

Multi-Site Multicast Experiments on Grid'5000

Corentin Detry
UCLouvain, Belgium
corentin.detry@uclouvain.be

Louis Navarre
UCLouvain, Belgium
louis.navarre@uclouvain.be

Olivier Bonaventure
UCLouvain & WELRI, Belgium
olivier.bonaventure@uclouvain.be

Abstract

Interest in *IP Multicast* is growing again as a way to reduce duplicate traffic, with recent proposals such as BIER, AMT, and multicast extensions for QUIC. Evaluating multicast protocols and applications requires many receivers distributed in realistic topologies. Existing techniques are rarely suited to multicast evaluations, e.g., emulators may not support multicast protocols, and large testbeds such as Grid'5000 do not support inter-site multicast routing. We present GEM, a framework that builds multi-site multicast topologies on Grid'5000. GEM deploys a multicast router running PIM and OSPF in each cluster, and connects them with GRE tunnels. GEM multiplexes receivers as network namespaces on bare-metal Linux servers. We demonstrate GEM with a Flexicast QUIC source serving 300 receivers across four sites, and observe run-to-run variance of up to 7.3 s in transfer duration for 10 MB files, a group-wide effect that evaluations with few homogeneous receivers may not reveal.

CCS Concepts

• **Networks** → **Network experimentation.**

Keywords

Multicast, Network Experimentation, Grid'5000, QUIC

ACM Reference Format:

Corentin Detry, Louis Navarre, and Olivier Bonaventure. 2026. Multi-Site Multicast Experiments on Grid'5000. In *The 22nd International Conference on emerging Networking EXperiments and Technologies (CoNEXT Companion '26)*, December 07–10, 2026, Utrecht, Netherlands. ACM, New York, NY, USA, 3 pages. <https://doi.org/10.1145/3844665.3859467>

1 Introduction

One-to-many communications, where multiple clients request the same content at (nearly) the same time, cause traffic surges in Internet Service Provider (ISP) networks. ISPs have reported traffic surges caused by live streaming [13]

or software releases [14]. Even Content Delivery Networks (CDNs) report large amounts of duplicated traffic; e.g., Akamai announced that between 20 % and 50 % of its traffic is duplicate [11]. This is mainly due to the way data is delivered using *unicast* transport protocols like TCP or QUIC.

IP Multicast is an efficient alternative that relies on packet replication on routers. Although IP Multicast has historically suffered from several issues that prevented large-scale deployment on the Internet [7], researchers are reconsidering its use because unicast-only protocols are becoming increasingly costly [1]. For example, new approaches to multicast in QUIC [9, 15] have emerged, showing that applications can rely on multicast for efficiency and unicast otherwise. Flexicast QUIC [15], a proposed Multipath QUIC extension, uses a shared unidirectional path called a multicast flow to transmit data efficiently with multicast, and per-receiver unicast paths for acknowledgments and retransmissions. The IETF standardized AMT [4] and BIER [18], and is now discussing multicast for large-scale live streaming [13].

An important step toward (re-)adopting and (re-)deploying IP Multicast protocols is *testing* and *evaluating* their performance at scale. Evaluations of reliable multicast protocols, such as Flexicast QUIC [15] or NORM [2], cannot be extrapolated from evaluations with few homogeneous receivers, as the protocols' behavior depends on the whole group; e.g., a client behind a slow or lossy link affects the transmission rate for all receivers [15]. Network simulations scale to high receiver counts at the cost of long run times, and they model link delay, loss, and host behavior rather than measure them. Emulation avoids this, but requires multiple servers to scale to large receiver counts. Medium- to large-scale emulation requires a distributed infrastructure that allows researchers to dynamically vary the number of receivers and topology to match the protocol and use cases. Such infrastructure rarely routes multicast traffic between servers, requiring emulators to use tunneling methods [6]. For example, Grid'5000 [3] does not route multicast between sites. These limits make multicast evaluation at scale difficult.

In this paper, we propose GEM (Grid'5000 Experimental Multicast framework), a framework that builds medium- to large-scale topologies across multiple clusters. We rely on Grid'5000 [3], a large-scale and flexible testbed for experiment-driven research that is free for academic researchers from partner institutions. Grid'5000 spans 12 sites across France and neighboring countries, interconnected with 10 Gbps links.



