Intradomain Multicast Routing for Modern Routers

Anthony Doeraene
anthony.doeraene@uclouvain.be
UCLouvain
Louvain-la-neuve, Brabant Wallon, Belgium

Olivier Bonaventure
olivier.bonaventure@uclouvain.be
UCLouvain & WELRI
Louvain-la-neuve, Brabant Wallon, Belgium

ABSTRACT

IP multicast was designed in an era of software-based routers, when forwarding state lived in expandable RAM. Today, multicast relies on hardware for efficient replication of packets across interfaces, which has hard, fixed limits on the number of multicast forwarding entries it may hold. We argue that the limited available multicast state calls for a redefinition of the multicast protocol stack.

In this paper, we survey multicast forwarding-state capacity across ISP core routers and enterprise routers to highlight their hardware limits. We then propose to extend existing routing protocols (OSPF and IS-IS) to expose the hardware replication capacities of nodes to the control plane. Building on these exposed capacities, we propose a capacity-aware path-selection algorithm for PIM-SSM based on Shortest Widest Path (SWP), allowing to load-balance multicast joins over paths with remaining multicast hardware entries. Our results show that, by exposing multicast capacities, we can not only improve the join acceptance rate by up to 22%, but our solution also allows to reach a two times higher join rate without rejecting any join. By increasing the join acceptance rate, more concurrent multicast flows can coexist in networks.

CCS CONCEPTS

• **Networks** → **Routing protocols**; **Network resources allocation**.

KEYWORDS

Multicast, PIM, Link-State

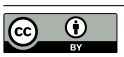
ACM Reference Format:
Anthony Doeraene and Olivier Bonaventure. 2026. Intradomain Multicast Routing for Modern Routers. In *ACM/IRTF Applied Networking Research Workshop (ANRW '26), July 20, 2026, Vienna, Austria.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3822163.3827931

1 INTRODUCTION

IP multicast is a point-to-multipoint technology allowing a source to efficiently send packets to a set of receivers, ensuring that each packet traverses each link of the network at most once. Traditional multicast deployments rely on IGMP/MLD [4, 10] and Protocol Independent Multicast (PIM) [13] to learn about receivers and create distribution trees to route packets in the network. With Source Specific Multicast (SSM), receivers can join a multicast group G from source S by sending an IGMP/MLD Join message to their router, which will use the PIM protocol to create the distribution tree. PIM-SSM [18] creates SSM source trees by sending `Join(S,G)` messages hop-by-hop toward the source using the reverse path forwarding (RPF) interface. Using this principle, intermediate routers directly learn and maintain for each (S, G) pair the list of interfaces on which packets should be replicated. In this paper, we focus exclusively on SSM[1]. One of the assumption of the original multicast design by Deering [11] is that multicast group membership is always granted, an assumption baked into every layer of the multicast protocol stack. However, as network sizes evolved over the years, reality now differs vastly from this assumption. With multicast, routers need to maintain forwarding state for each (S, G) channel. In practice, routers have limited capacities to store these multicast forwarding entries. A modern router holds between roughly 1 000 and 128 000 (S, G) entries depending on the forwarding ASIC (described in section 2). Whenever a router has exhausted its available entries, it either (a) silently rejects new joins [7, 19] and receivers **do not** receive multicast packets, or (b) falls back to full broadcast causing **traffic burst and losses** at hosts [31], both alternatives being suboptimal.

By routing all joins on the RPF path to the source, PIM may fail to create branches for certain receivers due to resource exhaustion on this path, even though alternate paths with available resource may exist. We propose to expose the spare multicast forwarding capacity to routers via existing link-state protocols (OSPF, IS-IS). By using this information, PIM can route joins on potentially non-shortest paths that have remaining capacity to offer multicast connectivity even if the RPF path towards the source has no spare capacity. To route the PIM join on non-shortest path, we propose a source-routing approach which encodes the path a join should follow directly in the join message. If the border router finds no alternate path exists, an ICMP/ICMP6 message can be sent to the host to inform it of the failure, which has already been discussed in a previous draft on Group Unreachable [17]. Correspondingly, a PIM `Join Rejected` message should also be added if a failure occurs higher in the tree towards the source due to concurrent joins exhausting the remaining capacity.

Our choice to build on OSPF and IS-IS is deliberate. When multicast routing was originally designed, the IETF made it independent of the unicast routing protocol, since standardised and proprietary unicast protocols coexisted and multicast had to work across all of them. The same reasoning drove the design of LDP [3] as a stand-alone label distribution protocol for MPLS. Segment Routing [14, 15] took a different approach: it assumed the availability of a link-state routing protocol and built upon it to support traffic



ANRW '26, July 20, 2026, Vienna, Austria

ACM ISBN 979-8-4007-2876-1/2026/07
https://doi.org/10.1145/3822163.3827931

[1]We focus on SSM as Any Source Multicast (ASM) is deprecated for interdomain use-cases [1], but the solutions we propose could also apply to the source and shared trees of ASM without any modification.

engineering, fast reroute, and services such as VPNs. This assumption proved well-founded — link-state routing (OSPF or IS-IS) is now deployed in virtually every ISP network. We take the same approach for multicast capacity: rather than designing yet another independent protocol, we extend OSPF and IS-IS to carry multicast forwarding capacity advertisements and build capacity-aware path selection on top.

