It Is Time to Reconsider Multicast

Louis Navarre
UCLouvain
Louvain-la-Neuve, Belgium
louis.navarre@uclouvain.be

François Michel
UCLouvain
Louvain-la-Neuve, Belgium
francois.michel@uclouvain.be

Olivier Bonaventure
UCLouvain
Louvain-la-Neuve, Belgium
olivier.bonaventure@uclouvain.be

***Abstract*—Multicast enables efficient point-to-multi-points communications. However, due to several deployment issues, multicast research slowed in the early 2000s and many of its use-cases were replaced by Content Delivery Networks and unicast communications. We argue that despite its past deployment complexities, multicast should be reconsidered to build a more energy-efficient Internet. We highlight using measurements in emulated networks the benefits of multicast regarding CPU cycles and traffic volume. Moreover, we discuss how the past limitations could be solved with today's Internet architecture and protocols, such as the Bit Index Explicit Replication mechanism.**

***Index Terms*—Multicast, energy-efficient communication, P2MP, Bit Index Explicit Replication, BIER**

I. Introduction

Multicast enables efficient resource utilization when considering point-to-multi-points (P2MP) communications. Packets are not duplicated for the segments that are identical to multiple receivers. It ensures that at most a single copy of each packet traverses each link of the network, thus constructing a multicast tree from the source towards all receivers. Multicast routing protocols (e.g., PIM [29]) usually leverage the underlying unicast routing to distribute the packets to the receivers.

This mechanism is efficient regarding resource-consumption because (i) the multicast source sends a single copy of each packet independently of the number of receivers and (ii) the number of packet copies in the network is minimal with respect to the underlying routing protocol.

Multicast was a hot topic in the late 90s. IP Multicast [6] was the first deployed large-scale multicast network, but mainly used for research purposes. Despite its attractive properties, the research in multicast slowed due to several deployment issues [8]. Most use-cases were replaced with easier solutions, such as Content Delivery Networks (CDNs). CDN servers are generally located close to the receivers and transmit their data using unicast communications.

Some applications still use multicast communications, notably IPTV or financial applications. However, point-to-multipoint applications such as video-conferencing [15] and live-streaming [23], which both gained in popularity in the past few years, still rely on unicast delivery while they could clearly benefit from a multicast architecture. The Bit Index Explicit Replication (BIER) protocol [26] was recently introduced to reduce the deployment burden of multicast networks, notably by removing the need of explicitly constructing and maintaining multicast routing trees.

This position paper reconsiders the use of multicast in existing networks. Beside the more complex architecture compared to unicast-only communications, we believe that its resource-efficient properties should encourage the network community to reconsider multicast and propose deployable solutions.

This paper is organized as follows. In Section II, we emphasize on the impacts of multicast compared to unicast in a P2MP scenario. We demonstrate again that multicast decreases the number of CPU cycles on the multicast sender, as well as decreases the utilization in the network while keeping the same workload. Section III presents some historical issues of IP Multicast, and how they could be resolved today with new protocols that do not require per flow-state in the routers such as BIER [26], and user-space implementations such as the QUIC transport protocol [12]. Section IV concludes this paper.

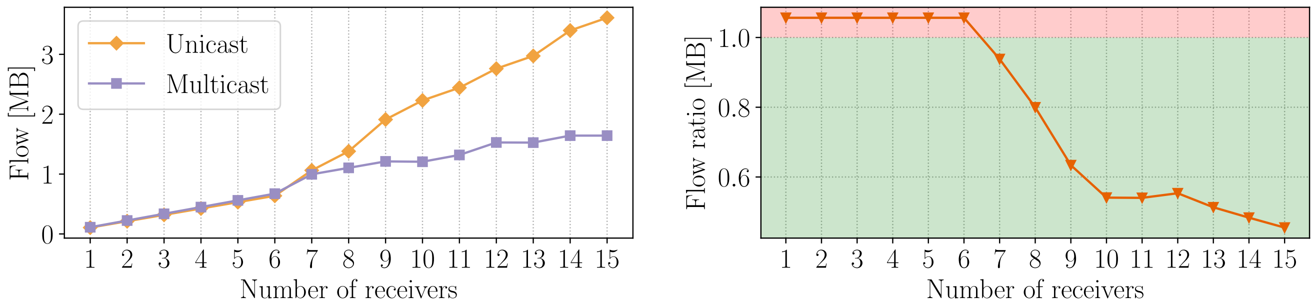
II. Benefits of multicast

In a multicast-capable network, packets flow from the source to the destinations with the help of intermediate routers composing the multicast tree. This multicast tree is usually built following the underlying routing protocol [29]. Packets are only copied by a router when receivers are reachable through different next hops.