This work is licensed under a Creative Commons Attribution 4.0 International License.

CoNEXT Companion '26, Utrecht, Netherlands

© 2026 Copyright held by the owner/author(s).

ACM ISBN 979-8-4007-3006-1/2026/12

<https://doi.org/10.1145/3844665.3859467>

Existing network emulators and testbeds typically rely on network and process virtualization to allow customization and isolation. Distributed emulators such as MaxiNet [17], Kollaps [10] scale to thousands of virtual nodes but abstract network conditions by shaping links. GEM instead runs real multicast routing processes over real network links, allowing delay and loss to come from the network itself rather than a model. Flexibility is not lost, however, as traffic shaping with `tc` remains available. GEM is open source¹. The results of the demonstration (Section 3) are available on GitHub². This work raises no ethical issues. AI tools were used for language editing, but the content was human-made. Source code is human-made, with LLM-assisted bug fixes.

2 Implementation

GEM is a Python library consisting of ~1500 LoC, built on EnOSlib [5], a framework that simplifies experiments on platforms such as Grid'5000. EnOSlib enables researchers to define experiments in Jupyter Notebooks with little testbed-specific code. Networking experiments, including multicast, on Grid'5000 are straightforward as long as they remain within a single cluster or site. Beyond that, researchers have to write significant boilerplate code. For example, multicast routing is not configured between sites, requiring encapsulation to be handled manually. GEM handles this configuration, letting the researcher focus on experiment-specific code. GEM orchestrates experiments spanning multiple clusters with many receivers. It was evaluated with up to 1000 receivers on 40 physical servers.

Grid'5000 integration. GEM relies on Grid'5000 for its scale and large number of servers. Each site hosts one or more clusters, varying in server type and quantity, for a total of 939 nodes and 31 272 CPU cores across all clusters. Multicast routing is not functional between sites. To bridge this gap, GEM uses one server per cluster to be a *multicast router*, whose role is to replicate multicast packets for downstream routers and receivers (Figure 1) and to forward unicast packets. Routers have no processes running besides FRRouting (v10.7). To ensure path parity, unicast and multicast traffic is forwarded by the routers. Multicast forwarding throughput reaches ~4.7 Gbps without loss (with `iperf2`).

GEM connects routers with GRE tunnels. Protocol Independent Multicast (PIM) [8] in Source-Specific Multicast (SSM) mode runs over the tunnels, forming a distribution tree rooted at the source. Replication happens at the cluster level, with each inter-site link carrying a single copy of each packet. GRE tunnels are used instead of Automatic Multicast Tunneling (AMT) [4] because they can carry PIM messages between routers to form multicast distribution trees, whereas AMT relies on unicast replication to the receivers.

¹<https://github.com/zeykafx/gem>

²https://github.com/zeykafx/fcquic_applications_master_thesis

Topology configuration. Experiments are configured with a YAML file containing the experiment settings, cluster information, and topology. Clusters hosting the source or receivers are assigned a multicast router. The topology is defined by specifying the links between routers. From this, GEM handles server and network configuration.

Cluster networks. Grid'5000 servers have one production IP address, preassigned and routable across Grid'5000. Subnets can be reserved, allowing users to assign addresses themselves but providing no additional isolation. GEM reserves an IPv4 /22 (or /16) subnet per cluster to assign addresses to the namespaces. Receivers use their cluster's router as their default gateway, sending all of their traffic to it through the cluster's switch. Routers advertise their cluster's subnet prefix over the GRE tunnels with OSPF.

Multiplexing clients on servers. Larger-scale experiments may require hundreds to thousands of receivers, but Grid'5000 enforces per-user CPU-hour limits, so each server must host multiple receivers. Deploying several virtual machines per host is an option, but virtualization adds overhead that affects measurements [16], and virtual NICs typically filter multicast unless configured otherwise [12]. GEM instead multiplexes several network namespaces on bare-metal Linux servers, each attached to the production interface through MACVLAN. We used 15 namespaces per server so that each gets at least one dedicated core; this number is adjustable according to the experiment's needs.

