Lcast: Software-defined Inter-Domain Multicast

Florin Coras*, Jordi Domingo-Pascual*, Fabio Maino† Dino Farinacci† and Albert Cabellos-Aparicio*
*Universitat Politécnica de Catalunya (BarcelonaTECH)
†Cisco Systems

***Abstract*—This paper presents Lcast, a scalable network-layer single-source multicast framework. Lcast builds a router overlay for inter-domain content distribution and interfaces with intra-domain multicast for seamless interaction with end-hosts. The architecture is built as an extension of the Locator/ID Separation Protocol (LISP), a proposed scalable Internet architecture, and leverages its decoupling of data and control planes. In Lcast, the computation and coordination of the content distribution tree are logically centralized and managed by a central entity whilst LISP routers are responsible for the controlled replication and forwarding of data. The design, besides allowing the two planes to scale independently, also enables a software (re)configuration of the inter-domain distribution tree to match changing operational requirements. By means of large scale simulations, based on realistic network topologies, we compare several management strategies for low latency content delivery and assess their scalability and performance. Results show that the architecture can scale at least to thousands of participating *domains* and, depending on applied optimizations, can deliver traffic at latencies close to unicast ones, even in constrained replication conditions.**

I. Introduction

The Internet is gradually becoming the preferred infrastructure for delivering live content such as sports events or news to large user sets. According to recent reports, video streaming is among the largest and the fastest growing bandwidth consumers [1] and IPTV driven revenues are to rise from less less than USD 6B in 2010 to USD 17B in 2016 [2].

For such scenarios, where one-to-many content delivery is required, IP-multicast [3] is the most efficient solution in terms of bandwidth consumption. However, inter-domain multicast has failed to be widely adopted by most ISPs. The reasons often cited have to do with its management complexity and the lack of a clear commercial service [4]. While the former leads to high operational expenditure the latter leads to situations when multicast implementation over links with unicast economical agreements leads to loss of revenue.

In light of this deployment predicament, both industry and the academia have switched efforts to application-layer multicast (ALM) where innovation was not fettered. Consequently, many academic and commercial streaming products were developed and are now available. Most notably, PPLive [5] is reported to be used daily by millions of users [6].

The ever growing popularity of these commercial solutions has lead to an intense scrutiny of their performance. In some cases, the results have uncovered significant limitations of these architectures in scaling user quality of experience with the increase of client population [7]. Reasons for such behavior, among others, have to do with unavailability of inter-peer bandwidth, inefficient supply of server (source) capacity or, in some cases, insufficient client upload capacity. These limitations also have a significant impact on the business model used by commercial entities employing application-layer multicasting who are forced to base their revenues on advertisements and not on paid subscriptions. In consequence, ALM cannot be considered as the long-term solution for streaming live multimedia content. We advocate for the return to a network-layer approach as the long-term solution.

Independent of the multicast hindrance, the Internet has its own set of challenges and of late, due to unexpectedly rapid and sizeable growth, started facing increased scaling costs. In a recent workshop [8], industry and academia have agreed that the overloading of IP semantics, which identify both *location* and *identity*, is to blame for the situation. LISP [9], is one of the proposed solutions aiming to distinguish between the two roles of IP and which, at the time of this writing, is undergoing IETF standardization. Additionally, LISP benefits from the support of a considerable community whose active research efforts are devoted to its development and deployment. Apart from aiding the scalability of the Internet, the introduction of new protocol mechanisms makes LISP's possible deployment offer a rare window of opportunity for enhancing the current routing infrastructure devoid of deployment costs.

Expanding on these considerations, we present Lcast, a network-layer, LISP based, multicast framework that aims to overcome the main drawbacks of IP-multicast while mirroring the flexibility of ALM. Lcast creates an over the Internet's core LISP router overlay that it organizes in a distribution tree where each router performs the replication and forwarding of data. Group management of the routers is implemented in a *logically centralized* fashion without changing the current LISP specification. Hence, participating routers require no configuration nor need to be managed. We stress this as an important property since it circumvents one of the most important problems plaguing traditional IP-multicast. Additionally, the design facilitates and encourages software innovation [10] at control-plane level, since LISP routers remain unchanged and unaware of the group management algorithms. In fact, content providers may use different algorithms depending on their specific operational requirements.

