Multicast · Control-plane protocol

1 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 resources. For example, in Unidata's Internet Data Distribution (IDD) system [1], the University Corporation for Atmospheric Research (UCAR) uses an application, called Local Data Manager (LDM) [2], to distribute 30 different types [3] of meteorological data (e.g., surface observations, radar data, satellite imagery, wind profiler data, lightning data, and high-resolution computer-model output) to over 570 sites in 217 domains [4]. Approximately 420,000 data products[1] comprising 50 gigabytes (GB) are generated each hour. The volume of data and number of subscribers have both been increasing. For example, the weather satellites of the GOES-R series, such as, GOES-16 and GOES-17, which came online in recent years has 14 times higher

✉ Yuanlong Tan
alongertan@gmail.com

Hwajung Lee
hlee3@radford.edu

Steven Emmerson
emmerson@ucar.edu

Jack W. Davidson
jwd@virginia.edu

[1] University of Virginia, Charlottesville, VA, USA

[2] Radford University, Radford, VA, USA

[3] University Corporation for Atmospheric Research, Boulder, CO, USA

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

operational bandwidth than the previous-generation satellite [5].

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's computing power and network resources requirements have been growing rapidly in the IDD system. While UCAR receives 50 GB/h from various input sources, it transmits, on average, about 2.34 TB/h from its sending cluster. A network multicast solution would alleviate the demand of both computing power and network resources. 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. IP multicast has three components: (i) Class-D multicast IPv4 addresses and automated mapping to multicast MAC addresses, (ii) Internet Group Multicast Protocol (IGMP), and (iii) multicast routing protocols. Distributed routing solutions, with their associated protocols, were developed to support IP multicast. Examples include Distance Vector Multicast Routing Protocol (DVMRP) [6] and Protocol Independent Multicast (PIM) [7]. However, the complexity of these solutions is one of the reasons for questioning their feasibility [8]. The use of centralized controllers in SDN greatly simplifies the control-plane actions required to configure forwarding tables in switches for multicast flows. Inter-domain multicast trees can similarly be configured through coordinated operations in SDN controllers, one in each domain. Thus, SDN provided the *bottom-up technological-advance motivation* to revisit the use of network multicast.

In prior work, we described a network-multicast architecture to support reliable file-stream multicasting [9, 10]. It has the following features: (i) Layer-2 (L2) multipoint Virtual LAN (VLAN) service, and (ii) File Multicast Transport Protocol (FMTP) [11], a reliable transport-layer protocol for delivering file-streams, which uses both UDP and TCP over the L2 network service. The proposed architecture was evaluated on the NSF GENI [12] and Chameleon testbeds [13].

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 which are connected via corresponding regional Research-and-Education Networks (RENs) and Internet2, the US-wide REN [14]. To accommodate this multi-domain Wide-Area Network (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:

1. The design and implementation of a DRFSM service.
2. The design and implementation of a performance monitoring system to collect data for a rigorous evaluation.
3. A deployment involving eight university campuses of this modified LDM, called LDM7, to compare its performance to the current meteorological data distribution application, LDM6.

In sum, the LDM6 delivers file-streams using unicast TCP connections from the publisher to each subscriber, which is an Application Layer Multicast (ALM) solution. As an example, the CONDUIT feedtype, which consists of high-resolution model data files, shows that each file is sent out multiple times because of the use of unicast TCP connections. The LDM7 multicasts file-streams over UDP sockets to all subscribers. Network switches/routers perform multicast function instead of the sending clusters. Thus, each CONDUIT file will be sent out only once, which alleviate the demand for sender's access bandwidth.

This paper has the following organization: Sect. 2 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 3 describes the design and implementation of a new LDM7 performance monitor and defines metrics used in our evaluation. Our trial deployment in a production WAN setting is described in Sect. 4. Section 5 presents results of our experimental evaluation on this deployment. Section 6 provides additional background material and reviews recent related work. Section 7 concludes the paper.

2 Dynamic reliable file-stream multicast service

Section 2.1 illustrates and reviews the DRFSM architecture in a bottom-up manner. Section 2.2 describes our implementation of DRFSM control-plane and integration of these modules into LDM7.

2.1 Architectural description

Figure 1 illustrates the 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.

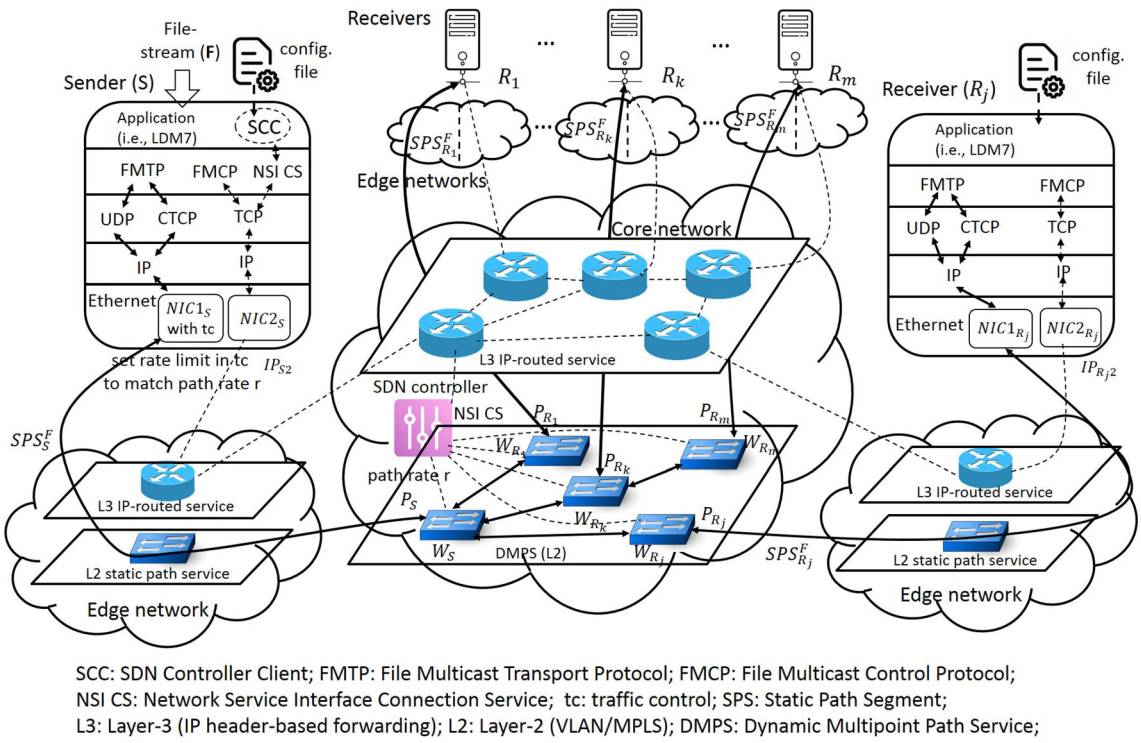


Fig. 1 Dynamic Reliable File-Stream Multicast (DRFSM) service architecture

The reason we require that the *multipoint network service be path-based* 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.

Consider the TCP solutions for unicast error control, flow control and congestion control. The TCP *error control* scheme of using positive acknowledgments (ACKs) from the receiver, in conjunction with retransmission timeouts at the sender, would suffer from the ACK-implosion problem when used with multiple receivers. Similarly, the window-based *flow control* scheme, in which the receiver notifies the sender of the available space in the TCP-receive buffer, is cumbersome to use in a multi-receiver setting. Finally, TCP's Slow Start and Congestion Avoidance (*congestion control*) schemes for handling congestion require the sender to estimate the state of the network path from the sender to the receiver and correspondingly adjust the size of the congestion window, which indirectly determines the sending rate. Using such schemes in a multi-receiver context is complex as the states of multiple network paths will need to be considered in determining the size of the sender congestion window/sending rate.

The use of SDN controllers to provision rate-guaranteed L2 multipoint virtual networks, such as VLANs, simplifies the three key transport-layer functions in a multi-receiver context. The path-based solutions support in-sequence delivery, which allows for the use of a Negative-acknowledgments (NACK) based error control mechanism. A receiver can assume that a packet was dropped when it sees an out-of-sequence block, and then send a NACK requesting block retransmission. The 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. It is the responsibility of each receiver to ensure that its network path from the sender has sufficient bandwidth and that its host has sufficient compute resources to keep up with packets arriving into its receive buffer.

If a receiver finds that it cannot handle the rate at which a file-stream of its interest is being offered by a sender, then it can use external means to communicate with the sender operator and request that the file-stream be served at a lower rate. The key point is that active sending rate adjustments in the data-plane, i.e., during file-stream transmission, is avoided so that one receiver with insufficient compute or network resources does not slow down data delivery to other receivers. In sum, choosing to design the DRFSM service on top of a network that offers a multipoint, path-based service simplifies the transport-layer functions needed to support reliable multicast.

The second feature required from the network service is *dynamic control*. In any data subscription service, subscribers[2] will need the ability to join/leave multicast groups. As subscribers add or drop from multicast groups, the network service should allow for the corresponding path segments to the subscribers to be added to or dropped from the multipoint paths. Hence dynamic control of the multipoint path service is required.

Having provided these high-level insights into our design choices for the DRFSM service, we now present details related to the networks and their configuration for this service.

2.1.1 Networks

Our architecture model allows file-stream publishers and subscribers 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 MultiProtocol Label Switching (MPLS) forwarding, referred to as L2, for path-based service.

2.1.2 Network interface cards (NICs)

Figure 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 publisher NIC should be limited to the rate of the multipoint path. The publisher 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.

2.1.3 Addressing, routing and configuration

To support L3 service, distributed intra- and inter-domain *routing protocols* are assumed to be deployed. For example, Open Shortest Path First (OSPF) [15] and Border Gateway Protocol (BGP) [16] 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 1 shows the parameters used by DRFSM to perform dynamic multipoint path control operations for one file-stream. Some of these parameters are per-host (publisher/subscriber), while others are per-file-stream. The first two rows of the publisher and subscriber sections are the per-host parameters. They 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. 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 1 are illustrated in Fig. 1.

SPSs are provisioned from the publisher to its DMPS core-network switch port for each file-stream that it multicasts. SPSs are provisioned from each subscriber to its DMPS core-network switch port for each of its subscribed file-streams. Examples of SPS are VLANs. For each file-stream, the publisher's configuration file has a rate-setting

[2] The terms "publisher" and "subscriber" are "sender" and "receiver" are used in data subscription services.