In this paper, our contributions are:

(1) We survey and quantify multicast forwarding-state capacity across ISP core routers and enterprise routers (Section 2).
(2) We propose extensions to OSPF and IS-IS, allowing to carry per-node and per-interface multicast capacity advertisements (Section 3).
(3) By leveraging capacity information exposed in (2), we propose a capacity-aware RPF algorithm for PIM-SSM based on Shortest Widest Path (SWP) that routes joins on paths with maximum remaining capacity and rejects infeasible joins explicitly (Section 4).
(4) We evaluated our capacity-aware RPF algorithm by simulating multicast joins on several topologies (e.g. ISPs from Topology Zoo [21] and fat-tree). Our results show that our solution allows to consistently achieve a higher number of joins, allowing more concurrent multicast channels to coexist in the network.

We used AI in this paper to improve the writing and to generate a basic skeleton for the simulator, which we extended to meet our needs. All our artefacts (simulation and plotting scripts) are available on a dedicated repository: https://github.com/IPNetworkingLab/capacity-aware-pim.

2 MODERN MULTICAST ROUTERS

Modern routers rely on hardware assistance to replicate multicast packets. Figure 1 reports all platforms for which we found a precise hardware limit using public documentation, sorted by capacity. The range spans three orders of magnitude, from 1,000 entries on entry-level devices to 128,000 on a recent Cisco 8000. Platforms differ not only based on the number of multicast forwarding entries, but also based on their overall architectures. Two main architectures exist.

Centralized. In some platforms the multicast forwarding table is a single, chassis-wide resource shared across all line cards and interfaces. One global counter describes the exhaustion risk: when the chassis-wide table is full, *no* new (S, G) entry can be installed regardless of which interface receives the PIM Join.

Distributed. Distributed forwarding architectures instantiate a separate forwarding ASIC (NPU) per line card. A new (S, G) entry must be installed on every LC that has relevant receivers or uplinks, with each line card having its own finite TCAM. Exhaustion on one line card does not block entry installation on another.

Even for routers using the same architecture, how multicast entries are store may differ vastly. For example, let us consider the architecture of the Cisco 8000, a popular ISP router. This router uses a P100 Silicon One ASIC [9], with two types of memory: Longest Prefix Match (LPM) to store IPv4 and IPv6 routes and the Central Exact Match (CEM) memory. The latter is more versatile, as it can store `/128` IPv6 host addresses, local MPLS labels, `(S,G)` and `(*,G)` multicast states and sometimes keys for ACLs. The operator can configure the part of the CEM memory used for each purpose [9]. As shown on Figure 1, a Cisco 8000 router can support up to 128k IPv4 `(S,G)` entries or 64k IPv6 `(S,G)` entries. Older generation 88xx routers used the Q100 ASICs [8] that contain smaller CEM memory. Another example is the NCS 5500 which uses a Broadcom Jericho ASIC, supporting up to 100k IPv4 `(S,G)` entries or 60k IPv6 `(S,G)` entries by using an external TCAM [6].

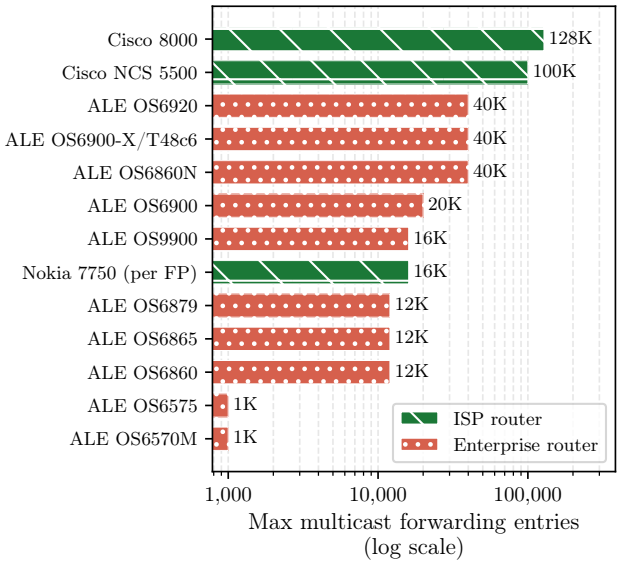


Figure 1: Maximum multicast forwarding-entry limits for platforms with publicly documented hardware bounds, sorted by capacity (log scale).

Other vendors also provide some information about the scalability and architecture of their platforms. For example, the Nokia 7750 router [26] uses one to eight Forwarding Plane (FP) ASICs per chassis, each supporting up to 16k `(S,G)` simultaneous states. Depending on the chassis configuration, such a router can support between 16k and 128k `(S,G)` states. It also supports the `pim-scaling` command which extends this value to 256k but disables some multicast features. We also obtained detailed information about a family of routers from Alcatel-Lucent [2]. The lower-end ones support up to 1k `(S,G)` entries while the larger ones support up to 40k `(S,G)` entries. In real networks, these numbers may vary based on how each operator has configured its routers, e.g. when the platform requires to reserve a fraction of the ASIC memory for unicast or MPLS routes.