The scalability of the architecture's data-plane is assured through design however, as for any centralized solution, that of its control-plane might be a concern. In order to assess Lcast's scalability and configurability, we evaluate its ability to deliver latency constrained content for multiple optimization scenarios. To this end, we perform large-scale simulations that

make use of real-world traces and topologies. Specifically, we generated traces using a costumer dataset ($3k$ ASes and around $140k$ unique IPs) obtained by a globally distributed capture of SopCast [11] overlays. Additionally, with datasets from RouteViews [12], CAIDA [13], RIPE [14] and iPlane [15] we generated an Internet-like topology.

The results show the control plane's ability to scale, even when active topology discovery mechanisms are used. We further observe that overall performance improves asymptotically with router replication factors and that the overlay could be optimized to deliver content at close to unicast latencies.

II. LISP Background

LISP [9] is one of the recently emerged architectural solutions to the Internet's scalability problem [8] and whose development is aided by a pilot-network [16] spanning 26 countries with members pertaining to academia and industry. Its main goal is that of splitting the semantics of IP addresses with the aim of forming two namespaces that unambiguously identify core (locators) and edge (identifiers) network objects. To facilitate transition from the current Internet infrastructure, both of the resulting namespaces use the existing IP addressing scheme. Therefore, the split does not affect routing within existing stub or transit networks. Nevertheless, as identifiers and locators bear relevance only within their respective namespaces, a form of conversion, from one to the other, has to be performed at border points between core and edge networks. LISP enabled border routers make use of a technique called map-and-encap [17] for the translation.

Apart from the need for data plane modifications, map-and-encap also requires the introduction of a new control plane *mapping function* able to provide bindings that link identifiers to locators. Therefore, prior to forwarding a host generated packet (see Fig. 1), a LISP router maps the destination address, or what LISP calls an Endpoint IDentifier (EID), to one or more corresponding destination Routing LOCator(s) (RLOC) by means of a LISP specific distributed database, called the *mapping system* [18], [19].

To retrieve a mapping, a LISP router directs a *Map-Request* message to an the mapping system and in return receives as answer a *Map-Reply*, whose content it stores in a local map-cache. Once a mapping is obtained, the border router selects a destination RLOC and tunnels the packet from edge to corresponding edge network through encapsulation with a LISP-UDP-IP header. At the receiving router, the packet is decapsulated and forwarded to its intended destination.

In LISP parlance, the source router, that performs the encapsulation, is called an Ingress Tunnel Router (ITR) whereas the one performing the decapsulation is named the Egress Tunnel Router (ETR). One that performs both functions is referred to as an xTR.

Additionally, LISP makes use of Re-encapsulating Tunnel Routers (RTRs), that perform packet re-encapsulation, to enable packet re-routing. This ability may be used to perform a form of loose source routing. Specifically, packets going from a source to a destination could follow a designated ordered

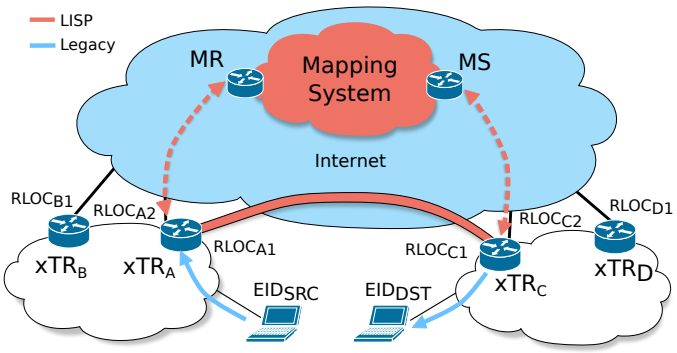


Fig. 1: LISP Architecture. xTRs obtain EID-to-RLOC bindings from the mapping-system and tunnel packets over the Internet's core.

set, or chain, of RTRs instead of the path computed by the underlying routing protocol. Although, this type of routing influences the end-to-end path followed by packets, the path between two adjacent RTRs would necessarily be the one recommended by the underlying routing protocol.

III. Proposed Architecture

Lcast is a single-source multicast framework that seeks to merge the efficiency of IP-multicast with the flexibility of ALM while aiming for a low deployment cost. To this effect, it exploits the recent development of LISP both as a means to leverage the benefits offered by the new architecture but also to speed up deployment.