Table 1 Configuration file entries for file-stream **F**

Publisher S	
(W_S, P_S)	A core-network switch and port to which the publisher's SPSs are provisioned
$NIC1_S$	an identifier of the NIC1 at the publisher
$SPS_S^{\mathbf{F}}$	An identifier of the SPS provisioned between the publisher 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-receive IP address and UDP port number used for multicast of **F**
Subscriber $R_j, 1 \leq j \leq m$	
(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 subscriber 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 publisher, along with a TCP port number

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 publisher to all subscribers 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 publisher; 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 shown in Table 1, at the publisher, 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 publisher can assign IP addresses within the same subnet to the SPSs connected logical interfaces at all the subscribers on the multipoint path. For each file-stream, Subscribers require the IP address of the corresponding publisher's NIC2 public IP address and TCP-port number for sending a subscription request. Other IP addresses required at the subscriber are communicated via signaling from the publisher.

2.1.4 Transport protocols

A reliable multicast transport protocol, FMTP, is used in the DRFSM 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 publisher and all subscribers ($NIC1_S$ and $NIC1_{R_j}$). 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 sender sends a special FMTP packet, which contains the metadata of the file, named Beginning-of-Product (BOP) message via L2 multicast to all receivers. Next, the FMTP sender divides the file into blocks, and multicasts these as UDP datagrams. Finally, FMTP multicasts an End-of-Product (EOP) message to all receivers.

Each FMTP receiver checks the FMTP packet header and detects missing blocks. Because sequenced delivery is guaranteed on the L2 paths of the multipoint VLAN, missing blocks are detected from the block sequence number. Unicast TCP connections, sent over the L2 paths from the publisher to the subscriber, are used for retransmissions. A unicast private IP address is assigned to the VLAN interface at NIC1 on the publisher and at each subscriber. This private IP address is used for the TCP 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 FMTP 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 after the BOP is sent. When this timer expires, the FMTP sender stops serving all pending and new retransmission requests and sends back rejections (see Fig. 2

example). This timer is required to prevent slow receivers from reducing multicast throughput for all other receivers.

On the FMTP receiver side, a receiver timer is set for each file when its BOP is received. This timer is required to handle the loss of one or more blocks at the end of the file and loss of EOP. If the timer expires before the reception of the corresponding EOP, the FMTP receiver requests retransmissions for all missing blocks and the EOP for that file.

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.

2.2 Implementation

Section 2.2.1 provides an overview of the LDM7 processes and inter-process communications. Section 2.2.2 describes how LDM7 uses the services of FMTP, and Sect. 2.2.3 describes a utility used to configure VLAN IDs and IP addresses at the end hosts.

2.2.1 Overview of LDM7 processes

Figure 2 illustrates how LDM7 processes are created at the publisher and at two subscribers for two feeds with FMTP-required communications setup across the L2 network. The numbered arrows offer a chronological representation of inter-process communications. When the top-level `ldmd` process at Subscriber1 (R1) processes a RECEIVE entry in its configuration file (in which the LDM administrator specifies the desired feeds and corresponding parameters), this top-level process forks a downstream LDM7 process (1). The latter generates a subscription request (2) for the feed, Feed1 (F1) in the example, which is sent using IP-routed (Layer3) service to the sender. At the sender, the top-level `ldmd` process forks an Upstream LDM7 process (3), which in turn forks a multicast-LDM (mLDM) process for Feed1 (4) if necessary. The R1 Downstream LDM7 process initiates a TCP connection to a private IP address assigned to the Feed1 VLAN at the publisher (5) on its NIC2. This TCP connection uses the VLAN created in the L2 network.

This five-step process is shown for Feed2 (F2) from R1 (arrows 6–10). As step (8) shows, a separate upstream LDM7 process and a corresponding mLDM process are forked for the new Feed2 at the publisher. Arrows 11–15 show what happens when a second subscriber, Subscriber2 (R2), subscribes to an existing feed, i.e., Feed1. A new

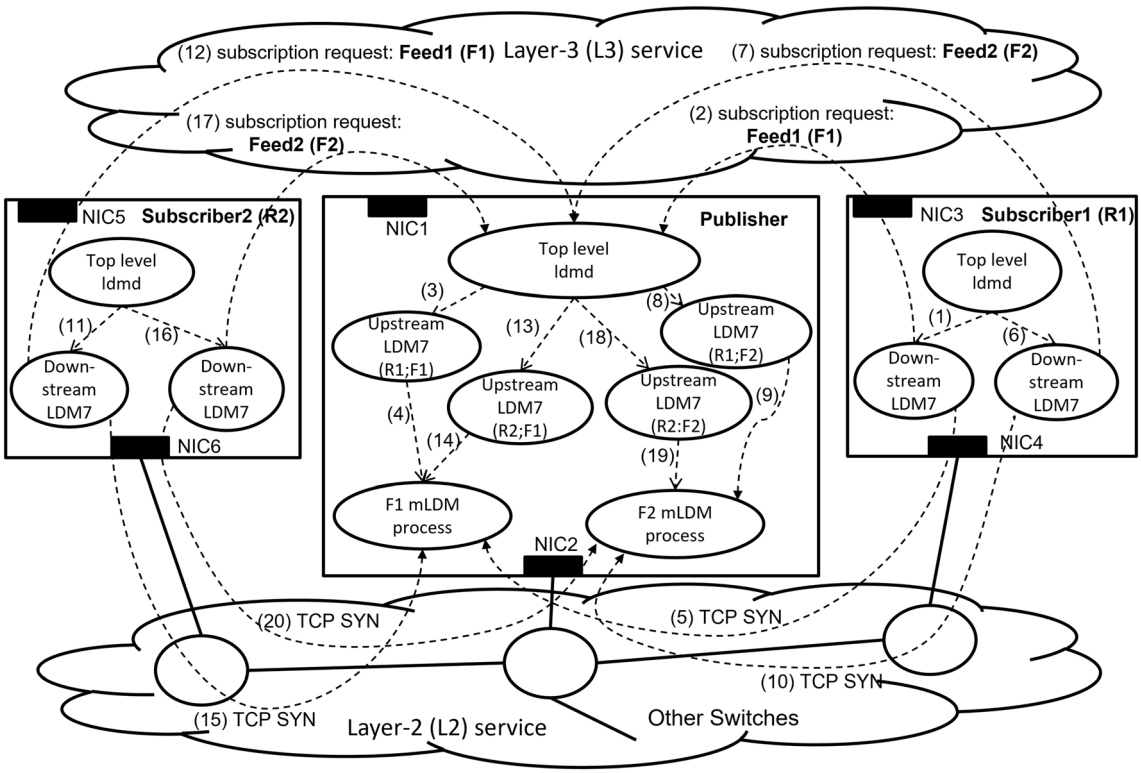


Fig. 2 LDM7 inter-process communications with one publisher and two subscribers R1 and R2 for feeds F1 and F2

Upstream LDM7 process is forked for R2 at the publisher for Feed1 (marked as R2; F1). But this new process connects to the same mLDM process for Feed1. Arrows 16–20 show the flow for R2 subscribing to Feed2.

2.2.2 LDM7 integration with FMTP

A modification to LDM to support LDM7 capabilities is the ability to exchange additional IP addresses and transport-layer ports to support the various paths/modes of communication. Administrators need to add information on VLAN IDs, the network-switch name and port to which the hosts have provisioned VLAN segments, and IP addresses to the LDM7 configuration files. As indicated in Table 1, administrators need to have entries of an identifier of the publisher's SPS ($SPS_S^{\mathbf{F}}$), core-network switch and port to which the publisher's SPS are provisioned ((W_S, P_S)), and two IP addresses (($IP_{mr}^{\mathbf{F}}, N_{mr}^{\mathbf{F}}$) and ($IP_{NIC1_S}^{\mathbf{F}}, M^{\mathbf{F}}$)) in the LDM7 configuration files. As only modifications relative to LDM6 are described here, a reader interested in the details of LDM6 is referred to the LDM software site [2].

Downstream LDM7 process This process sends a subscription request for a feed to the publisher top-level LDM server, which specifies: (i) desired feed, (ii) core-network switch and port to which the publisher's SPS are provisioned, and (iii) an identifier of the subscriber's SPS. The subscription reply specifies: (i) parameters required by the subscriber FMTP for the requested feed: the IP address and port number of the FMTP/UDP multicast group, the publisher's private IP address and port number for use by the subscriber to request the TCP connection for FMTP retransmissions; (ii) the private IP address (unique on that multipoint VLAN) and netmask that the downstream LDM should use to assign to its virtual interface.

When the downstream LDM7 process receives a positive subscription response (which indicates a successful subscription) from the upstream LDM process, it:

1. Forks a child process that uses the Linux `ip` utility to create a virtual interface with the VLAN ID specified in the RECEIVE entry, and assign the private IP address specified in the subscription reply.
2. Executes the FMTP Receiving thread to receive multicast data-packets over the VLAN as shown in Fig. 3, while passing this thread the IP address and port number of the FMTP/UDP multicast group.
3. Creates the FTMP Retransmission request thread, which sends requests for FMTP block retransmissions as shown in Fig. 3. The private IP address and port number for the TCP connection over Layer-2 service are provided to this thread to initiate TCP connection setup to the publisher.

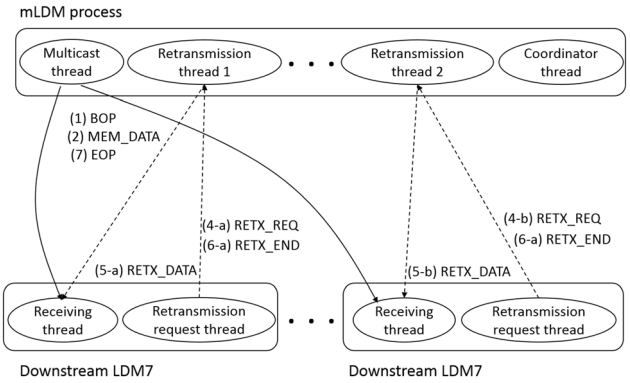


Fig. 3 FMTP threads and messaging

4. Executes two more FMTP threads: a retransmission handler thread, which is responsible for receiving FMTP blocks retransmitted on the unicast connection, and a timer thread, which maintains a receive-side timer to handle block losses at the end of a data product transmission.
5. Responds to a notification from the FMTP retransmission request thread about a missed data-product (due to sender-side retransmission period timeout) by requesting the product from its upstream LDM7 over its Layer-3 DRFSM data-plane TCP connection.
6. Requests the backlog, if any, of data-products missed since the end of the previous session from its upstream LDM7 over its Layer-3 control-plane TCP connection.

Upstream LDM7 process This process manages a subscription request received from its corresponding downstream LDM7 process by:

1. Executing authorization actions: the top-level `ldmd` process requires that the hostname or IP address associated with a downstream LDM match an ALLOW entry in the upstream LDM's configuration file. This process ensures that only authorized hosts can obtain the information necessary to receive multicast LDM feeds. Additionally, the multicast LDM process (mLDM) performs a second authorization step as described below.
2. Ensuring that the FMTP-using mLDM process is running for the feed requested by the downstream LDM7 process, and if not, initiating a new mLDM process for the feed. The upstream LDM7 process passes the following parameters to the mLDM process: (i) IP address and UDP port number of the multicast group; (ii) Private IP address assigned to the VLAN at the sender; and (iii) Netmask assigned to the feed multipoint VLAN.
3. Invoking the Internet2 Advanced Layer-2 Service (AL2S) Open Exchange Software Suite (OESS) client to request an addition of the subscriber's VLAN to the

feed's multipoint VLAN. The parameters provided in this request to the OESS client include the feed-dependent publisher's SPS identifier and publisher core-network switch and port information contained in the publisher configuration file along with the subscriber's SPS identifier and the subscriber's core-network switch and port information that was received in the subscription request.