To enforce those state limits, all major router vendors use *silent drop*: when the configured per-interface cap is hit, subsequent IGMP Membership Reports are simply discarded. Some routers (e.g., Cisco, Juniper) log this issue as a *group limit reached* problem, but no protocol message reaches the host. The client does not receive any data on the multicast stream, making it difficult for clients to debug this problem. To avoid this issue, operators that deploy IPTV services [16] use a limited amount of channels (about 2K `(S,G)`) for their entire IPTV service, well below the capacity of their routers. Operators offering BGP-MVPN [22, 27] services can also suffer from the limited amount of multicast entries available on routers, as it

relies on the creation of a multicast distribution tree for each high-rate flow. Several applications could also benefit from multicast, due to their inherent group communication patterns. Examples of such applications includes: live streaming platforms for streaming an identical video to all users, databases for efficient replication [30], iterative algorithms for machine learning [23] and recommendation systems [5]. Multicast state problems is one of the reason that may discourage the usage of multicast for these applications, at the cost of a worse bandwidth usage.

3 TOWARDS CAPACITY-AWARE MULTICAST ROUTING

Multicast distribution trees built by PIM-SSM treat every node as an equivalent replication engine with unlimited forwarding state. The data in Figure 1 show this assumption is false. From this observation, we argue that the control plane should be extended to advertise multicast state capacity.

3.1 OSPF and IS-IS Extensions

OSPF and IS-IS already carry traffic-engineering extensions that advertise per-link bandwidth [20, 24], enabling MPLS-TE to route around congested links. We propose an analogous *multicast state capacity* advertisement operating at two granularities, based on the router architecture.

Centralized. A new *node attribute* sub-TLV is added to the IS-IS Router Capability TLV or the OSPF Router Information LSA, carrying an additional field *available-states*, the number of available multicast entries for the router in its global TCAM.

Distributed. A new sub-TLV is appended to the existing IS-IS Extended Reachability TLV or the OSPF TE Link LSA, carrying the same field containing the number of available multicast entries in the TCAM of this interface.

With this information flooded network-wide, PIM-SSM's RPF computation and (S, G) tree-building can prefer paths through nodes with ample remaining capacity and steer new groups away from nodes near exhaustion. An ingress router can perform per-group admission checks by verifying that every node on the candidate tree branch has at least one free entry before committing the join.

3.2 Implications for Protocol Design

Advertisement frequency. Advertising exact utilisation at every (S, G) installation or deletion would generate excessive control-plane chatter. We adopt the threshold-triggered flooding model used for MPLS-TE bandwidth advertisements [20, 24, 28]: a router originates a new LSA/LSP only when its multicast state utilisation crosses one of a fixed set of thresholds. To prevent oscillation when utilisation fluctuates around a boundary, each level uses *hysteresis*: a separate *up* threshold (fired when utilisation rises above it) and a slightly lower *down* threshold (fired when utilisation falls below it), such that a new LSA/LSP is originated only when the crossing occurs in the direction that changes the advertised level. We propose coarse bands at low utilisation (25%,50%,75%), slightly finer bands at high utilisation (80%,85%,90%) and near exhaustion (>90%), we propose 1-percentage-point bands. This approach ensures that capacity advertisements are more frequent when a node has few remaining entries, where a single additional (S, G) can make the difference between a feasible and infeasible path. When many entries are available, advertisements are less frequent, reducing the flooding overhead. A thorough evaluation of the impact of threshold values on the number of advertisement is left as future work.

Backward compatibility. For routers having legacy PIM-SSM implementations will simply ignore the new TLVs, which is the default in OSPF/IS-IS. As consequence, these routers treat nodes as having an infinite capacity, preserving today's behaviour. For routers having a capacity-aware implementation, they can conservatively assign a small, locally configurable default budget to nodes that do not advertise their capacity. Another possibility would be to try to avoid nodes missing capacity information and prioritize sending joins on paths where nodes expose their multicast capacity.

4 CAPACITY-AWARE PIM-SSM PATH SELECTION

Standard PIM-SSM selects the upstream neighbour for a `Join(S,G)` from the unicast routing table, choosing the shortest path toward the source S with no knowledge of multicast forwarding state. We propose to replace this unicast next-hop lookup with a *capacity-aware path selection* derived from the capacity information flooded by OSPF/IS-IS (Section 3). Unlike plain RPF, this selection must account for intermediate routers that may lack available forwarding entries: a path that is topologically valid under unicast routing may be infeasible for multicast if any router along it has exhausted its number of available multicast forwarding entries. Note the path selection algorithms we present are pessimistic. They consider the worst case where every hop on the path will install one entry for the (S, G) group. In practice certain nodes may already have an entry for this channel, such that no new entry is installed. In consequence, no path may be found by these algorithms even though a feasible path exists due to existing entries. Finding every feasible path would require to expose all multicast forwarding entries, which is completely intractable.