Many use-cases on the Internet currently rely on unicast routing to send the same information to several recipients. For instance video-conferencing solutions often rely on relay servers duplicating the video content to every participant of a meeting. This may add pressure on several network nodes that are far away from the data recipients. These data could be sent in a single network packet and duplicated by the servers close to the recipients, saving important network resources. The CDN providers may play a key role in this process.

We highlight the benefits of transitioning from unicast to multicast for such use-cases. We emulate the GEANT network using Mininet [10]. GEANT is composed of 22 routers and 36 bidirectional links.

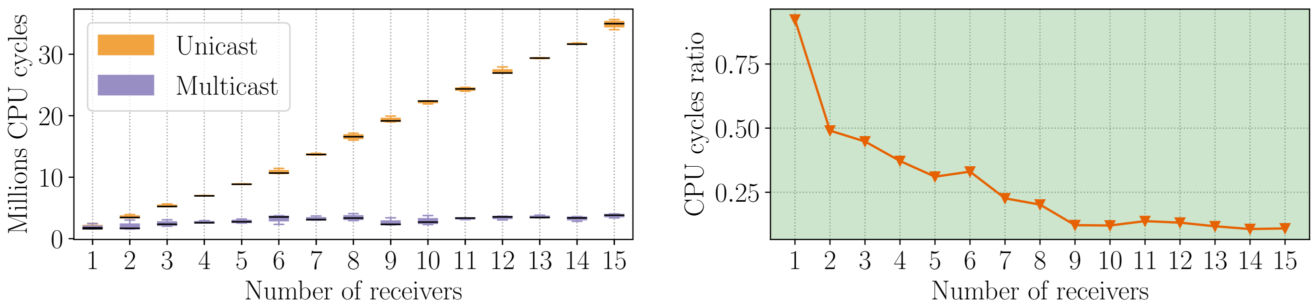
We use Bit Index Explicit Replication (BIER) [26] as our multicast forwarding mechanism. BIER is a recently standardized stateless multicast architecture. In a nutshell, the multicast tree is implicitly embedded in each packet as a *bitstring*, where each bit uniquely identifies a multicast BIER-capable router in the network. The BIER Ingress Forwarding Router (BFIR) is the only node keeping state for the receivers' location of the multicast flow. BIER leverages the underlying routing protocol to forward and duplicate the packets in the



(a) Raw values.

(b) Ratio multicast/unicast.

Fig. 1. Number of MB flowing in the network from the source to the destinations.



(a) Raw values.

(b) Ratio multicast/unicast.

Fig. 2. CPU cycles count at the sender.

network. Our experimental setup leverages our open-source implementation of BIER that exposes a socket-like API to send BIER packets [18].

An experiment consists in a source sending 100 UDP packets containing 1000 bytes of payload to an increasing number of receivers in the network. In the unicast case, the source sends a copy of each packet to each receiver individually. In the multicast scenario, the source sends a single copy of each packet and relies on the multicast forwarding mechanism. We willingly set aside the communication initialization overhead, i.e., how the receivers notify their intention to receive the UDP traffic.

For an increasing number of receivers n, we measure (i) the total number of bytes flowing in the network (Fig. 1) and (ii) the number of CPU cycles executed by the source (Fig. 2). In the unicast model, this is the sum of the CPU cycles of each process sending traffic to the receivers.

Fig. 1a presents the evolution of the number of bytes flowing in the network with respect to the number of receivers. This value increases both with unicast and multicast as we increase the number of receivers, but multicast scales better with wider groups. When the number of receivers is small ($n \leq 6$), the evolution is identical for both solutions. This is confirmed by Fig. 1b. We noticed that the first 6 receivers of our experiments are directly connected to the source. All packet copies are thus performed by the BIFR, making no difference between unicast and multicast.