4. Responding to the subscription request with information listed above so that the downstream LDM7 process can configure the private IP address for the VLAN, and join the multicast group to start receiving the feed.
5. Acting as a backstop for data-products that the FMTP layer in the downstream LDM7 process was unable to successfully receive because the FMTP sender-side retransmission period expired, or for a backlog of data products. These missed and backlog products are delivered over the DRFSM control-plane TCP connection.

Multicast LDM (mLDM) process The mLDM process runs multiple FMTP threads as shown in Fig. 3. The FMTP multicast thread divides data-products into fixed-sized blocks, and sends these blocks to the UDP port with the multicast IP address that is configured on all subscribers on the multipoint VLAN receiving the feed. The timer thread is responsible for maintaining an FMTP sender-side retransmission timer for each product. The coordinator thread is a supervisor that accepts incoming TCP connections and forks a new retransmission thread for each subscriber. The retransmission thread is responsible for handling retransmission requests from the associated subscriber. Therefore, on the sending side, there will be $3 + N$ threads, where N is the number of subscribers.

The mLDM process has additional functions. First, it assigns private IP addresses from the allocated subnet to each receiver's VLAN on its L2-network NIC so that all nodes in the multipoint VLAN have the same subnet ID and netmask. Second, it responds with the assigned private IP addresses to `GetClientAddr` requests from the upstream LDM process so that the latter can send the assigned private IP address in its subscription response to the subscriber (the mLDM process acts as a DHCP server for private address assignment on the feed-dependent multipoint VLAN). Third, it handles authorization requests from the FMTP Coordinator thread. Authorization is based on the private IP addresses assigned to receivers for their VLANs. Because the FMTP Coordinator thread will receive a TCP SYN segment from this private IP address, the Coordinator thread passes this IP address to the Authorizer component of the mLDM process. The latter checks whether this private IP address is in its set of assigned addresses, and if so, the Authorizer approves the request. Finally, it executes disconnect actions when a receiver stops feed reception.

2.2.3 Utility program `vlanUtil`

On the sending side, the `vlanUtil` utility is executed when the LDM7 server starts. It executes `ip` commands to statically configure virtual interfaces (VLANs) and assign private IP addresses to the VLANs for all the feeds. It then executes the Linux `tc` utility to limit the sending rate and sending-buffer size. While this utility configures VLANs, IP addresses and rates when the LDM7 server starts, the mLDM process only starts sending data products for a feed when the first subscription is approved.

On the receiving side, the `vlanUtil` utility can only be executed by the downstream LDM process after a subscription reply for a feed is received. The `ip` command in `vlanUtil` takes the assigned private IP address received in the subscription reply to configure its virtual interface (VLAN).

3 LDM7 performance monitoring system

An LDM7 performance monitoring system was designed and implemented to collect statistics on the operation of an LDM7 data-distribution network, and offer LDM7 administrators and users a Graphical User Interface (GUI) for visualization. This section describes the performance monitoring-system's software architecture (in Sect. 3.1), and the performance metrics collected (in Sect. 3.2).

3.1 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 in an *LDM7-rtstats database*, and offers visualization services for users through a *dashboard*.

Figure 4 illustrates the overall architecture. The LDM7 performance monitor consists of (i) downstream LDM server, (ii) LDM pqact utility, (iii) LDM7-rtstats-decoder, (iv) LDM7-rtstats database, and (v) dashboard. The downstream LDM server can be an LDM6 or LDM7 server. A LDM 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

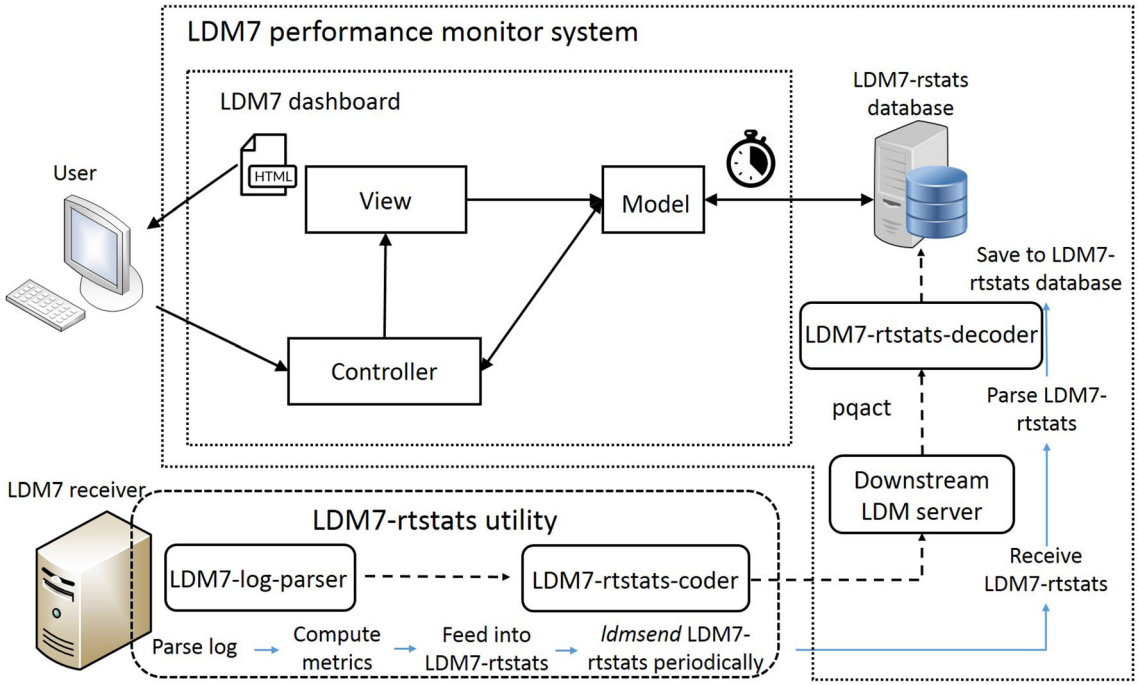


Fig. 4 LDM7 performance monitoring architecture

are saved in the LDM7-rtstats database. This database uses MongoDB,[3] 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.

In the LDM7-rtstats feed, an LDM7 publisher sends the rate of the multipoint path for each feed, while LDM7 subscribers send performance metrics computed from their LDM7 log files. Figure 5 shows a few sample lines for an LDM7-subscriber log file. Table 2 lists and explains each field of these log lines. The last field, msg, includes the product-delivery mechanism, delay, index, retrans, and product information.

Each product-delivery mechanism is assigned a keyword. Specifically, LDM7 uses the keyword "Received" for products delivered by FMTP, "Inserted" for products that required retransmissions via the LDM6-backstop mechanism (in which LDM6 is used to retransmit a product that could not be delivered via LDM7 before the FMTP retransmission timer expired), and "Backlog" for products delivered in a backlog-clearance process. Thus, we define two file types as follows: (i) *backstop-needed files* are data products that required LDM6-backstop retransmissions; (ii) *FMTP-received files* are data products that include both *Multicast-itself-sufficient* files that 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 *FMTP-retx-needed* files that are data products that were received by FMTP but required one or more FMTP block retransmissions.

Delay, index and retrans are three sub-fields that are relevant to FMTP. The sub-field "delay" is the file latency incurred in delivering the file from the publisher to the subscriber; it is computed at the subscriber by subtracting the BOP timestamp of this product placed at the publisher from the time instant at which product reception is completed (FMTP retransmissions, if required, are completed). Servers run Network Time Protocol (NTP) to synchronize clocks. The sub-field "index" represents the file index assigned by FMTP sender to ensure sequential delivery, and the sub-field "retrans" is the number of FMTP-retransmitted blocks, where each FMTP-retransmitted block has 1448 bytes FMTP of payload. The product information ("info") sub-field includes file size (in bytes), the time instant (also called product creation-time) when this file was first injected into the IDD system, feedtype of the file (e.g., NGRID, HDS), and other information (which are not related to LDM7 performance metrics). The size S_i of file i is obtained from the product size reported in the info subfield, and latency L_i of file i is obtained from the delay subfield of the msg field. The metric B_t is computed by adding the values in the retrans sub-field of the msg field of all FMTP-received files.

[3] https://www.mongodb.com/

```
20200907T235958.434607Z ldmd[18901]                MldmRcvr.c:eop_func:262            INFO  Received: {delay: 0.0224119 s, index: 1543
6661, retrans: 0, info: "     40199 20200907235958.328889 NGRID 68842374  LODZ65 KWBG 072300 !grib2/ncep/RUC2/#130/202009072300F013/VVEL
/650 hPa PRES"}
20200907T235958.439532Z ldmd[18901]                MldmRcvr.c:eop_func:262            INFO  Received: {delay: 0.0262699 s, index: 1543
6662, retrans: 0, info: "     88741 20200907235958.342930 NGRID 68842375  LKDZ98 KWBG 072300 !grib2/ncep/RUC2/#130/202009072300F013/VIS/
0 - NONE"}
20200907T235958.439824Z ldmd[18901]                MldmRcvr.c:eop_func:262            INFO  Received: {delay: 0.0232526 s, index: 1543
6663, retrans: 0, info: "       488 20200907235958.346540 NGRID 68842376  LMDZ98 KWBG 072300 !grib2/ncep/RUC2/#130/202009072300F013/WXTP
/0 - NONE"}
20200907T235958.441724Z ldmd[18901]                MldmRcvr.c:eop_func:262            INFO  Received: {delay: 0.0228160 s, index: 1543
6664, retrans: 0, info: "     38088 20200907235958.361659 NGRID 68842377  LUDZ17 KWBG 072300 !grib2/ncep/RUC2/#130/202009072300F013/UGRD
/175 hPa PRES"}
20200907T235958.444883Z ldmd[18901]                MldmRcvr.c:eop_func:262            INFO  Received: {delay: 0.0248404 s, index: 1543
6665, retrans: 0, info: "     57627 20200907235958.383733 NGRID 68842378  LTDZ86 KWBG 072300 !grib2/ncep/RUC2/#130/202009072300F013/TMP/
120-90 hPa PDLY"}
20200907T235958.578415Z ldmd[18901]                MldmRcvr.c:eop_func:262            INFO  Received: {delay: 0.0223855 s, index: 1543
6666, retrans: 0, info: "     74531 20200907235958.498687 NGRID 68842379  LVDZ96 KWBG 072300 !grib2/ncep/RUC2/#130/202009072300F013/VGRD
/0 - MWSL"}
20200907T235958.580593Z ldmd[18901]                MldmRcvr.c:eop_func:262            INFO  Received: {delay: 0.0217099 s, index: 1543
6667, retrans: 0, info: "     39267 20200907235958.523152 NGRID 68842380  LODZ60 KWBG 072300 !grib2/ncep/RUC2/#130/202009072300F013/VVEL
/600 hPa PRES"}
```