Available multicast forwarding entries. Each router v_1 advertises $a(v_1, v_2)$ for each interface connected to neighbour v_2, its *number of available multicast forwarding entries*: the count of free rows remaining in its TCAM for this interface that can accept a new (S, G) state. We model routers with a single global TCAM with e available entries by assigning this same number of entries to all interfaces, i.e. for all interface i of v_1, $a(v_1, v_i) = e$. The quantity $a(v_1, v_2)$ directly determines the feasibility of a path used to forward joins.

Formal path metric. Let $G = (V, E)$ be the network graph annotated with the capacity $a(v_1, v_2)$ and IGP cost $c(v_1, v_2) > 0$ on each link (v_1, v_2). For a path $p = (v_1, \ldots, v_k)$, we define:

$$A(p) = \min_{1 \le i < k} a(v_i, v_{i+1}) \qquad \text{(bottleneck available entries)}$$

$$C(p) = \sum_{i=1}^{k-1} c(v_{i+1}, v_i) \qquad \text{(IGP path cost)}$$

We consider only the capacity from v_i to v_{i+1}, as multicast entries are installed for replication only in the TCAM of the RPF interface towards the source. For the IGP cost, we use the cost of the reversed edges (v_{i+1} to v_i). Paths that joins will follow are computed from the

receiver to the source, thus we need to use the cost of the reversed edges to compute the best paths from the source to the receiver.

For the path selection, we use the **Shortest Widest Path (SWP)** preference relation:

$$p_1 \prec p_2 \text{ iff } \begin{cases} A(p_1) > A(p_2), & \text{or} \\ A(p_1) = A(p_2) \text{ and } C(p_1) < C(p_2). \end{cases}$$

SWP selects the path that maximises the bottleneck available-entry count, breaking ties in favour of the shortest IGP path. SWP is an interesting algorithm, because it is guaranteed to produce loop-free paths due to its monotonicity property [29] as long as IGP weights are strictly positive. Additionally, SWP can easily be computed by running twice a Dijkstra's algorithm [25], thus running in $O((|V| + |E|) \log |V|)$.

An alternative to SWP would be to prune all nodes with $a(v) = 0$ and run standard Dijkstra. However, this solution treats a node with one entry remaining identically to one with a thousand, making the former a likely failure point after the next join.

Rejection when no feasible path exists. If no path to S can be found, the router *rejects the join immediately*, does not forward any `Join` upstream, and signals failure back downstream. The host is notified that the group is currently unreachable via multicast and may retry, select an alternative source or fall back to unicast.

Stale-state fallback. If a `Join(S,G)` reaches a router whose table has filled up since its last LSA/LSP update, that router discards the join. It then notifies the rejection downstream with a new PIM message (`Join Rejected`). The downstream router directly connected to the host announce the failure to establish a multicast branch using ICMP/ICMP6 [17]. The host falls back to unicast, and can retry to join the multicast tree after some time has passed, hoping that some multicast entries have been freed.

5 SIMULATION

To validate the SWP heuristic, we simulate join admission on three real ISP topologies drawn from the Internet Topology Zoo [21]: Cogentco (197 nodes, 243 edges), Tata NLD (145 nodes, 186 edges), and GÉANT 2012 (40 nodes, 61 edges), as well as a fat-tree topology (180 nodes, 1728 edges) as multicast can be used in datacenter for group communication (e.g. machine learning [23]). Each router is assigned a maximum multicast forwarding entry count drawn from a log-uniform distribution over $[50, 5000]$, reflecting the order-of-magnitude spread seen in Figure 1. For the fat-tree, all nodes have a fixed capacity of 100 to reflect the homogeneity of devices in datacenters. We scale down by a factor 20 the available capacities to have tractable simulations, thus the load should be multiplied by 20 in real network to obtain realistic results. We assume that multicast groups are independent of each others, such that multicast joins arrive as a Poisson process (rate λ). Receivers listen to the joined channel for an exponentially distributed time with mean $1/\mu = 10$ s; The average network load is $\rho = \lambda/\mu$. Each join selects randomly a receiver from the set of all nodes and a channel from a pool of ρ channels. Each experiment is carried out for $5/\mu$ s, to ensure that joins and leaves are interleaved.

Capacity information is disseminated via threshold triggered LSA updates at the thresholds defined in Section 3; We consider a 10ms delay on links, such that the propagation of LSAs is not instantaneous. Results are averaged over ten independent seeds; error bars show 95% confidence intervals.

We compare the three path-selection criteria we presented in section 4: **SWP**, which takes paths with maximum remaining capacity first; **Pruned SP**, which removes fully exhausted nodes ($a(v) = 0$) before running Dijkstra, but treats all remaining nodes identically; and **SP**, plain shortest path, which rejects joins only when the chosen path passes through an exhausted node (current behaviour of PIM).