As in LISP's case, Lcast requires no end-host stack changes except for intra-domain multicast support. Although apparently a stringent constraint, the requirement is generally easily satisfiable as multicast has seen much wider adoption intra-domain than inter-domain.

A. Architecture Overview

Lcast builds a router overlay that it organizes in an inter-domain distribution tree. Within it, member routers perform unicast replication of packets and interface with the end-hosts they serve through intra-domain multicast.

We presented the protocol mechanisms needed for building a LISP router overlay for inter-domain multicast in [20]. In the associated distribution tree, content replication is performed by RTRs pertaining to the source domain or a third party and packet replication is done through unicast encapsulation. Lcast makes use of the same mechanisms however, instead of relying on RTRs, it require that the ETRs of the domains with end-host interested in the streamed contend implement RTR functionality and replicate the content themselves. As a result, Lcast builds a unicast router overlay that it organizes as shown in Fig. 2.

On the data path, the source domain's border router (ITR) heads the distribution tree and is the first to perform unicast encapsulated replication of the streaming content for its children. Subsequently, all downstream overlay members (ETRs), save for the leaves, are obliged to unicast replicate the received

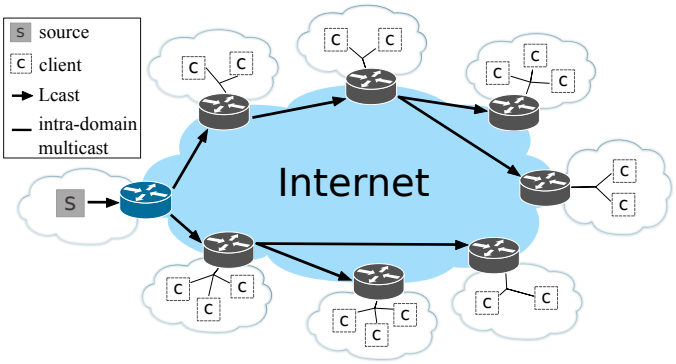


Fig. 2: Lcast data-plane architecture. The ITR is the first to replicate the content and all downstream ETRs, save for the leaves, replicate up to a fixed fan-out value.

packets. End-hosts finally receive the data from their domain border router (ETR) through intra-domain multicast.

Unicast encapsulated replication, when performed multiple times out the same interface, quickly reduces the interface's throughput with a quantity linearly proportional with the replication factor. Because of this inefficiency, in Lcast routers have a *fixed, constrained, fan-out* and, thus, force the distribution tree to become a degree-constrained tree.

Regarding the control plane, to avoid increasing the computational overhead of domain border routers, group management functions are centralized. Besides providing enhanced overlay configurability, the decision also implies no router management costs as all data paths are computed and disseminated to overlay members by a central entity. Possible implementators of the central entity are the source domain's ITR or an external, third party owned, orchestration system. For brevity, we will generally consider the these tree coordinating functions as implemented by the ITR.

B. Group Management

An ETR's *subscription* to the overlay is triggered by the receipt of an end-host subscription for multicast content. Thus, whenever a client part of a domain not yet member of an overlay sends a request for multicast content, it triggers a subscription request, in essence a Map-Request, on it's domain tunnel border router. The request propagates through the mapping system up to the ITR, which ensuing the request's receipt, starts a search for an overlay parent with spare capacity. The search may be done randomly or, if additional topological information exists, in accordance to a predefined heuristic. Once found, the ITR notifies the parent about the new child, thus ensuring the creation of a new distribution tree branch linking the two.

A member's lack of intra-domain clients prompts its *unsubscription* from the overlay. Lcast handles such departures by requiring that member routers *leave gracefully*. This offers the ITR the chance to efficiently reorganize the routers on the affected branch prior to acknowledging the member's departure. As in the case of tree optimizations, not to generate packet loss, a virtual topology is first created and only after its initialization, the departure is confirmed.

To avoid packet loss and to assure a seamless transition when performing a tree reorganization, either due to a member departure or when optimizing the distribution tree (see III-C), a *make before break* procedure is used. Hence, before breaking the data path links, the ITR configures the nodes to form a virtual topology that mimics the final one. Once the virtual topology is completed, the ITR requests the (sub)tree's parent, whose position in the overlay has not changed, to switch to the new configuration. To mark the change, and subsequently force all downstream members to use the new distribution tree, a LISP flag is set in the first data packet.