3 Demonstration

We demonstrate GEM with an experiment that aims to showcase that real network conditions with heterogeneous receivers are beneficial for evaluating reliable multicast protocols. A Flexicast QUIC (FCQUIC) source in *Lille* sends a file of varying size to 300 receivers in four clusters: *Nancy*, *Rennes*, *Nantes*, and *Toulouse*. Each cluster has 5 servers with 15 network namespaces, totaling 75 receivers per cluster. File sizes vary in { 10 KB, 100 KB, 1 MB, 10 MB }, to cover sizes ranging from those requiring a single flight of packets to transfers that require congestion window growth. Each experiment is run 10 times and we extract transfer duration, which is the time between the first and last stream frames of the transfer from receiver qlogs. FCQUIC uses Cubic CCA.

Figure 1 shows the topology used. The source cluster in *Lille* has a GRE tunnel with *Nancy*, which has 2 other tunnels: one to *Toulouse* and one to *Rennes*, which is itself connected to *Nantes*, forming a multicast distribution tree, with replication happening in *Nancy* for packets to *Rennes* and *Toulouse*.

Transfer duration variations. Within a run, all 300 receivers complete within milliseconds of one another (median spread of 4.26 ms at 10 MB), consistent with a multicast flow adapting its rate for the receiver group. Across runs, however, the median transfer duration varies by 42 ms at 100 KB, 83 ms

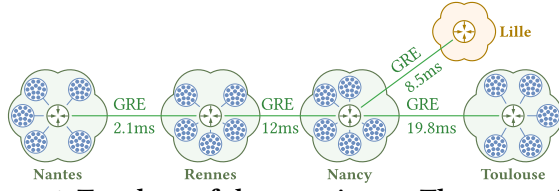


Figure 1: Topology of the experiment. The source is located in Lille. Each receiver cluster contains a multicast router (center circle) and 5 servers (circles) running 15 namespaces each (dots). Links are GRE tunnels labeled with the inter-site RTT.

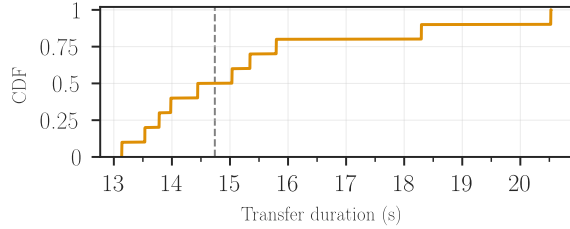


Figure 2: CDF of transfer duration for 10 MB files and all runs. Each step contains one or more test runs, gaps highlight run-to-run spread. Receivers in different clusters have a few ms difference in transfer duration. Run-to-run transfer duration varies from ~13.2 s to ~20.5 s. Median transfer duration shown as dotted line.

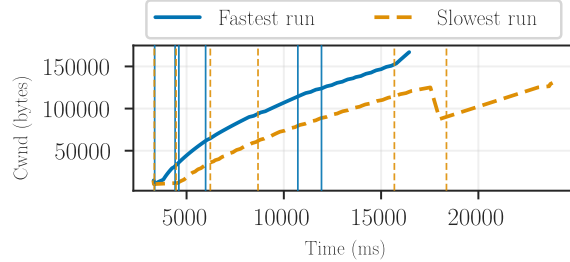


Figure 3: Multicast flow congestion window growth (10 MB files) for selected runs, starting at the beginning of file transfer. Fastest transfer duration run in solid blue, slowest one in dashed orange. Vertical lines indicate retransmissions on unicast paths.

at 1 MB, and 7.3 s at 10 MB (Figure 2). At 10 MB, the run-to-run spread comes from slower congestion window growth on the multicast flow in some runs, coinciding with retransmissions on the unicast paths (Figure 3). The FCQUIC source adapts to the slowest receiver; a retransmission for one receiver can slow down the group. As a result, experiments with many receivers on a real topology expose group-wide protocol effects that a homogeneous setup may hide.