Impact on the acceptance rate. Figure 2 shows the acceptance rate across all four topologies. At low load ($\rho = 100$) all algorithms accept every join because the network is far from saturation. In this condition, utilisation remains below the first flooding threshold and SWP falls back to plain shortest-path behaviour. As load increases, plain SP degrades fastest: it commits to short paths regardless of intermediate capacity, so bottleneck nodes become exhausted early and subsequent joins fail even when alternative routes exist. Pruned SP improves on SP by sidestepping fully exhausted nodes, but it still ignores the residual available capacity of non-exhausted nodes and tends to quickly fill up the state of nodes on the shortest-path. SWP retains the highest acceptance rate on all topologies: on Cogentco at $\rho = 500$ it accepts 99% of joins vs. 98.3% for Pruned SP and 91% for SP; at $\rho = 1000$ the gap widens to 83.7% (SWP) vs. 71.9% (SP), a **+11 percentage-point improvement**. Results on Tata NLD (+17 pp), GÉANT 2012 (+22 pp) at the same load are consistent. The fat-tree shows the largest gap between SWP and SP at high loads because it offers many alternate paths, making SWP's capacity-aware detours correspondingly more valuable. Compared to plain SP, SWP allows to maintain an acceptance rate of 100% for a 2x higher load. At very high load ($\rho \geq 10^4$), the three methods tend to achieve the same acceptance rate, as the network is completely overloaded and no path has any remaining capacity to install new entries. SWP thus works best at moderate loads, where it can find alternate paths with available capacity.

Impact on flooding. In parallel to the join acceptance rate, we wanted to study the impact of our solution on the flooding process. Indeed, to maintain up-to-date capacity information, nodes need to flood the capacity information when utilisation threshold are reached. Figure 3 shows the impact of our proposition on flooding for the Cogentco and fat-tree topologies. Results for the other ISP topologies are similar to Cogentco. Both Pruned SP and SWP leads to additional flooding. At moderate loads, the number of capacity advertisement is lower when using SWP compared to Pruned SP on all studied topologies. Indeed, SWP tends to spread joins over several paths depending on the available capacity, while Pruned SP will fill up the shortest-path first until the capacity of nodes is exhausted. As capacity threshold are tighter at high utilisation, flooding is directly impacted, explaining the higher number of advertisements when using Pruned SP. However, at high loads, this trend is reversed. As SWP spread joins to maintain the maximum bottleneck capacity, it may send joins on longer paths. As paths are longer, more multicast entries are consumed per path, which results in a higher number of advertisements due to increased utilisation.

Impact on path length. Finally, as SWP may reroute joins on non-shortest path, we wanted to see the impact of this property on path length. Figure 4 presents the average path length with reference to the load. At low load all three algorithms produce

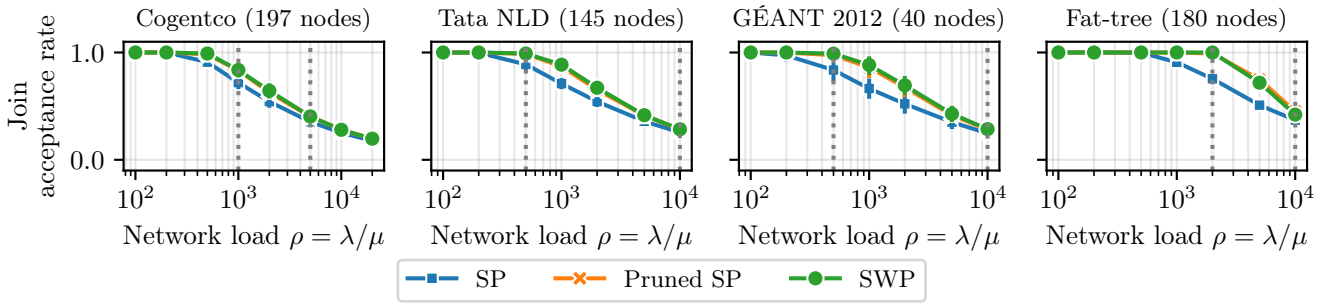


Figure 2: Join acceptance rate vs. network load $\rho = \lambda/\mu$ on selected topologies. Both SWP and Pruned SP consistently accepts the most joins at moderate loads, and have the same acceptance rate as SP at high loads. Dotted lines delimit loads under which SWP outperforms classical SP.

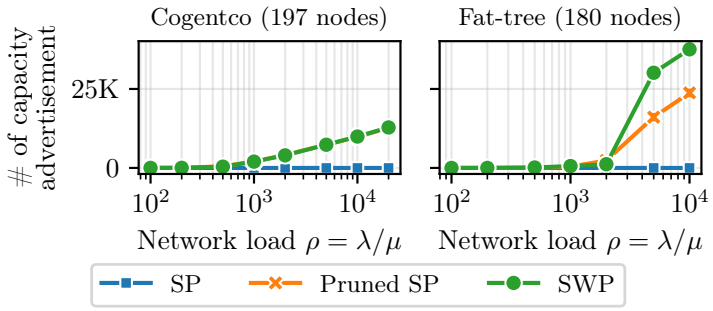


Figure 3: Number of capacity advertisement vs. network load $\rho = \lambda/\mu$ on selected topologies. Pruned SP and SWP generates flooding overhead compared to classical SP in order to distribute capacity information.