Fig. 5 A sample from an LDM7-subscriber log file

Table 2 Format of LDM7 log messages

Field name	Meaning	Example
time	Creation-time of the log message (which is just after the time instant at which product is inserted into product queue at the receiver) in the form YYYYMMDDThhmmss.uuuuuuZ	20200907T235958.434607Z
proc	Identifier of the process in the form id[pid]	ldmd[2613]
loc	Code location where the log message was created in the form file:func, where file and func are, respectively, the names of the file and function that generated the message	*MldmRcvr.c* : *eop*$_f$*unc* : 262
level	Logging-level (i.e., priority) of the message. One of DEBUG, INFO, NOTE, WARN, or ERROR	INFO
msg	Actual message given to one of the logging functions, e.g., to create a line in a log file, KEYWORD: delay: , index: , retrans: , info: " "	Received: delay: 0.0224119 s, index: 15436661, retrans: 0, info: " 40199 20200907235958.328889 NGRID 68842374 LODZ65 KWBG 072300 !grib2/ncep/RUC2/#130/202009072300F013 / VVEL /650 hPa PRES"

3.2 Metrics

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 [17]) 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:

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

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

$$\mathbb{T}_t^{retx} = \frac{\mathbb{S}_t' - \mathbb{S}_t''}{\mathbb{L}' - \mathbb{L}_t''} \tag{3}$$

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.

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 ($\mathbb{F}_t^{size}$):

$$\mathbb{F}_t^{count} = \frac{N_t'}{N_t} \times 100\% \tag{4}$$

$$\mathbb{F}_t^{size} = \frac{\mathbb{S}_t'}{\sum_{i \in N_t} S_i} \times 100\% \tag{5}$$

where N_t is the number of LDM7-received files (backstop-needed files and 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.

3.2.1 Multicast packet loss rate (MPLR)

Multicast packet loss rate is measured as a percentage of packets lost with respect to packets sent. For the specific application, LDM7, we define MPLR ($\mathbb{R}_t^{mc}$) to quantify the multicast packet loss by measuring the requested FMTP block retransmissions. Since the LDM7 subscriber would request every lost/undelivered multicast packet. The equation is formed as follow:

$$\mathbb{R}_t^{mc} = \frac{B_t * 1448}{\mathbb{S}_t'} * 100\% \tag{6}$$

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, TCP and IP headers are 12, 8, 20, and 20 bytes, respectively; to avoid a multicast block from requiring two TCP 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.

4 Trial deployment over multi-domain SDN

The trial deployment was feasible because the two types of network services required by LDM7 are both supported on university campus networks, RENs and Internet2, the US-wide REN. The Unidata IDD project distributes meteorology data mostly to universities, hence our trial WAN deployment was in this environment. 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. Consequently, the scale of the trial deployment was limited due to the high cost of human resources necessary to implement the regional and campus segments of the static VLAN. For example, it took almost two years to create an eight-campus trial deployment. Also, it should be noted that because of LDM7 and the DRFSM architecture allowing subscribers to dynamically join and leav the multicast group, theoretically, the LDM7 system is scalable. Currently, the UCAR uses LDM6 to distribute meteorological data to 581 hosts in 217 domains in Unidata's IDD system [4]. And, we tested the proposed multicast solution over GENI and Chameleon testbed with 24 subscribers and 16 subscribers, respectively [9, 10]. The goal of this trial deployment was to test the DRFSM architecture in a production, WAN multi-domain setting.

The DRFSM architecture requires dynamic provisioning because subscribers add and delete subscriptions to different feeds, and each feed should have its own multipoint VLAN for bandwidth and subscriber 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. 6, we deployed LDM7 servers at eight university campuses/institutions: (1) University Cooperation for Atmospheric Research (UCAR), (2) University of Virginia (UVA), (3) University of Wisconsin (UWisc), (4) University of Maryland (UMD), (5) University of Utah (U.Utah), (6) University of Missouri (UMiss), (7) University of Washington (UWash), and (8) University of California San Diego (UCSD).

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 UCAR, is connected via a 100 GE link to the Starlight Internet2 PoP in Chicago, for legacy reasons. This connection makes the UWisc, UVA, and UMD LDM7 servers the closest in terms of Round-Trip Time (RTT) to the 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 Fig. 1). 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 As shown in Fig. 1, SPSs are provisioned from the sending and receiving hosts through edge networks to the DMPS core-network switch ports. The publisher then uses its SDN Controller Clients (SCC) to send signaling

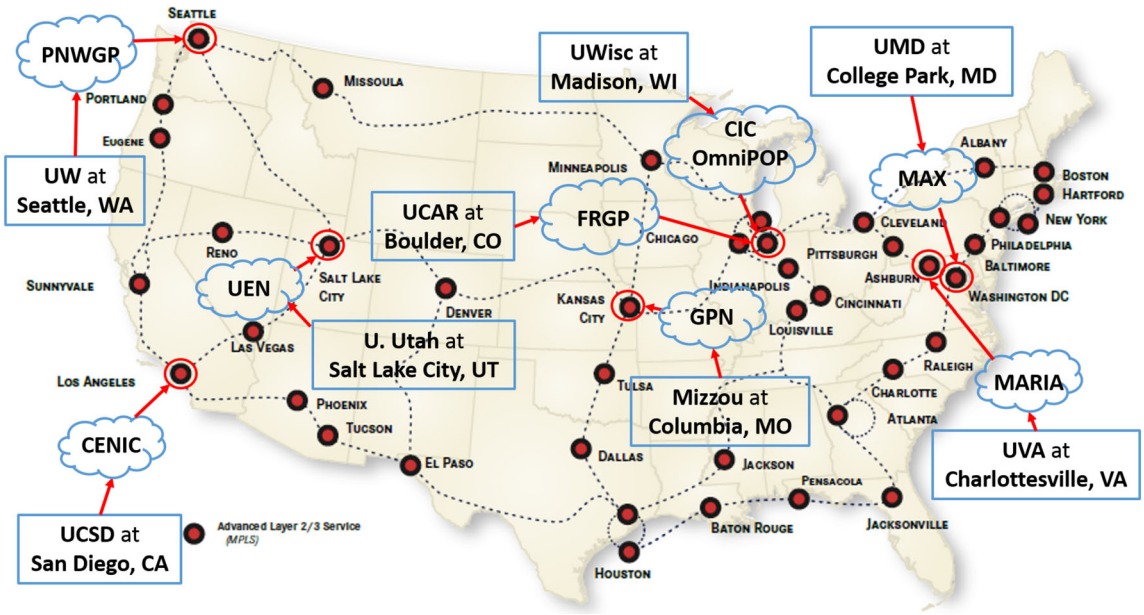


Fig. 6 Trial deployment of LDM7 across Internet2

messages to the core-network SDN controller to request the dynamic configuration of connecting/disconnecting SPSs in the DMPS network. 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.

Figure 7 shows an installation of the DRFSM architecture, with a publisher at UCAR, and subscribers at the UVA and UMD. MAX is the regional REN provider for UMD. VLAN segments are manually provisioned for the LDM7 servers in the three sites to their corresponding Internet2 router/switch ports, e.g., VLAN III from UVA. This 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.

Figure 7 also illustrates how Internet2's AL2S 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

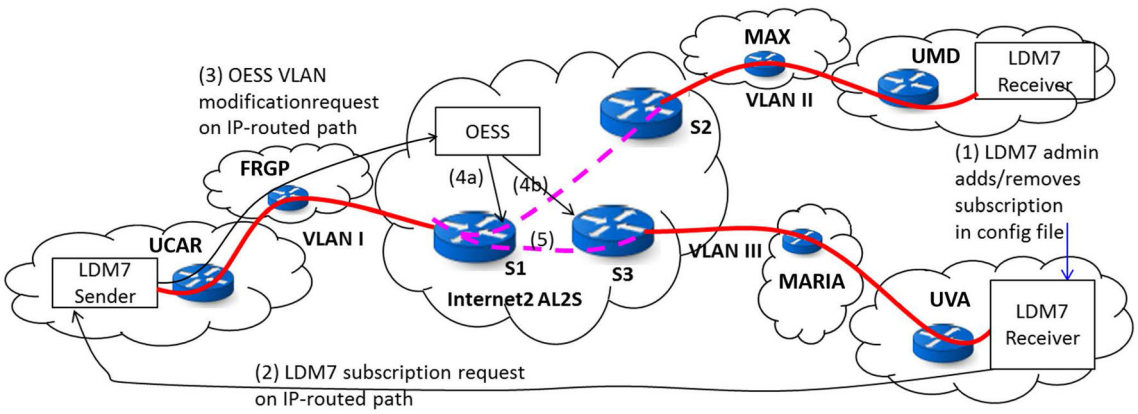


Fig. 7 An installation of DRFSM architecture on the LDM7 trial deployment; black arrows: control-plane messages; red lines: provisioned VLAN segments; magenta dashed lines: dynamic MPLS paths

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. 7. Step (2) shows the LDM7 subscription request being sent on the IP-routed path. The LDM7 publisher 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. 7, 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.

VLAN rate limit settings Multicast data products in a continuous stream using UDP could impact general-purpose flows if the rate required by these feeds was high. This situation is not the case with the Unidata IDD meteorology feeds. In aggregate, all 30 feeds require only 2 Gbps on average. If all 580 hosts subscribed to all 30 feeds, then UCAR would require access link rates greater than 1 Tbps just for this Unidata IDD project because in the current LDM6 solution, individual TCP connections are used from the publisher to each subscriber (current topology uses application layer multicast in which some receivers also host Upstream LDM servers to relay data, and not all hosts request all 30 feeds).

Most of the feeds are continuous in the sense that data products that need dissemination arrive 24/7, but the data arrival rate is not constant. For example, the one-hour set of data products from the NGRID feed used in these experiments has data products with a median size of 31290 B and IQR of 118618 B, and median file inter-arrival time of 4.16 ms and IQR of 23.79 ms.