Fig. 1b also highlights that for $n \leq 6$, multicast generates more bytes on the wire compared to unicast. First, BIER adds an overhead due to its additional header [27]. Second, our implementation [18] forwards the BIER packets inside IP tunnels [28], increasing this overhead.

With more receivers ($n \geq 7$), we see that multicast outperforms unicast in terms of resource-utilization. **With an identical workload, fewer bytes flow in the network with multicast.**

Fig. 2 shows the number of CPU cycles executed by the source. We run the experiment 3 times for each value of n. Fig. 2b shows the ratio between the two median values. Fig. 2a shows that for a unique receiver ($n = 1$), multicast and unicast behave similarly. The apparent small speed-up of multicast is certainly due to artifacts. The number of CPU cycles remains constant for multicast, independently of the increasing number of receivers. On the other hand, the CPU cycles of the unicast sender linearly increases with n. Fig. 2b shows that when $n = 2$, the multicast sender already uses half the number of CPU cycles compared to the unicast version. This ratio further decreases with n. **With multicast, the number of CPU cycles executed by the source remains constant, independently of the number of receivers.**

One could argue that Fig. 2 should take in consideration that the overhead of duplicating the packets is spread among the multicast-capable routers instead of the source. However, as fewer packets are forwarded in the network (see Fig. 1), we assume that the reduction of the number of packets to process compensates this overhead. Hardware implementations of BIER (e.g., with P4 [16]) would further reduce the cost of packet copies on routers.

This section supposed that the source sends UDP packets with an unprotected payload. If we consider secured communication with protocols such as QUIC [12], TLS over TCP and TCPLS [22], the overhead on the sender further increases compared to a unicast solution. For each receiver, the sender shall separately encrypt the payload at great cost. With a unified multicast flow, the sender would only encrypt the data once. This design rises several security concerns, such as source authentication and dynamicity in multicast groups, which are still open research problems. Multicast extensions for QUIC [11] suggest solutions to these issues.

III. Past issues with multicast

Despite its attractive properties for P2MP communication, several deployment issues contributed to the decline in interest in multicast [8], [25]. In this section, we discuss some of these aspects, and how they could be solved using new standardized protocols such as BIER and the existing Internet architecture.

a) Multicast does not scale in large networks: An historical issue of IP Multicast concerns the state required on intermediate routers, which grows with the number of multicast groups in a network. Routers must maintain state for each multicast tree that is explicitly built. In the late 90s, these routers did not have much memory to support numerous groups. Moreover, the signaling used to build and maintain these trees adds a processing overhead.

With new multicast architectures such as BIER [26], the routers only need to maintain state proportional to the size of the network, independently of the number of multicast groups. The multicast tree is implicitly built inside the *bitstring* of the BIER header [27]. Moreover, such source-routed solutions enable for traffic-engineering purposes, e.g., with BIER-TE [9] and Yeti [7]. Other stateless approaches suggest the use of IPv6 Segment routing for multicast [4]. With the large addressing space of IPv6, it is also possible to embed the *bitstring* directly in the IPv6 destination address [20]. **With stateless mechanisms such as BIER, it becomes easier to deploy multicast in large-scale networks.**

b) New multicast protocols are not easily deployed: In the late 90s, deploying new protocols was a tedious task as they were mainly implemented directly in the kernel. It required complex programming and updates of the network stacks. Nowadays, the performance gap between kernel-space and user-space implementations has decreased. It becomes easier to design, implement and deploy new protocols. The fast adoption of QUIC [12] is an example. The P4 programming language [2] enables to easily implement various data-plane functions at high-speed. Multicast forwarding mechanisms could benefit from this modularity. **It becomes possible to implement multicast protocols in user-space.**

c) Inter-domain multicast is a hard mission: Multicast forwarding protocols have initially been designed to work in intra-domain networks. With the advent of Wide Area Networks, it became mandatory to think about inter-domain multicast, which needs consensus between different entities. Some efforts suggested generic solutions [1], [21] but never got widely accepted in real networks.

Inter-domain multicast is still an open research problem, and wide deployment of multicast will require solutions. **Overlay networks may create an intra-domain overlay for multicast forwarding.** Past research already explored this solution [5], [13]. Nowadays, Content Delivery Networks are usually located near their customers [24]. These CDNs could act as multicast relays to their receivers, and benefit from intra-domain multicast in the networks that they serve.