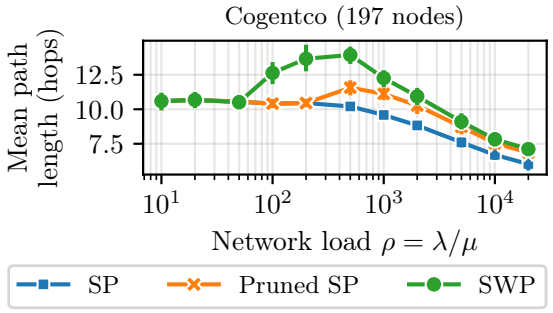


Figure 4: Mean accepted path length vs. network load on Cogentco. All three algorithms follow the same shortest-path baseline at low load. As load increases, SWP takes longer routes to avoid capacity-constrained nodes and SP path length decrease as it rejects more often longer paths.

the same mean path length: no capacity constraints have been advertised, so SWP has no reason to deviate from the shortest path. As ρ grows and nodes begin crossing advertisement thresholds, SWP starts routing around them, and its mean path length increases compared to SP and Pruned SP. At $\rho = 1000$ the mean SWP path on Cogentco is $\sim$ 12.3 hops vs. $\sim$ 11.1 for Pruned SP and $\sim$ 9.6 for SP. The detours are the direct consequence of the widest-path primary criterion: SWP routes around nodes nearing exhaustion, consuming extra hops to preserve the widest bottleneck. We consider this trade-off as appropriate: in a hardware-constrained multicast network, admitting a join on a slightly longer path is strictly preferable to rejecting it and delivering nothing. A counter-intuitive observation is that the mean path length for all three methods is decreasing as the load increases. At high load, the distribution of the acceptance rate depending on the path length is not uniform. Longer paths have a higher chance of being rejected, as they go through more nodes that may have exhausted their available multicast entries (e.g. 53% acceptance rate for a path with 5 hops vs. 28% for a path with 10 hops). All methods thus exhibits an unfair behaviour at high load and, as short paths are more often accepted than longer paths, the mean path length decreases accordingly.

6 RELATED WORK

To our knowledge, the multicast forwarding capacity of routers has not been taken into account in the design of multicast routing protocols within the IETF. Certain protocols, such as the RSVP protocol [33] takes certain resources into account. However, RSVP mainly deals with delay/bandwidth and does not consider the multicast forwarding state. Its deployment for IP multicast was also limited.

BIER [32] takes a different approach from PIM by encoding the distribution tree directly in packet headers, so all intermediate routers need **no** per-(S, G) state at all. The tree is represented as a *bitstring* identifying all egress border routers that should receive the packet. However, BIER trades core state for ingress border router state. Even though core routers do not need to store any state, each ingress border router must still store, for every (S, G), the associated *bitstring*, whose size grows linearly with the number of border routers. Additionally, BIER add overhead in the data plane, as it requires to store the bitstring inside packets. Similarly to PIM, we could extend BIER ingress border routers to send an ICMP/ICMP6 Group Unreachable[17] if the number of BIER fowarding entries has been exhausted.

Yeti [12] is another stateless multicast approach, which encodes distribution trees using labels. Yeti proposes an interesting extension to multicast by allowing service chaining, where multicast

packets are forced to pass through specific network functions. However, Yeti has not been standardised, and is thus inappropriate for deployment on the Internet or in public datacenters.

BGP-MVPN [27] is an extension of BGP allowing customers connected by a provider to benefit from multicast connectivity between their different sites. Providers ensure multicast connectivity between the sites of a customer by instantiating Multicast Distribution Tree (MDT), allowing an efficient distribution by replicating packets only when needed. For each customer, providers typically have one MDT (called Default MDT) inside their network for control and low-rate multicast flows, and one MDT (called Data MDT) for each high-rate multicast flow [22]. A Data MDT is created for a flow whenever this flow's rate exceeds a configurable threshold. To create the Data MDTs, provider networks typically use PIM-SSM [22]. Our solution could thus be used as-is in BGP-MVPN deployments relying on PIM for creating MDTs, allowing to increase the number of concurrent multicast flows coexisting in provider networks.

7 CONCLUSION

Multicast allows efficient distribution of the same information to many receivers. It provides a much better utilization of the network resources than unicast for one-to-many communications. However, it was designed for a world where group membership always succeeded, replication state was unlimited, and the only relevant limit was link bandwidth. These assumptions have been overtaken by hardware reality with limited resources.