If a member loses network connectivity, its data path children will sense the failure either as a lack of data packets or by means of LISP specific mechanisms. Hence, after a relatively short time period, the ITR is informed about the occurrence and will act as in the event of a graceful leave. Still, such circumstances will result in packet loss for all members of the subtree headed by the affected node and out of band mechanisms would be required for remedying the failure. Such mechanisms are out of the scope of the current paper. However, we note that sudden loss of network connectivity for a domain's border router is a seldom occurrence.

C. Distribution Tree Software Configuration

A member's position on the overlay data-path is established at subscription time. But, if provided with an overlay topology, the ITR could be configured to further optimize the distribution tree for the minimization of a chosen metric. In such situations, the tree's efficiency is controlled by the ITR through optimal placement of joining members and/or through periodic or enforced tree reshaping.

One of the key features of Lcast is that it enables easy customization of the distribution tree as it supports the on-line replacement of the optimization algorithm ran by the ITR.

A possible tree optimizing algorithm and ways to obtaining topological information are discussed in the next section.

IV. Overlay Optimization

The previous section gave an overview of Lcast's overlay architecture and high level group management mechanisms. Nevertheless, it avoided discussing overlay optimization related mechanisms. Such concerns are orthogonal to the Lcast protocol and their choice is driven by operational concerns. However, to illustrate the framework's configurability, in this section we propose an optimization algorithm and two topology discovery mechanisms that could be used to create a distribution tree for latency constrained content. One of the topology discovery mechanisms is based on BGP and the other on latency. We shall further refer to the combination of an optimization algorithm and a topology discovery mechanism as an *optimization strategy*.

A. Distribution Tree Optimization Algorithm

In what follows, we use the term *distance* when referring to a relative length or amplitude of a metric, observed on a path connecting two points, but when the exact nature of the metric is of no interest. Considering our goal of delivering content for delay sensitive applications, the function we minimize in our experiments is the maximum distance (e.g. latency or number of AS hops) from a host to the stream's source. Notice that the reference is the end-host and not the domain border router. Thus, what matters in deciding a member's position in the overlay tree is not solely its distance to the source but also the number of clients it serves. Then, a router close to the source but serving few clients might find itself lower in the hierarchy than another with a slightly higher distance to the source but with a larger client set. In other words, the algorithm optimizes the quality of experience for end-hosts and not for routers.

The problem described above, henceforth named *minimum average distance, degree-bounded spanning tree (MADDBST)*, may be formally stated the following way:

Definition 1. *Given an undirected complete graph G=(V,E), a designated vertex $r \in V$, a degree bound $d(v) \leq d_{max}$, $\forall$ $v \in V$, $d_{max} \in \mathbb{N}$, a vertex weight function $c(v) \in \mathbb{N}$ and an edge weight function $w(e) \in \mathbb{R}^+$, $\forall$ edge $e \in E$. Let $P^T_{r,v}$ be the set of edges e on the path from vertex r to v in the graph's spanning tree T. Also, let $W^T_{r,v} = \sum_{e \in P_{r,v}} w(e)$ represent the cost of the path linking r and v in the spanning tree T. Find the spanning tree T of G, routed at r, satisfying $d_T(v) \leq d_{max}$, such that $\sum_{v \in V, v \neq r} c(v) W^T_{r,v}$ is minimized.*

We note that [21] and [22] have previously defined and solved similar optimization problems. Shi et al. [21] also proved that a particular instance of the problem, where all vertices have weight 1, is NP-complete for degree constraints $2 \leq d_{max} \leq |V| - 1$. Similarly to our approach, they were interested in a centralized solution whereas Banerjee et al. [22] have successfully manged to distribute the algorithm.

The heuristic we used to solve the MADDBST problem is similar to the one used by Banerjee and it is a variant of the one proposed by Shi. In short, the algorithm works by incrementally growing a tree started at the root node r until it becomes a spanning tree. For each node v, not yet a tree member, it selects a potential parent node u in the tree T, such that the metric $\delta(v) = (W^T_{r,u} + w(u,v))/c(v)$, or the distance to the source per client, is minimized. At each step, the node with the smallest metric value is added to the tree and the parent selection is redone.

B. BGP Topology Map

One of the best sources of topological information that is not or can not be commonly used by application layer overlays is the BGP routing table. The BGP information an AS router holds attempts to present an Internet wide interconnection map. But, due to the algorithm's distributed nature and its use of policy, both inaccuracies and incomplete data may exist.