Other research problems remain open and should be addressed when considering multicast communications at large scale: (i) multicast applications using encrypted payload [8], [17], [25] require renewal of encryption keys whenever a receiver joins/leaves the group; (ii) reliable multicast transport protocols (such as RMTP [19]) must avoid the ACK-implosion problem [14]; (iii) resource authentication requires techniques ensuring that the receivers read data sent by the correct source.

IV. Discussion

This paper proposes to reconsider the use of multicast protocols for point-to-multi-points communications.

Multicast adds complexity in the network and in the transport protocols compared to a unicast model [3], [8]. However, with an identical workload, multicast reduces the resource-consumption as less duplicated packets flow in the network. This, in turn, can lead to a reduction in energy consumption.

Additionally, we showed that some past issues of multicast could be solved with recent protocols such as BIER and the existing Internet architecture.

The last decades focused on designing easy-to-deploy and easy-to-maintain solutions, at the cost of an overuse of Internet resources. A naive solution has historically been to increase the network capacity and the servers' computing power. It led to an ever-increasing energy consumption, which was not a concern at that time.

We argue that it may be the time to reconsider this tradeoff and focus more on the energy-efficiency of our protocols and communications. Multicast can be a step towards this direction.

Acknowledgements

Louis Navarre is an F.R.S.-FNRS Research Fellow. We would like to thank Maxime Piraux and Nicolas Rybowski for their ideas and comments that helped shape this paper.

REFERENCES

[1] Kevin C Almeroth. The evolution of multicast: From the mbone to interdomain multicast to internet2 deployment. *IEEE Network*, 14(1):10–20, 2000.

[2] Pat Bosshart, Dan Daly, Glen Gibb, Martin Izzard, Nick McKeown, Jennifer Rexford, Cole Schlesinger, Dan Talayco, Amin Vahdat, George Varghese, et al. P4: Programming protocol-independent packet processors. *ACM SIGCOMM Computer Communication Review*, 44(3):87–95, 2014.

[3] Germano Caronni, K Waldvogel, Dan Sun, and Bernhard Plattner. Efficient security for large and dynamic multicast groups. In *Proceedings Seventh IEEE International Workshop on Enabling Technologies: Infrastucture for Collaborative Enterprises (WET ICE'98)(Cat. No. 98TB100253)*, pages 376–383. IEEE, 1998.

[4] Weiqiang Cheng, Gyan Mishra, Zhenbin Li, Aijun Wang, Zhuangzhuang Qin, and Chi Fan. Design Consideration of IPv6 Multicast Source Routing (MSR6). Internet-Draft draft-cheng-msr6-design-consideration-00, Internet Engineering Task Force, July 2022. Work in Progress.

[5] Yi Cui, Baochun Li, and Klara Nahrstedt. ostream: asynchronous streaming multicast in application-layer overlay networks. *IEEE Journal on selected areas in communications*, 22(1):91–106, 2004.

[6] Dr. Steve E. Deering. Host extensions for IP multicasting. RFC 1112, August 1989.

[7] Khaled Diab and Mohamed Hefeeda. Yeti: Stateless and generalized multicast forwarding. In *19th USENIX Symposium on Networked Systems Design and Implementation (NSDI 22)*, pages 1093–1114, 2022.

[8] Christophe Diot, Brian Neil Levine, Bryan Lyles, Hassan Kassem, and Doug Balensiefen. Deployment issues for the ip multicast service and architecture. *IEEE network*, 14(1):78–88, 2000.

[9] Toerless Eckert, Michael Menth, and Gregory Cauchie. Tree Engineering for Bit Index Explicit Replication (BIER-TE). RFC 9262, October 2022.

[10] Nikhil Handigol, Brandon Heller, Vimalkumar Jeyakumar, Bob Lantz, and Nick McKeown. Reproducible network experiments using container-based emulation. In *Proceedings of the 8th international conference on Emerging networking experiments and technologies*, pages 253–264, 2012.

[11] Jake Holland, Lucas Pardue, and Max Franke. Multicast Extension for QUIC. Internet-Draft draft-jholland-quic-multicast-02, Internet Engineering Task Force, July 2022. Work in Progress.