In this paper, we first surveyed the hardware limits of several ISP and enterprise routers and showed that the multicast forwarding capacity of modern routers is finite and needs to be managed like other limited resources. We then proposed to revisit intradomain multicast routing protocols to take these limitations into account. Since OSPF and IS-IS are the dominant intradomain routing protocols, we propose to extend them to carry multicast state capacity advertisements, exposing the available replication capacity to the control plane. Building on this capacity information, we introduced two capacity-aware path-selection algorithms for PIM-SSM, Pruned SP and SWP, allowing to load balance multicast joins on alternate paths to leverage spare multicast capacity.

We validated our design with simulation on representative Internet Topology Zoo networks and a fat-tree topology. Pruned SP and SWP accepts 11–22 percentage points more joins at $\rho = 1000$ (20k concurrent (S, G) in real networks) than plain shortest-path routing. For 100% acceptance rate, both methods allows to achieve a 2x higher load compared to the maximum load plain shortest-path routing can support.

Our future work will include deployment trials to evaluate SWP in operational networks, as well as extensions of the path selection algorithm to include other interesting metrics, such as available link bandwidth or link delay.

8 ACKNOWLEDGEMENTS

This publication is supported by the Walloon Region as part of the FNRS WEL-T strategic axis.

REFERENCES

[1] M. Abrahamsson, T. Chown, L. Giuliano, and T. Eckert. 2020. Deprecating Any-Source Multicast (ASM) for Interdomain Multicast. RFC 8815 (Best Current Practice). https://doi.org/10.17487/RFC8815

[2] Alcatel-Lucent Enterprise 2025. *OmniSwitch AOS Release 8 Specifications Guide.* Alcatel-Lucent Enterprise. Part No. 060972-00, Rev. A.

[3] L. Andersson (Ed.), I. Minei (Ed.), and B. Thomas (Ed.). 2007. LDP Specification. RFC 5036 (Draft Standard). https://doi.org/10.17487/RFC5036 Updated by RFCs 6720, 6790, 7358, 7552.

[4] B. Cain, S. Deering, I. Kouvelas, B. Fenner, and A. Thyagarajan. 2002. Internet Group Management Protocol, Version 3. RFC 3376 (Proposed Standard). https://doi.org/10.17487/RFC3376 Obsoleted by RFC 9776, updated by RFC 4604.

[5] Mosharaf Chowdhury, Matei Zaharia, Justin Ma, Michael I. Jordan, and Ion Stoica. 2011. Managing data transfers in computer clusters with orchestra. *SIGCOMM Comput. Commun. Rev.* 41, 4 (Aug. 2011), 98–109. https://doi.org/10.1145/2043164.2018448

[6] Cisco. 2024. Multicast Configuration Guide for Cisco NCS 5500 Series Routers, IOS XR Release 24.1.x, 24.2.x, 24.3.x, 24.4.x. (Dec 2024). https://www.cisco.com/c/en/us/td/docs/iosxr/ncs5500/multicast/24xx/configuration/guide/b-multicast-cg-ncs55k-24xx/implementing-multicast.html.

[7] Cisco. 2026. Multicast Configuration Guide – Mroute Limit and IGMP Limit. (2026). https://www.cisco.com/c/en/us/td/docs/switches/lan/c9000/multicast/multicast-configuration-guide/mroute-limit-and-igmp-limit.html.

[8] F. Cuiller. 2023. Cisco 8000 FIB Scale. (March 2023). https://xrdocs.io/8000/tutorials/cisco-8000-fib-scale.

[9] F. Cuiller. 2025. Cisco 8000 FIB Scale for Silicon One P100. (May 2025). https://xrdocs.io/8000/tutorials/cisco-8000-fib-scale-for-silicon-one-p100.

[10] S. Deering, W. Fenner, and B. Haberman. 1999. Multicast Listener Discovery (MLD) for IPv6. RFC 2710 (Proposed Standard). https://doi.org/10.17487/RFC2710 Updated by RFCs 3590, 3810, 9777.

[11] S. E. Deering. 1988. Multicast routing in internetworks and extended LANs. *SIGCOMM Comput. Commun. Rev.* 18, 4 (Aug. 1988), 55–64. https://doi.org/10.1145/52325.52331

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

[13] B. Fenner, M. Handley, H. Holbrook, I. Kouvelas, R. Parekh, Z. Zhang, and L. Zheng. 2016. Protocol Independent Multicast - Sparse Mode (PIM-SM): Protocol Specification (Revised). RFC 7761 (Internet Standard). https://doi.org/10.17487/RFC7761 Updated by RFCs 8736, 9436.

[14] Clarence Filsfils, Nagendra Kumar Nainar, Carlos Pignataro, Juan Camilo Cardona, and Pierre Francois. 2015. The Segment Routing Architecture. In *2015 IEEE Global Communications Conference (GLOBECOM).* 1–6. https://doi.org/10.1109/GLOCOM.2015.7417124

[15] C. Filsfils (Ed.), S. Previdi (Ed.), L. Ginsberg, B. Decraene, S. Litkowski, and R. Shakir. 2018. Segment Routing Architecture. RFC 8402 (Proposed Standard). https://doi.org/10.17487/RFC8402 Updated by RFC 9256.