All the universities and their regional RENs have 100 Gb/s links and the LDM7 servers have 10 GE NICs. We experimented with low rate settings, such as 20 Mbps to 60 Mbps for several feeds, and found that products could arrive in bursts requiring large sender-size buffers to prevent packet loss. However, large buffers can add significant latency. For example, if a 300 MB buffer is used at the sender to absorb bursts, with a 20 Mbps VLAN, the added latency could be as high as 2 mins. When we increased rates to 1 Gbps per VLAN (i.e., per feed), this sender-buffering delay was eliminated, but more retransmissions were required. Presumably the higher rate was overwhelming small buffers typically found in enterprise and data-center switches, or the receiver UDP buffer was being overwhelmed. A sweet spot of 500 Mbps was reported in a recent paper [10].

This higher per-feed rate requirement to reduce latency diminishes the value of multicasting. For example, with 30 feeds, if 500 Mbps was required from a latency consideration for all 30 feeds, UCAR would still need 15 Gbps, which is better than the projected 1 Tbps, but still fairly high. Therefore, we explored the use of Linux `tc` queuing disciplines in which multiple VLANs could be configured to use bandwidth borrowing. A combination of Hierarchical Token Bucket (HTB) and Bytes First In First Out (BFIFO) queuing disciplines was used, and two queues, one for multicast and the second for retransmissions, was created for each VLAN. Because one VLAN was created for each feed, bandwidth borrowing was allowed. This reduced the total required bandwidth, and the possibility of packet losses stemming from the use of too high a rate by one feed (if multiple other feeds were silent) was avoided by setting the `ceil` parameter, which limits the maximum rate used to serve packets from a class.

5 Experimental evaluation

Section 5.1 describes the experiments executed on the trial deployment. Sections 5.2 to 5.4 presents our numerical results of three experiment sets.

5.1 Setup

For our experiments, we created a multipoint VLAN in Internet2 AL2S network, which connects the eight servers geo-located in the trial deployment, seen in Fig. 6. Each of the deployed servers has at least 64 GiB RAM, 500 GB disk space and two network interfaces. One 1/10Gbps Ethernet (GbE) NIC connects to the general-purpose campus network for L3 IP-routed services, and the other 10GbE NIC connects to a switch through which VLANs are provisioned to the nearest Internet2 switch port via the campus network and regional REN. One more server at UVA campus network runs the LDM7 performance monitor to collect the subscribers' ldm7-rtstats feeds and display the results on the dashboard. The software used in our

experiments consists of: (i) LDM7 application, and (ii) LDM7 performance monitoring system.

Table 3 shows the experimental parameters that influence output metrics including feedtypes (file-streams), RTT on publisher-to-subscriber paths, packet loss rate, and VLAN rate limit [10]. We chose the NGRID feed (numerical model output from NOAA) as the file-stream to distribute, and a specific publisher (UCAR), but varied the VLAN rate limits, the set of subscribers (among the six subscribers, UVA, UWisc, UMD, U.Utah, UWash, and UCSD), and the simulated packet loss. The publisher sending-buffer size (`tc`-layer buffer) was set to a large enough value (e.g., 600 MB in our experiments) to prevent packet drops at the sending side. The `tc` statistics confirmed that no packets were dropped due to publisher sending-buffer overflows. A 100 MB receiving buffer was set to avoid buffer overflow. The maximum VLAN rate limit used in the experiments was 500 Mbps. A higher VLAN rate would have made the performance evaluation more difficult. This paper mainly discusses propagation delay, emission (transmission) delay, sending-buffer queuing delay, and synchronization offsets. The processing delay and switch/router buffering delay are ignored because their impact on the overall delay is negligible in a WAN [18]. For a multicast-itself-sufficient file i, latency L_i was described as follows:

$$L_i = L_{prop} + S_i/r + L_{sending-buffer} + L_{sync-offset}, \quad (7)$$

where L_{prop} is the one-way propagation delay from the publisher to the subscriber, S_i is the size of the file i, r is the VLAN rate limit, $L_{sending-buffer}$ is the sending-buffer queuing delay and $L_{sync-offset}$ is the clock synchronization offset. Although NTP is used in all LDM servers to synchronize clocks, there can still be synchronization offsets on the order of milliseconds [19]. Such offsets are comparable to the emission delay, S_i/r, for the megabyte-size files in the feed [20] when the VLAN rate is 5000 Mbps.

We ran three sets of experiments. The goal of the first set of experiments was to verify the feasibility of the latest LDM7, a DRFSM implementation, and the LDM7 performance monitoring system. A second set of experiments was performed to evaluate the performance of LDM7. Lastly, we ran a third set of experiments to compare the performance of LDM7 with LDM6 over the trial deployment.

5.2 Experiment Set 1: feasibility of the DRFSM solution

The UCAR LDM7 publisher subscribed to a live NGRID file-stream from the Unidata IDD system, and then multicast the NGRID file-stream to six LDM7 subscribers: UVA, UWisc, UMD, U.Utah, UWash, and UCSD. During the NGRID file-stream distribution, we limited the VLAN rate to 40 Mbps. The reason for rate-limiting at the publisher is explained in Sect. 2. All of the subscribers sent their ldm7-rtstats feeds to the LDM7 performance monitor.[4] Three metrics defined in Sect. 3.2 (throughput, FFDRs, and MPLR) were used to verify the feasibility of the DRFSM implementation.

Figure 9 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.[5] Each server (publisher or subscriber) that joins the meteorological data distribution is geo-located on a U.S. map. Servers 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. 9, there are seven green points, which indicates the UCAR publisher and six LDM7 subscribers, which shows the subscribers receiving an NGRID feed from UCAR. The two red points represent Rutgers and UMiss, which failed to join the multicast group. The red lines between the green

Table 3 Values for experimental parameters

Parameters	Value
Servers	{the University Cooperation for Atmosh- peric Research (UCAR), University of Virginia (UVA), University of Wisc- son (UWisc), UMD, University of Utah (U.Utah), University of Washington (UWash), University of California at San Diego (UCSD)}
RTT	from UCAR to {UVA: 37.2, UWisc: 39.0, UMD: 36.5, U.Utah: 51.7, UWash: 61.8, UCSD: 51.7} ms
Simulated packet loss	{ 0; 1; 5; 10; 15; 20}%
VLAN rate limits	{20; 30; 40; 50; 100; 500} Mbps

points, such as the publisher (UCAR) and the UVA LDM7 subscriber represent the logical connection (link) between the two servers. A "click" action on a link triggers a time-series plot with per-minute information for the link. Figure 8, for example, shows the per-minute throughput of NGRID feed from UCAR to UVA in last one hour. From the Fig. 8, we can see over the past one hour the per-minute throughput of the UVA subscriber varied between 5 Mbps and 35 Mbps with a 40 Mbps VLAN rate limit. The most reasonable explanation for the difference is sending-buffer queuing delay.

Table 4 shows the three metrics and other information included in the ldm7-rtstats feeds. First, We used two FFDRs and MPLR to evaluate the reliable capability of LDM7. Their 100% values indicate that all 45642 files were delivered by the FMTP data-plane protocol without requiring the LDM6-backstop mechanism. As previously noted, FMTP uses a TCP retransmission mechanism to deliver data-blocks that were not received via the UDP multicast. The MPLR row shows the proportion of such retransmitted data blocks, which is also the proportion of multicast packets that weren't received. Although the retransmission mechanism reliably delivered the lost multicast packets, it incurred significant delays in the production WAN setting, due to the RTTs on the path from the publisher to each subscriber. Interestingly, the subscribers of UVA, UMD, and Uwisc had the same number of FMTP block retransmissions and files that needed FMTP retransmissions. Table 5 shows the index of the files that required retransmitted data-blocks and indicates that the same three files required retransmissions. A reasonable explanation is that the multicast packet loss happened in the publisher's regional REN.

Conclusively, we observe that the overall DRFSM solution worked well on the trial deployment. The metric of average throughput indicates that different subscribers achieved different values. The throughput was affected by differences in product-latency, whose components are: (i) processing delays, (ii) sending-buffering delays incurred because of the sender `tc` rate limiting used in our experiments, (iii) emission (transmission) delays, (iv) propagation delays (roughly half of RTT), (v) switch/router packet queuing delays, and (vi) retransmission delays. Processing delays are typically negligible when compared to RTT in the WAN setting. All subscribers have the same sending-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., 18 ms between UCAR and UVA). Given the high link capacities in the production RENs on which our trial was deployed, switch/router packet queuing delays should be small.

5.3 Experiment set 2: analysis of LDM7 performance

To analyze the performance of LDM7 over the trial deployment, we collected one-hour NGRID products, 03:00 to 04:00 UTC on Jul. 12, 2020, and replayed the data repeatedly (using the LDM utility `pqinsert`) in multiple experiments to avoid differences in incoming file-stream product sizes or inter-arrival times. This feed was sent with various VLAN rate limits. Figure 10 shows per-file sizes and arrival times for this collected set of 45703 files.

Of the five variables listed in Table 3, we selected one publisher (UCAR) and four subscribers (UVA, UWisc, U.Utah, and UWash) and did not attempt to modify RTTs on the paths from the publisher to each subscriber. Then, we investigated the impact of the other three parameters, VLAN rate limit, packet loss rate and LDM feed types, on the performance of LDM7.

5.3.1 Impact of VLAN rate limit

Figure 11 shows average FMTP throughput for the NGRID feedtype at four subscribers. We replayed the one-hour NGRID file-stream, whose traffic pattern followed Fig. 10, from UCAR to the four subscribers with varying VLAN rate limits of 20, 30, 40, 50, 100, and 500 Mbps. Generally, as the VLAN rate limit increased, the throughput of each subscriber increased. But different subscribers achieved different throughput at the same VLAN rate limit. This result is consistent with the conclusion that for different subscribers, the differences in throughput are mainly due to propagation delay, in particular, the difference is larger with a higher VLAN rate limit. For example, with a 20 Mbps VLAN rate limit, the average throughput of UWash is 7.65% less than the average throughput of UVA (UWash: 10.75 Mbps, UVA: 11.64 Mbps), but the difference increases to 38.53% at 500 Mbps, where the average throughput of UWash is 36.22 Mbps with a 500 Mbps VLAN rate limit and the average throughput of UVA is 58.92 Mbps.

5.3.2 Impact of packet loss rate

Figure 12 shows the impact of the simulated network packet loss on the performance of LDM7, where the publisher, UCAR, distributed the file-stream at a 100 Mbps VLAN rate limit to four subscribers: UVA, UWisc, U.Utah, and UWash. Figure 12, uses solid lines to present the throughput performance, and the dashed lines to show how MPLR varies with the simulated network packet loss

[4] http://idc-uva.dynes.virginia.edu:3000/.

[5] https://d3js.org/.