The ITR has two options for obtaining BGP topological information. First, it may aggregate partial BGP feeds from multiple overlay members (*global view*) or second, it may itself connect to BGP (*local view*). The former could ensure a more detailed view of the topology, and thus grounds for better decisions, while the latter a more restricted, partial view of the interconnection map and seemingly worse performance. An off-line obtained BGP map may be rendered inadequate due to churn whereas one obtained through on-line aggregation of multiple BGP tables may be a technically challenging task. Moreover, some domains may be reluctant to provide such information which they deem as sensitive. By contrast, the local, on-line topology gathering mechanism requires nothing more than the BGP feeds from the routers upstream of the source domain's overlay participating border router. Additionally, there is no need for a communication protocol between the ITR and the overlay members for the conveying of BGP reachability information. We carried several experiments to assess the benefits of a global vs. a local BGP view of the topology for the overlay management. The results, although always indicating the former option as a better performer, showed small absolute differences between the two.

Due to its implementation simplicity and good performance we have opted in our experiments for the BGP topology discovery mechanism based on local information. The metric it provides, inter AS hops, in combination with the optimization algorithm results in a degree-constrained shortest AS path tree optimization strategy. For brevity, we shall refer to it as *bgp*. Should there be interest in obtaining a minimum AS hop cost tree, at the expense of larger member latencies to the source, a degree-constrained minimum spanning tree heuristic should be employed.

C. Latency Topology Map

Inter-member latency is a metric commonly employed by application layer overlays in topology optimizations. Yet, the obtaining of a full inter-member latency map scales poorly with the population size and may require unacceptably high number of measurements. Hence, a more intelligent approach for the selection of link latencies worth estimating is needed.

We obviate a large number of measurements and assure they are performed in an optimized order by exploiting a mechanism similar to the one used by Banerjee et al. in [23] for NICE's group management. The NICE control plane protocol works by clustering nodes that are close to one another in terms of latency and thus by limiting the majority of inter-member measurements to just the pairs of nodes finding themselves in close proximity. The amortized cost analysis shows that the number of control plane peers (i.e., for us the number of measured peers) at an average member is constant $O(k)$ whereas in the worst case it can reach $O(k \log(N))$. Where k is a constant limiting the node degree and the size of the cluster and N is the number of overlay members. Despite the very good scaling properties, Lcast member churn may induce slightly higher number of measured links per member despite the limited number of adjacencies.

In addition to these very good scaling properties, our implementation further reduces the member communication overhead as all the NICE group management decisions are centralized and taken by the ITR. Thus, there is no need for message exchanges between overlay members however, LISP's extension is required to provide for a simple communication mechanism between ETRs and the ITR. By default, ETRs check the liveness of the locators associated to cached mappings. Therefore, the extension requires just the implementation of a message, similar to Map-Reply, for reporting latency estimates.

The combination of the latency discovery protocol and the optimization algorithm results in an optimization strategy we further refer to as lat.

V. Evaluation Methodology

In order to evaluate the expected performance of the overlay optimization strategies proposed in Section IV we implemented an event-based simulator. The datasets used in building a network topology were obtained from Internet wide measurements carried out by iPlane [15], RouteViews [12], CAIDA [13] and RIPE [14]. The participating domains, and their respective number of clients were obtained from a worldwide distributed capture of SopCast [11] P2P Internet TV channels streaming an UEFA Champions League semifinal football match. The traces are available at [24].

In what follows we start by describing the datasets and methodology used to build a realistic Internet inter-domain topology. Next, we present the traces we generated by using the captured P2P traffic. We continue by introducing the metrics we considered for the evaluation of the simulated optimization strategies and we conclude the section with a brief presentation of our simulator.

A. Internet Inter-Domain Topology

To obtain a realistic global inter-domain topology we have aggregated datasets that estimate how autonomous systems interconnect from multiple sources: iPlane, RouteViews, CAIDA and RIPE. All the used data is from April 2011. We have knowingly dismissed link specific BGP policy information that could transform part of the AS graph's edges in arcs (directed links). However, most affected by this assumption are the links between customers and their upstream providers. Such links can not be used by the upstream providers for transiting traffic to destinations other than those found in the client's network. Yet, the use of member routers for transiting/replicating traffic is one of the main features of Lcast. We therefore think that the discounting of BGP policy information should be of limited influence to our results.