4 Conclusion and Future Work

This paper presented GEM, a framework for medium- to large-scale multicast experiments on Grid'5000 capable of orchestrating receivers over a multicast distribution tree

formed by connecting clusters with GRE tunnels. We demonstrated GEM with an experiment using FCQUIC to highlight the ability to perform multicast experiments in Grid'5000. Future work on GEM includes support for other tunneling protocols, dynamic failures, and per-link or per-receiver traffic shaping from the configuration file.

Acknowledgments

This work has been partially supported by the Walloon Region (Belgium) as part of the funding of the FRFS-WEL-T strategic axis. Experiments presented in this paper were carried out using Grid'5000, one of the precursor infrastructures of the SLICES-FR testbed, operated by a scientific interest group including 14 major French academic institutes and Universities (see <https://slices-fr.eu/>).

References

- [1] Sepehr Abdous, Jinqi Lu, et al. 2025. One to Many: Closing the Bandwidth Gap in AI Datacenters with Scalable Multicast. In *Proc. HotNets '25*. ACM. <https://doi.org/10.1145/3772356.3772425>
- [2] B. Adamson, C. Bormann, et al. 2009. NACK-Oriented Reliable Multicast (NORM) Transport Protocol. RFC 5740.
- [3] Daniel Balouek, Alexandra Carpen Amarie, et al. 2013. Adding Virtualization Capabilities to the Grid'5000 Testbed. In *CCIS*.
- [4] Gregory Bumgardner. 2015. *Automatic Multicast Tunneling*. Request for Comments RFC 7450. IETF. doi:10.17487/RFC7450
- [5] Ronan-Alexandre Cherrueau, Marie Delavergne, et al. 2022. EnosLib: A Library for Experiment-Driven Research in Distributed Computing. *IEEE TPDS* 33, 6 (June 2022), 1464–1477.
- [6] Giuseppe Di Lena, Andrea Tomassilli, et al. 2021. DistriNet: A Mininet Implementation for the Cloud. *SIGCOMM CCR* (2021).
- [7] Christophe Diot, Brian Neil Levine, et al. 2000. Deployment issues for the IP multicast service and architecture. *IEEE network* 14, 1 (2000).
- [8] B. Fenner, M. Handley, et al. 2016. Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised). RFC 7761.
- [9] Max Franke, Jake Holland, and Stefan Schmid. 2024. MCQUIC: A multicast extension for QUIC. In *Proc. IEEE NCA*. IEEE, 185–192.
- [10] Paulo Gouveia, João Neves, Carlos Segarra, et al. 2020. Kollaps: decentralized and dynamic topology emulation. In *Proc. EuroSys*. 1–16.
- [11] Jake Holland. 2020. IP Multicast: Next steps to make it real. (2020). NANOG79, available from <https://youtu.be/2aihLUb1elg?list=PLO8DR5ZGla8i-aVXtTFRZ6l7BRBvYdrkO>.
- [12] libvirt. 2026. Domain XML format: network interfaces. <https://libvirt.org/formatdomain.html#network-interfaces>
- [13] McGill. 2025. Using Multicast for Large-Scale Live Streaming over the Internet. (2025). <https://datatracker.ietf.org/meeting/124/materials/slides-124-mops-side-mtg-using-multicast-for-large-scale-live-streaming-over-the-internet-01>
- [14] Flavio Luciani (Namex). 2025. Yesterday Fortnite launched a massive update that significantly impacted the network, visible also through IXPs. (2025). <https://www.linkedin.com/feed/update/urn:li:activity:7258748607485386753/>
- [15] Louis Navarre, Quentin De Coninck, et al. 2025. Taking the best of multicast and unicast with flexicast QUIC. *SIGCOMM CCR* 55, 2 (2025).
- [16] Guohui Wang and Eugene Ng. 2010. The impact of virtualization on network performance of amazon EC2 data center. In *Proc. INFOCOM*.
- [17] Philip Wette et al. 2014. Maxinet: Distributed emulation of software-defined networks. In *Proc. IFIP Networking*. IEEE, 1–9.
- [18] J. Wijnands, E. Rosen, et al. 2017. Multicast Using Bit Index Explicit Replication (BIER). RFC 8279. doi:10.17487/RFC8279