[16] Swati Gupta. 2020. IPTV Statistics 2026 and Beyond: Market Size, Growth, and Key Trends. (April 2020). https://www.apprupt.com/iptv-statistics/.

[17] Mickael Hoerdt, Benoit Hilt, and Jean-Jacques Pansiot. 2006. Simple Join Failure Notification for PIM-SM Multicast Routing (draft-hoerdt-pim-group-unreachable-00). https://datatracker.ietf.org/doc/html/draft-hoerdt-pim-group-unreachable-00.

[18] H. Holbrook and B. Cain. 2006. Source-Specific Multicast for IP. RFC 4607 (Proposed Standard). https://doi.org/10.17487/RFC4607

[19] Juniper. 2026. Junos CLI Reference — group-limit (IGMP). (2026). https://www.juniper.net/documentation/us/en/software/junos/cli-reference/topics/ref/statement/group-limit-edit-protocols-igmp-interface.html.

[20] D. Katz, K. Kompella, and D. Yeung. 2003. Traffic Engineering (TE) Extensions to OSPF Version 2. RFC 3630 (Proposed Standard). https://doi.org/10.17487/RFC3630 Updated by RFCs 4203, 5786.

[21] Simon Knight, Hung X. Nguyen, Nickolas Falkner, Rhys Bowden, and Matthew Roughan. 2011. The Internet Topology Zoo. *IEEE Journal on Selected Areas in Communications* 29, 9 (2011), 1765–1775. https://doi.org/10.1109/JSAC.2011.111002

[22] Gkavogiannis L. 2026. Multicast Services over SRv6 / IPv6 Cores. (June 2026). https://xrdocs.io/multicast/blogs/multicast-services-over-srv6-ipv6-cores.

[23] Mu Li, David G. Andersen, Jun Woo Park, Alexander J. Smola, Amr Ahmed, Vanja Josifovski, James Long, Eugene J. Shekita, and Bor-Yiing Su. 2014. Scaling distributed machine learning with the parameter server. In *Proceedings of the 11th USENIX Conference on Operating Systems Design and Implementation* (Broomfield, CO) *(OSDI'14).* USENIX Association, USA, 583–598.

[24] T. Li and H. Smit. 2008. IS-IS Extensions for Traffic Engineering. RFC 5305 (Proposed Standard). https://doi.org/10.17487/RFC5305 Updated by RFCs 5307, 8918.

[25] Qingming Ma and Peter Steenkiste. 1997. On path selection for traffic with bandwidth guarantees. In *Proceedings 1997 International Conference on Network*

Protocols. IEEE, 191–202.

[26] Nokia. 2026. SR OS 26.3.R1: Log events guide: Maximum number of multicast (S,G) records reached on IOM. (2026). https://documentation.nokia.com/sr/26-3/7x50-shared/log-events/log-pim.html#ariaid-title20.

[27] E. Rosen (Ed.) and R. Aggarwal (Ed.). 2012. Multicast in MPLS/BGP IP VPNs. RFC 6513 (Proposed Standard). https://doi.org/10.17487/RFC6513 Updated by RFCs 7582, 7900, 7988.

[28] Stefano Salsano, Alessio Botta, Paola Iovanna, Marco Intermite, and Andrea Polidoro. 2006. Traffic engineering with OSPF-TE and RSVP-TE: Flooding reduction techniques and evaluation of processing cost. *Comput. Commun.* 29, 11 (July 2006), 2034–2045. https://doi.org/10.1016/j.comcom.2005.12.007

[29] João Luís Sobrinho. 2002. Algebra and Algorithms for QoS Path Computation and Hop-by-Hop Routing in the Internet. *IEEE/ACM Transactions on Networking* 10, 4 (2002), 541–550. https://doi.org/10.1109/TNET.2002.801397

[30] S. Stallion. 2010. Using Reliable Multicast for Data Distribution with OpenDDS. (February 2010). https://objectcomputing.com/resources/publications/mnb/2010/02/15/using-reliable-multicast-data-distribution-opendds.

[31] Ymir Vigfusson, Hussam Abu-Libdeh, Mahesh Balakrishnan, Ken Birman, and Yoav Tock. 2008. Dr. Multicast: Rx for data center communication scalability. In *Proceedings of the 2nd Workshop on Large-Scale Distributed Systems and Middleware* (Yorktown Heights, New York, USA) *(LADIS '08)*. Association for Computing Machinery, New York, NY, USA, Article 10, 12 pages. https://doi.org/10.1145/1529974.1529988

[32] IJ. Wijnands (Ed.), E. Rosen (Ed.), A. Dolganow, T. Przygienda, and S. Aldrin. 2017. Multicast Using Bit Index Explicit Replication (BIER). RFC 8279 (Proposed Standard). https://doi.org/10.17487/RFC8279

[33] L. Zhang, S. Deering, D. Estrin, S. Shenker, and D. Zappala. 1993. RSVP: a new resource ReSerVation Protocol. *IEEE Network* 7, 5 (1993), 8–18. https://doi.org/10.1109/65.238150