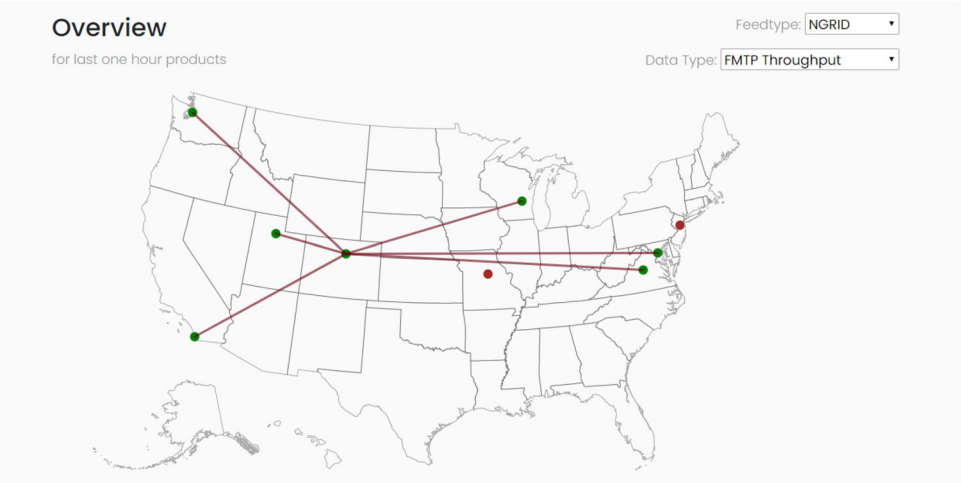


Fig. 8 LDM7 performance monitoring system dashboard

Table 4 Experiment Set 1: Statistics for files received by the UVA, UWash, UMD, UWisc and UCSD LDM7 subscribers; 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

Table 5 Experiment Set 1: Files requested FMTP retransmissions

Product index	8931043	8925724	8925341
Size	12274	11676	11107
Number of block retransmissions	9	8	4
Subscribers requested retransmissions	UVA, UMD, UWisc, UWash, UCSD, Utah	UVA, UMD, UWisc, UWash, UCSD, Utah	UVA, UMD, UWisc, UWash, UCSD, Utah

rate. Firstly, we observe that with the increase of simulated network packet loss rate, the throughput of the four subscribers decreases. This decrease is because the total delay incurred by subscribers requesting to retransmit the missing multicast UDP datagrams. Secondly, the MPLR of four subscribers is higher than the simulated network packet loss rate. This is reasonable because the MPLR is defined by using the number of retransmission requests. Thus, in addition to the multicast packets lost in the network, the multicast packets dropped by the application would also be retransmitted to the subscribers. For example, if a subscriber missed the FMTP BOP message of the multicast file, the subscriber would drop the subsequent FMTP-UDP datagrams before it received the retransmitted BOP message. The number of retransmission requests consists of the number of multicast packets lost in the network (in this case, they are dropped by the Linux utility `iptables` at the subscriber) and the number of multicast packets

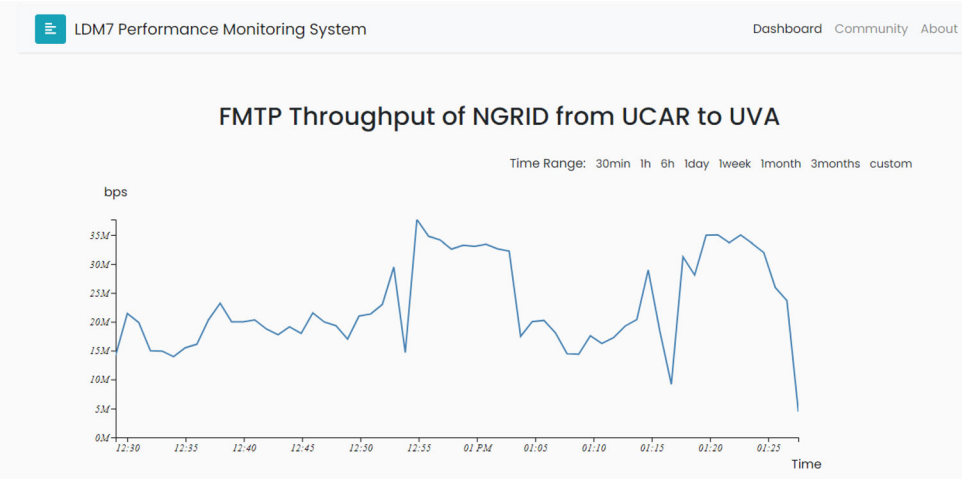


Fig. 9 1-hour throughput of NGRID from UCAR to UVA on Dashboard

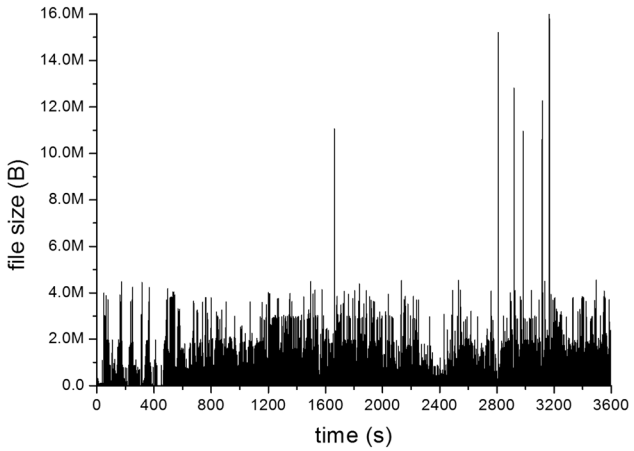


Fig. 10 NGRID, 03:00-04:00 UTC, on Jul. 12, 2020

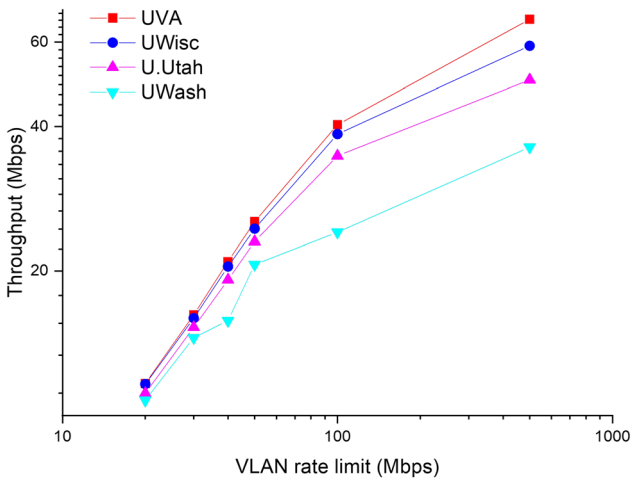


Fig. 11 LDM7 Throughput varying VLAN rate limits

dropped by the LDM7. Consequently, we can conclude that the impact of FMTP BOP message loss depends mainly on the VLAN rate limit, file size, and round-trip-time (RTT) on the path between the publisher and the subscriber. Furthermore, if the transmission delay of the multicast file is less than the delay of retransmitted BOP message, the subscriber would request to retransmit the whole file. In our experiments, the MPLR consists of two cases, which are (i) BOP message lost in the network and (ii) BOP message successfully received but some FMTP datagrams lost. Thus, we can see the observed MPLR is much higher than the simulated network packet loss rate. Thirdly, the degradation of throughput was due to retransmission delay, which was incurred by request missing data blocks. Because a retransmission request for a missing data-block incurs a RTT penalty at least, such requests have a significant effect on throughput. Thus, we observed a large throughput drop when we simulated a 5% packet loss rate. For UVA, the throughput decreased 20.3 Mbps (from 36.4 Mbps to 16.1 Mbps), while there was only a 2.1 Mbps decrease (from 12.4Mpbs to 10.3 Mbps) when the simulated packet loss increased from 15% to 20%.

5.4 Experiment set 3: performance comparison between LDM6 and LDM7

The one-hour collected NGRID products (shown in Fig. 10) were used in Experiment Set 3 to evaluate and compare the performance of the DRFSM incorporated LDM7 and the current LDM6. The comparison between the DRFSM solution and the unicast-based solution was executed with the same VLAN rate limits and sets of subscribers as in Experiment Set 2.

This set of experiments had two goals: (i) compare the average throughput of LDM6 and LDM7 under the same VLAN rate limits, and (ii) compare the VLAN rate limits for LDM6 and LDM7 that achieve the same throughput.

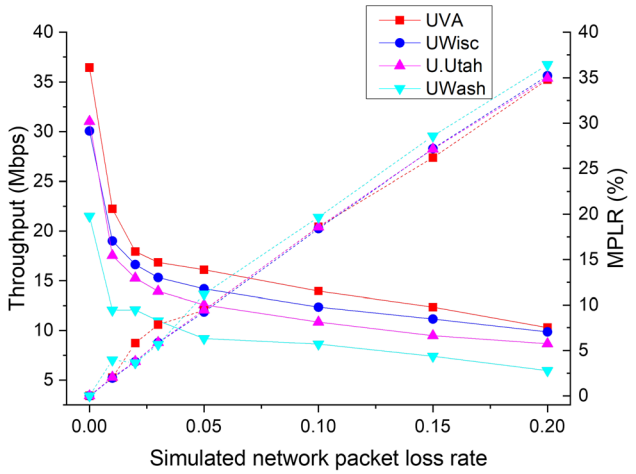


Fig. 12 LDM7 throughput and MPLR with various simulated network packet loss

The main parameters varied in Experimental Set 3 were the VLAN rate limit and set of subscribers.

First, we replayed the one-hour NGRID feed from UCAR to four subscribers, UVA, UWisc, Utah, and UWash, with varying VLAN rate limits of 50, 100, 200, 300, 400, and 500 Mbps via LDM6 and LDM7. Figure 13 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 rate limit increases, the differences between LDM7 throughput and LDM6 throughput increases. For example, with a 50 Mbps VLAN rate limit, the difference is 21.88 Mbps, while at 100 Mbps, the value increases to 31.62 Mbps. However, it stabilizes around 31 Mbps when we increase the VLAN rate limit to 100 Mbps. Furthermore, Fig. 13 shows that there is an approximately 90% bandwidth saving for LDM7 (with a 40 Mbps VLAN rate limit), compared to LDM6 (with a 400 Mbps VLAN rate limit) when achieving a 20 Mbps average throughput with four subscribers. This result verifies that LDM7 significantly reduces the need for bandwidth. The bandwidth savings is expected to increase as the number of subscribers increases.