The log-log plot for the complementary cumulative distribution function (CCDF) of the AS-node degree for the resulting inter-AS topology follows a straight line (not shown here), property found in power law distributions. As previously shown in [25] and [26] the Internet AS topology is a scale-free network with power law node degree distribution. Further, the average path length in our topology is 3.5 or 5.4% lower than the 3.7 observed [27] in the Internet. These two results corroborate our claim that the aggregate topology has properties similar to those of Internet's AS graph.

For estimating inter-AS latency, we made use of iPlane's [15] proven latency prediction abilities for IP pairs [28]. Because we needed to estimate the latency between domain border routers we had to elect for all participating ASes a representant. We did so by using iPlane's estimations that associate points of presence (PoP) to ASes and their interconnection map. For any domain, the PoP with the largest degree was elected as the representant. In about 30% of the cases, when iPlane failed to provide an answer, we used an estimator based on geographical distance described in [19].

B. The Client Traces

We evaluate Lcast and the proposed management strategies by means of simulation with the help of client traces that emulate distinct types of user behavior. In our traces, the participating domains and their respective number of clients were obtained from a passive distributed capture of several P2P TV channels whereas the client churn was modeled in accordance to recent results in the field. We detail both efforts in what follows.

SopCast [11] is one of the P2P TV applications frequently used for streaming of live sports events. Wanting to model client distribution for large events of global, or at least widespread interest, we captured the traffic of several SopCast overlays during an 2011 UEFA Champions League semifinal. For the capturing process we used 2 vantage points in USA, 5 in Europe and 2 in Asia. They span a total of 6 countries. We were interested in understanding how clients cluster in autonomous systems, not in the specific performance of a channel's overlay. Thus, depending on the upload capabilities of each vantage point, we joined a number of P2P channels, streaming the same event, at each node. The traces contain more than 145k distinct IPs spread over 3.8k ASes. Our of them, we use of 3k ASes in our simulations.

Despite the size of our captured dataset, lack of logs from the overlay's bootstrapping server made it impossible to approximate client lifetime in the overlay. We thus resorted to synthetic modeling of client churn. It is generally accepted [29], [30], [6], [31], [32] that client arrival process, at least for periods spanning dozens of minutes, can be modeled by a Poisson process. Furthermore, Sripanidkulchai in [29], after analyzing 3 months worth of Akamai logs, observe that *short duration* events, which last a couple of hours, present flash crowds whereas non-stop streams have a time of day behavior. These findings were confirmed by Veloso in [31] who also noted that for *long* streams client inter-arrivals can be modeled through a Pareto or a piecewise stationary Poisson.

For client session lengths however, consensus could not be found. Thus, depending on stream length or the type of system being analyzed by either paper, they may follow different distributions. Still, with the exception of [30], there seems to be an agreement that sessions should have lengths distributed

according to a power law but opinions diverge when assessing the weight of the tail.

Considering the works discussed above, in order to perform an evaluation of our proposed architecture that acknowledges the wide range of client behaviors, we have generated 3 traces with distinct properties. The goal was to model a short event, spanning 2h 30min, with a piece-wise Poisson arrival process but with different shapes for the session length distributions. In order to capture the flash crowd effect we required that 80% of the clients join during the first 30min, and the rest spread over the time left. For the session lengths we used a Pareto distribution with a shape parameter of 1.5 and a scale parameter that took the values 1min, 15min and 1h in order to emulate low, medium and respectively high client interest in the streamed content.

The first trace represents the worst case scenario from churn perspective, as clients leave the stream soon after joining (i.e. channel surfing). Obviously, in such situation the number of active clients in the overlay is the lowest. For brevity, we shall be referring to the three traces as tli, tai and thi respectively.

C. Metrics

We evaluate the performance of the proposed schemes along the following dimensions:

1) latency stretch: this metric measures a client's relative gain in latency to the stream's source when compared to the unicast one way delay between the two.

2) hop stretch: it measures a client's relative gain in number of AS hops to the stream's source when compared to the number of hops on the unicast path linking the two.