[12] Jana Iyengar and Martin Thomson. QUIC: A UDP-Based Multiplexed and Secure Transport. RFC 9000, May 2021.

[13] John Jannotti, David K Gifford, Kirk L Johnson, M Frans Kaashoek, and James W O'Toole Jr. Overcast: Reliable multicasting with an overlay network. In *Fourth Symposium on Operating Systems Design and Implementation (OSDI 2000)*, 2000.

[14] Brian Neil Levine and Jose J Garcia-Luna-Aceves. A comparison of reliable multicast protocols. *Multimedia systems*, 6(5):334–348, 1998.

[15] Kyle MacMillan, Tarun Mangla, James Saxon, and Nick Feamster. Measuring the performance and network utilization of popular video conferencing applications. In *Proceedings of the 21st ACM Internet Measurement Conference*, pages 229–244, 2021.

[16] Daniel Merling, Steffen Lindner, and Michael Menth. Hardware-based evaluation of scalable and resilient multicast with bier in p4. *IEEE Access*, 9:34500–34514, 2021.

[17] Matthew J Moyer, Josyula R Rao, and Pankaj Rohatgi. A survey of security issues in multicast communications. *IEEE network*, 13(6):12–23, 1999.

[18] Louis Navarre, Nicolas Rybowski, and Olivier Bonaventure. Experimenting with bit index explicit replication. In *CoNEXT-SW '22*, Rome, Italy, 2022.

[19] Sanjoy Paul, Krishan K. Sabnani, JC-H Lin, and Supratik Bhattacharyya. Reliable multicast transport protocol (rmtp). *IEEE journal on selected areas in communications*, 15(3):407–421, 1997.

[20] Maxime Piraux, Tom Barbette, Nicolas Rybowski, Louis Navarre, Thomas Alfroy, Cristel Pelsser, François Michel, and Olivier Bonaventure. The multiple roles that ipv6 addresses can play in today's internet. *ACM SIGCOMM Computer Communication Review*, 52(3):10–18, 2022.

[21] Maria Ramalho. Intra-and inter-domain multicast routing protocols: A survey and taxonomy. *IEEE Communications Surveys & Tutorials*, 3(1):2–25, 2000.

[22] Florentin Rochet, Emery Assogba, Maxime Piraux, Korian Edeline, Benoit Donnet, and Olivier Bonaventure. Tcpls: modern transport services with tcp and tls. In *Proceedings of the 17th International Conference on emerging Networking EXperiments and Technologies*, pages 45–59, 2021.

[23] Kunwadee Sripanidkulchai, Bruce Maggs, and Hui Zhang. An analysis of live streaming workloads on the internet. In *Proceedings of the 4th ACM SIGCOMM conference on Internet measurement*, pages 41–54, 2004.

[24] Sipat Triukose, Zhihua Wen, and Michael Rabinovich. Measuring a commercial content delivery network. In *Proceedings of the 20th international conference on World wide web*, pages 467–476, 2011.

[25] Debby Wallner, Eric Harder, and Ryan Agee. Key management for multicast: Issues and architectures. Technical report, 1999.

[26] IJsbrand Wijnands, Eric C. Rosen, Andrew Dolganow, Tony Przygienda, and Sam Aldrin. Multicast Using Bit Index Explicit Replication (BIER). RFC 8279, November 2017.

[27] IJsbrand Wijnands, Eric C. Rosen, Andrew Dolganow, Jeff Tantsura, Sam Aldrin, and Israel Meilik. Encapsulation for Bit Index Explicit Replication (BIER) in MPLS and Non-MPLS Networks. RFC 8296, January 2018.

[28] Zheng Zhang, Zhaohui (Jeffrey) Zhang, IJsbrand Wijnands, Mankamana Prasad Mishra, Hooman Bidgoli, and Gyan Mishra. Supporting BIER in IPv6 Networks (BIERin6). Internet-Draft draft-ietf-bier-bierin6-05, Internet Engineering Task Force, September 2022. Work in Progress.

[29] Bill Fenner (), Mark J. Handley, Hugh Holbrook, Isidor Kouvelas, Rishabh Parekh, Zhaohui (Jeffrey) Zhang, and Lianshu Zheng. Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised). RFC 7761, March 2016.