Second, we take a close look at the throughput with a 100 Mbps VLAN rate limit. Figure 14a shows the throughput comparison of LDM6 and LDM7 (four subscribers) at 100 Mbps. LDM6-FS (four subscribers) shows throughput of LDM6 with four subscribers, while LDM6-SS (single subscriber) shows the throughput of LDM6 with a single subscriber. The first finding in Fig. 14a is that LDM7 achieved a 77.6% higher average throughput than LDM6 with a single subscriber (LDM6-SS). This improvement likely results from using UDP rather than TCP over a dedicated VLAN. This appears backwards: the figure shows that LDM6-FS has much worse throughput than LDM6-SS. One possible explanation is a larger sender-buffer queuing delay with multiple copies in LDM6-FS. Moreover, Fig. 14b presents the impact of the number of subscribers on the performance of LDM7 and LDM6 with a 100 Mbps VLAN rate limit. As the number of subscriber increases, the throughput of LDM7 varies only slightly, while the throughput of LDM6 decreases significantly. At UVA, in particular, with six subscribers, the

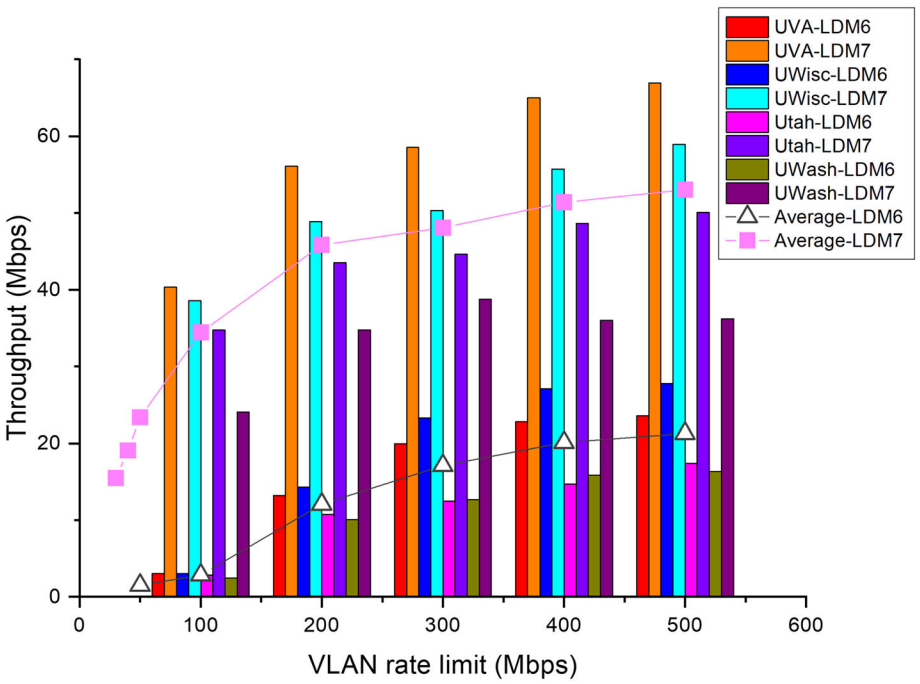
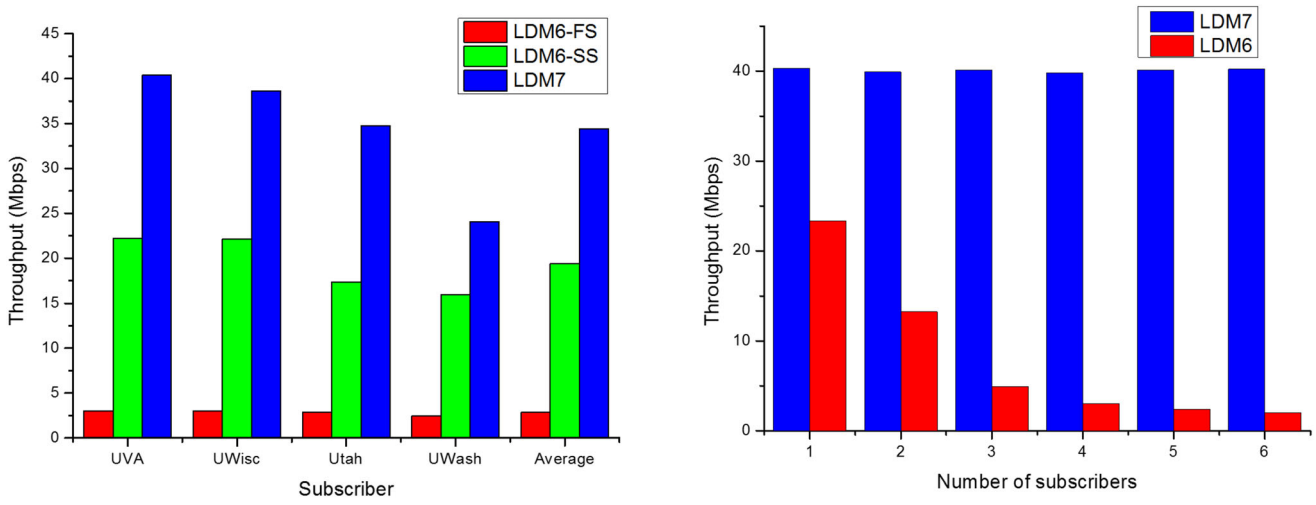


Fig. 13 Average throughput of LDM6 and LDM7 with four subscribers varying VLAN rate limits at UVA



(a) Throughput at various subscribers

(b) Throughput varying number of subscribers observed at UVA

Fig. 14 Experiment Set 3: Throughput comparison of LDM6 and LDM7 with a 100 Mbps VLAN rate limit

throughput of LDM7 (40.12 Mbps) is an almost 22-fold improvement to the throughput of LDM6 (1.79 Mbps).

6 Related work

Section 6.1 offers the reader background on early reliable multicast transport layer protocols. Section 6.2 describes the particular SDN controller used in this work. Section 6.3 reviews recent related work.

6.1 Reliable multicast transport protocols

Transport protocols are required to support reliable multicast on top of IP-multicast trees. Early proposals include Reliable Multicast Transport Protocol [21], and SRM by Sally Floyd [22]. The digital fountain solution by Luby et al. [23] and IETF's Asynchronous Layered Coding (ALC) [24] are designed for multicast to millions of receivers. These methods use layered coding transport, forward error correcting codes, and no reverse-direction messages from receivers to the sender. IETF's NACK-Oriented Reliable Multicast (NORM) [25] is designed for IP-multicast networks. NORM takes into account the need for data-plane congestion control, and the NORM sender keeps adjusting its sending rate based on Round-Trip Times (RTTs) and packet loss rates reported by each receiver.

6.2 Software-defined networks

Internet2's support for a L2-based path service required two components. First, the protocols for the control-plane were specified and standardized. These include Inter-Domain Controller Protocol (IDCP) [26] and the Open Grid Forum Network Service Interface Connection Services (NSI CS) version 2.0 [27]. Both protocols support inter-domain signaling for advance reservation and provisioning of rate-guaranteed dynamic L2 paths. Second, an SDN controller [28] must exist. Internet2 deployed an OESS server as the controller for its 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) [29]. User working group identifiers are used for access control allowing a regional REN to authorize a remote user group to connect to a particular Virtual LAN (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.

6.3 Recent related work

Solutions have been proposed to leverage SDN techniques to provide efficient and well-managed network-multicast services. Many of these solutions [30–32] 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) [33], and ECast [34]. 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. [35] 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) [36], ATHENA [37], and Datacast [38], but these are not readily extendible for WAN deployment. The DCCast solution [39] 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.

Solutions based on Peer-to-Peer (P2P), an example of which is Bit Torrent [], show the efficiency for transferring a single large file over the networks. However, for file-streams carrying new data, P2P will incur higher latency than our proposed DRFSM/LDM7 solution as they require multiple downloads before all subscribers can receive all blocks. One of the co-authors is exploring the potential of a hybrid multicast/P2P solution in which missed multicast blocks are recovered via the P2P network.

7 Conclusions

This work described a trial deployment of an L2 path-based network multicast solution and IP-routed service. Using this deployment, our experiments showed that file-delivery ratios and throughput metrics achieved in this DRFSM implementation met LDM7 application requirements. The deployment showed that capturing data using a performance dashboard to display key LDM7 performance metrics worked well. The comparison of LDM7 with LDM6 in the trial deployment showed an almost 22-fold throughput improvement with a 100 Mbps VLAN rate limit 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 gaps in multipoint path availability well enough for the LDM/IDD application that we tested, a new design is needed to alleviate the impact of DRFSM control-plane overheads.

Acknowledgements This work was supported by National Science Foundation award 1659174. We also wish to thank 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. The genesis and inspiration for this work was from our colleague, Malathi Veeraraghavan, who passed away in May 2020. She is greatly missed by all those that knew her.

Author contributions All authors contributed to the study conception and design. Especially, Prof. Malathi Veeraraghavan contributed significantly to this work before her passing away in 2020. Material preparation, data collection and analysis were performed by Yuanlong Tan. The first draft of the manuscript was written by Yuanlong Tan and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript, except Prof. Malathi Veeraraghavan. She did not have a chance to read the submitted version of this paper

Funding Funding was provided by National Science Foundation Award 1659174.

Data availability The datasets generated during and/or analysed during the current study are available from the corresponding author on reasonable request

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Ethical approval This work does not raise any ethical issues.

Informed consent None

References

1. Unidata Internet Data Distribution (IDD). https://www.unidata.ucar.edu/projects/idd/iddams.html
2. Local Data Manager (LDM). https://github.com/Unidata/LDM
3. LDM-6 Feedtypes. https://www.unidata.ucar.edu/software/ldm/ldm-current/basics/feedtypes/
4. Real-Time IDD Statistics. https://rtstats.unidata.ucar.edu/rtstats/stats.html
5. NOAA GOES Overview. https://noaasis.noaa.gov/GOES/goes_overview.html
6. Waitzman, D., Partridge, C., Deering, S.: Distance Vector Multicast Routing Protocol. RFC 1075, IETF (1988). http://tools.ietf.org/rfc/rfc1075.txt
7. Fenner, B., Handley, M., Holbrook, H., Kouvelas, I., Parekh, R., Zhang, Z., Zheng, L.: Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised). RFC 7761, IETF (2016). http://tools.ietf.org/rfc/rfc7761.txt
8. Ratnasamy, S., Ermolinskiy, A., Shenker, S.: Revisiting IP Multicast. In: Proceedings of the 2006 Conference on