3) tree cost: in order to quantify the efficiency of Lcast in using network resources we defined a cost that relates the number of AS hops crossed for the delivery of one piece of information to all member nodes in the overlay to those in unicast. It is computed as: $\sum_{v \in V} hop(v; p_v) / \sum_{v \in V} c(v) hop(v, root)$ Where V is the set of all members, hop is a function that returns the number of hops between two nodes, p_v is the overlay parent for v and c is a function that returns the number of clients for a member. The metric's value is lower than 1 if the overlay is more efficient than unicast delivery.

4) control traffic overhead: to evaluate the scalability of Lcast's control plane we use a set of metrics that measure the number of messages exchanged by the source with the tree members for the purpose of creating a tree, maintaining tree integrity, optimizations and topology discovery.

D. Simulator

To evaluate the overlay management algorithms presented in Section IV, we have implemented an event-based simulator that makes use of the the previously described AS topology and client traces. The effort resulted in the partial implementation of a Lcast compliant ITR.

We used as content source a well connected AS in China. For comparison, we performed experiments with other source ASes and observed similar results. Operationally, members join or leave the overlay in accordance with the client traces and member subscriptions are not optimized, hence, done at the first randomly found free position in the distribution tree. However, periodically (10 min) or if more than 10 members sustain a client increase above 10 or drop to 1, the ITR proceeds to optimizing the distribution tree. The values were chosen to balance the computation costs and the overlay performance.

VI. Results

In order to better gauge the performance of bgp and lat, we also ran simulations with a very simple overlay management strategy that performs no topology discovery or tree optimizations and which joins new members at random positions in the distribution tree. Moreover, in this scenario, we further refer to as rnd, member departures require the affected children to re-perform the join procedure.

To evaluate the overlay optimization strategies we ran simulations considering various router fan-out values and the three client traces described in Section V-B. For readability we present a comparison of average performance but for completeness and, in some situations, a better understanding we also provide the 95% confidence intervals or worst case results.

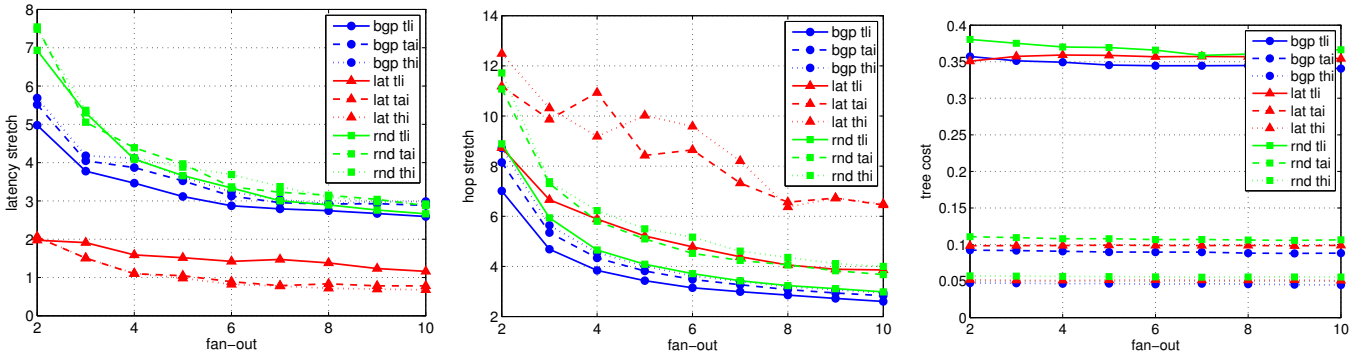
A. Latency Stretch

Figure 3a presents the average latency stretch for the tree group management algorithms. Independent of client traces, lat outperforms bgp and rnd and presents tight bounds (Figure 4a) for the metric. However, what surprises is the effectiveness of rnd relative to that of bgp. Save for the low fan-outs, the two exhibit similar performance. Overall, client churn seems to have little influence over the performance. Although all management algorithms show small advantages for tli, they are the result of the lower number of active members (i.e. a smaller graph) present in the trace and not due to optimization improvements.

The results for lat with high fan-out values (Figure 4a) exhibit latency stretch values lower than 1. They are due to a peculiarity of BGP routing called latency triangle inequalities violations [33], [34]. Their effect is that the BGP selected paths possess higher latency than others that, despite looking like detours to BGP's decision process due to increased number of hops, have low aggregated latency. As a result, it may happen that overlays might offer lower inter-member latencies when compared to their underlying, unicast, ones. Both [33], [34] have identified lower latency paths than the BGP selected ones for more than 20% of the pairs in their datasets.