Applications, Technologies, Architectures, and Protocols for Computer Communications, SIGCOMM '06, pp. 15–26. ACM, New York, NY, USA (2006). https://doi.org/10.1145/1159913.1159917
9. Chen, S., Ji, X., Veeraraghavan, M., Emmerson, S., Slezak, J., Decker, S.G.: A cross-layer Multicast-Push Unicast-Pull (MPUP) architecture for reliable file-stream distribution. In: Proceedings of the 2016 IEEE 40th COMPSAC, vol. 1, pp. 535–544 (2016). https://doi.org/10.1109/COMPSAC.2016.28
10. Tan, Y., Chen, S., Emmerson, S., Zhang, Y., Veeraraghavan, M.: Advances in Reliable File-Stream Multicasting over Multi-Domain Software Defined Networks (SDN). In: Proceedings of the 2019 28th International Conference on Computer Communication and Networks (ICCCN), pp. 1–11 (2019). https://doi.org/10.1109/ICCCN.2019.8847110
11. Li, J., Veeraraghavan, M., Emmerson, S., Russell, R.: File multicast transport protocol (FMTP). In: IEEE CCGrid, SCRAMBL workshop, pp. 1037–1046 (2015). https://doi.org/10.1109/CCGrid.2015.121
12. Berman, M., Chase, J.S., Landweber, L., Nakao, A., Ott, M., Raychaudhuri, D., Ricci, R., Seskar, I.: GENI: a federated testbed for innovative network experiments. Comput. Netw. **61**, 5–23 (2014)
13. Keahey, K., Riteau, P., Stanzione, D., Cockerill, T., Mambretti, J., Rad, P., Ruth, P.: Chameleon: a scalable production testbed for computer science research. In: Vetter, J. (ed.) Contemporary High Performance Computing: From Petascale toward Exascale Computational Science, vol. 3, 1st edn, chap. 5. Chapman & Hall/CRC, Boca Raton (2018)
14. Internet2 Network Connections. https://internet2.edu/network/state-and-regional-r-e-networks/
15. Moy, J.: OSPF Version 2. RFC 2328, IETF (1998). http://tools.ietf.org/rfc/rfc2328.txt
16. Rekhter, Y., Li, T., Hares, S.: A Border Gateway Protocol 4 (BGP-4). RFC 4271, IETF (2006). http://tools.ietf.org/rfc/rfc4271.txt
17. Jacob, B., Mudge, T.: Notes on calculating computer performance. University of Michigan, Tech. Rep. CSE-TR-231-95 (1995)
18. Yuan, B., Zhao, C.: Research on transmission delay of sd-wan cpe. In: Proceedings of the 2020 IEEE 20th International Conference on Communication Technology (ICCT), pp. 955–960. IEEE (2020)
19. Mills, D.L.: Internet time synchronization: the network time protocol. IEEE Trans. Commun. **39**(10), 1482–1493 (1991)
20. Ji, X., Liang, Y., Veeraraghavan, M., Emmerson, S.: File-stream distribution application on software-defined networks (SDN). In: Proceedings of the Computer Software and Applications Conference (COMPSAC), 2015 IEEE 39th Annual, vol. 2, pp. 377–386 (2015). https://doi.org/10.1109/COMPSAC.2015.23
21. Paul, S., Sabnani, K.K., Lin, J.C., Bhattacharyya, S.: Reliable multicast transport protocol (RMTP). IEEE J. Select. Areas Commun. **15**(3), 407–421 (1997). https://doi.org/10.1109/49.564138
22. Floyd, S., Jacobson, V., Liu, C.G., McCanne, S., Zhang, L.: A reliable multicast framework for light-weight sessions and application level framing. IEEE/ACM Trans. Netw. **5**(6), 784–803 (1997). https://doi.org/10.1109/90.650139
23. Byers, J.W., Luby, M., Mitzenmacher, M.: A digital fountain approach to asynchronous reliable multicast. IEEE J. Select. Areas Commun. **20**(8), 1528–1540 (2002). https://doi.org/10.1109/JSAC.2002.803996
24. Luby, M., Watson, M., Vicisano, L.: Asynchronous Layered Coding (ALC) Protocol Instantiation. RFC 5775, IETF (2010). http://tools.ietf.org/rfc/rfc5775.txt
25. Adamson, B., Bormann, C., Handley, M., Macker, J.: NACK-Oriented Reliable Multicast (NORM) Transport Protocol. RFC 5740, IETF (2009). http://tools.ietf.org/rfc/rfc5740.txt
26. Lake, A., Vollbrecht, J., Brown, A., et al.: Inter-domain Controller (IDC) Protocol Specification (2008)
27. Roberts, G., Kudoh, T., Monga, I., Sobieski, J., MacAuley, J., Guok, C.: NSI Connection Service v2.0. GFD. 212 pp. 1–119 (2014)
28. Kreutz, D., Ramos, F.M.V., Veríssimo, P.E., Rothenberg, C.E., Azodolmolky, S., Uhlig, S.: Software-defined networking: a comprehensive survey. Proc. IEEE **103**(1), 14–76 (2015). https://doi.org/10.1109/JPROC.2014.2371999
29. Rosen, E., Viswanathan, A., Callon, R.: Multiprotocol Label Switching Architecture. RFC 3031, IETF (2001). http://tools.ietf.org/rfc/rfc3031.txt
30. Sheu, J.P., Chang, C.W., Chang, Y.C.: Efficient multicast algorithms for scalable video coding in software-defined networking. In: Proceedings of the 2015 IEEE 26th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), pp. 2089–2093. IEEE (2015)
31. Lin, Y.D., Lai, Y.C., Teng, H.Y., Liao, C.C., Kao, Y.C.: Scalable multicasting with multiple shared trees in software defined networking. J. Netw. Comput. Appl. **78**, 125–133 (2017)
32. Yu, P., Wu, R., Zhou, H., Yu, H., Chen, Y., Zhong, H.: Multicast routing tree for sequenced packet transmission in software-defined networks. In: Proceedings of the 8th Asia-Pacific Symposium on Internetware, pp. 27–35. ACM (2016)
33. Zhu, T., Wang, F., Hua, Y., Feng, D., Wan, Y., Shi, Q., Xie, Y.: MCTCP: Congestion-aware and robust multicast TCP in software-defined networks. In: Proceedings of the 2016 IEEE/ACM 24th International Symposium on Quality of Service (IWQoS), pp. 1–10. IEEE (2016)
34. Zhang, X., Yang, M., Wang, L., Sun, M.: An OpenFlow-enabled elastic loss recovery solution for reliable multicast. IEEE Syst. J. **12**(2), 1945–1956 (2018)
35. Desmouceaux, Y., Clausen, T.H., Fuertes, J.A.C., Townsley, W.M.: Reliable multicast with B.I.E.R. J. Commun. Netw. **20**(2), 182–197 (2018). https://doi.org/10.1109/JCN.2018.000025
36. Shukla, S., Ranjan, P., Singh, K.: MCDC: Multicast routing leveraging SDN for Data Center networks. In: Proceedings of the 2016 6th International Conference - Cloud System and Big Data Engineering (Confluence), pp. 585–590 (2016). https://doi.org/10.1109/CONFLUENCE.2016.7508187
37. Mahajan, K., Sharma, D., Mann, V.: Athena: Reliable multicast for group communication in SDN-based data centers. In: Proceedings of the 2017 9th International Conference on Communication Systems and Networks (COMSNETS), pp. 174–181 (2017). https://doi.org/10.1109/COMSNETS.2017.7945374
38. Cao, J., Guo, C., Lu, G., Xiong, Y., Zheng, Y., Zhang, Y., Zhu, Y., Chen, C., Tian, Y.: Datacast: a scalable and efficient reliable group data delivery service for data centers. IEEE J. Select. Areas Commun. **31**(12), 2632–2645 (2013). https://doi.org/10.1109/JSAC.2013.131205
39. Noormohammadpour, M., Raghavendra, C.S., Rao, S., Kandula, S.: DCCast: Efficient Point to Multipoint Transfers Across Datacenters. In: Proceedings of the 9th USENIX Workshop on Hot Topics in Cloud Computing (HotCloud 17). USENIX Association, Santa Clara, CA (2017). https://www.usenix.org/conference/hotcloud17/program/presentation/noo-rmohammadpour

Yuanlong Tan received his B. Tech.degree from School of Information and Communication Engineering and MS degreefrom the State Key Laboratory of InformationPhotonics and Optical Communication atBeijing University of Posts andTelecommunications (BUPT), China. He iscurrently pursuing the Ph.D. degree with theCharles L. Brown Department of Electrical andComputer Engineering, University of Virginia(UVA). His research interests include softwaredefinednetworks, network multicast, filestreamdistribution, and optical networks.

Malathi Veeraraghavan (Fellow, IEEE, In Memoriam) receivedher B. Tech degree from Indian Instituteof Technology (Madras) in 1984, and MSand Ph.D. degrees from Duke Universityin 1985 and 1988, respectively. After aten-year career at Bell Laboratories, sheserved on the faculty at PolytechnicUniversity, Brooklyn, New York from1999-2002, where she won the Jacobsaward for excellence in education in 2002.She served as Director of the Computer Engineering Program at Universityof Virginia (UVA) 2003–2006. In 2007, she was promoted to a Professor with the Charles L. Brown Department of Electrical and Computer Engineering, UVA. In 2020, she was elevated to IEEE Fellow, "forcontributions to control-plane architectures, signal protocols and hybridnetworks." She holds 30 patents, has over 130 publications and has received eight Best paper awards. She served as the Technical Program Committee Co-Chair for the High-Speed Networking Symposium in IEEE ICC 2013,as Technical Program Committee Chair for IEEE ICC 2002 and AssociateEditor for the IEEE/ACM Transactions on Networking. She was GeneralChair for IEEE Computer Communications Workshop in 2000, and servedas an Area Editor for IEEE Communication Surveys. She served as Editorof IEEE ComSoc e-News and as an Associate Editor of the IEEETransactions on Reliability from 1992-1994.

Hwajung Lee received her Doctor of Science degree in Computer Science from theGeorge Washington University in 2003. She iscurrently a professor in the School ofComputing and Information Sciences atRadford University and visiting professor at theUniversity of Virginia. Her research interestsare broadly in the areas of Computer Network,Computer Security, and Algorithm. Recently,her research has focused on Resilient Networksin Overlay Networks, Distributed Networks,and Wireless Sensor Networks.

Steven Emmerson works as a seniorsoftware developer in the Unidata Program Center of the University Corporation forAtmospheric Research in Boulder, Colorado, which he joined in 1989. Before that, Stevenwas a senior research associate in the Satellite Remote Sensing Group of the RosenstielSchool of Marine and Atmospheric Science ofthe University of Miami in Florida. Stevenreceived an MSc in Physical Oceanographyin 1979 from the University of Miami and a BSc in Physics in 1974 fromthe University of California, Irvine.

Jack W. Davidson (Life Fellow,IEEE, Fellow, ACM) received his B.A.S.and M.S. degree from Southern MethodistUniversity in 1975 and 1977, respectively.He received a Ph.D. in Computer Sciencein 1981 from the University of Arizona, andhe joined the faculty at the University of Virginia in 1982. His research interestsinclude compilers, computer security,programming languages, computerarchitecture, and embedded systems. Heserved as an Associate Editor of ACM's Transactions on ProgrammingLanguages and Systems from 1994–2000, and as Associate Editor of ACM'sTransactions on Architecture and Compiler Optimization from 2005–2014.Professor Davidson is co-author of two best-selling introductory programming textbooks, C++ Program Design: An Introduction to Object-Oriented Programming, 3rd edition and Java 5.0 Program Design: AnIntroduction to Object-Oriented Programming, 2nd edition. He and hiscolleague, James P. Cohoon, received the 2008 IEEE Taylor L. BoothAward for their sustained effort to transform introductory computer scienceeducation. In 2020. he was elevated to IEEE Fellow, "For contributions tocompilers, computer security and computer science education."