B. Hop Stretch

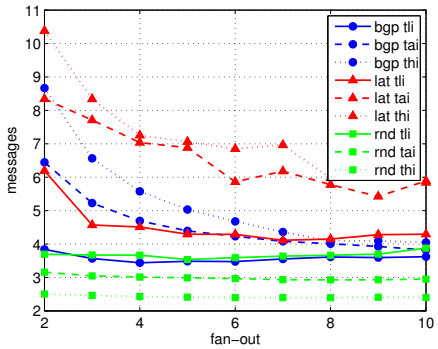
Simulation results for this metric can be seen in Figure 3b. Not surprisingly, bgp outperforms the other two algorithms but its absolute performance is strongly influenced by fan-out values. Again, rnd shows very good performance relative to that of bgp, despite the lack of tree optimizations, and actually performs better than lat. The result can be explained by the fact that rnd generally tries to build k-ary complete (i.e. low



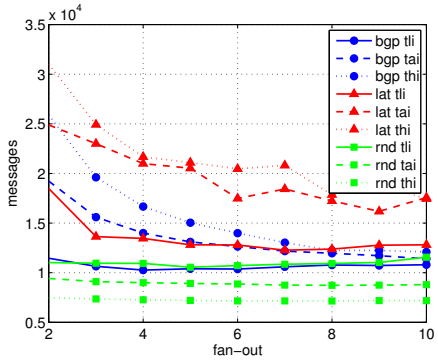
(a) Latency stretch

(b) Hop stretch

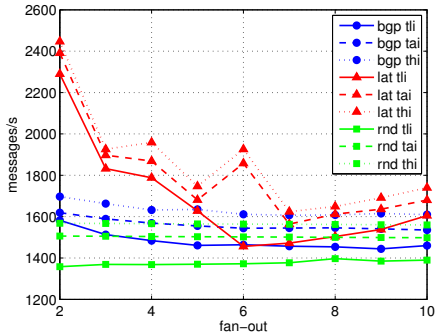
(c) Tree Cost



(d) Management messages per member



(e) Management messages per ITR



(f) Peak management messages/s for ITR

Fig. 3: Simulation results

depth) trees with short overlay paths. bgp manages to improve over rnd's result as it optimizes tree construction for low hop stretch. But, the margins are not large because the average AS path length in our topology (and the Internet) is very low, at about around 3.5 hops. In contrast, lat builds low latency path trees at the cost of higher tree depth and therefore higher overlay path lengths and hop stretch.

C. Tree Cost

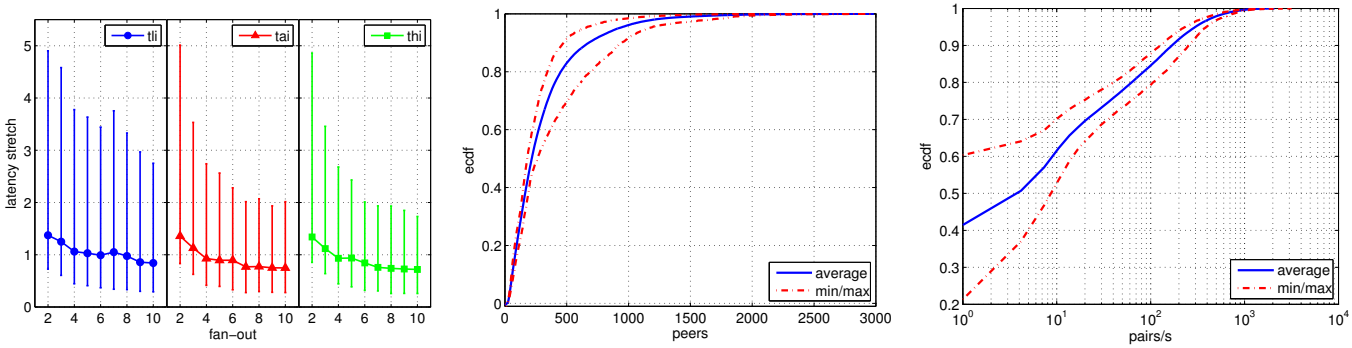
It is worth noting that MADDBST, the optimization algorithm we use, does not aim to minimize bandwidth usage but average client distance (latency or AS hops) to the source. Nevertheless, as it can be seen in Figure 3c, the architecture still manages to considerably improve tree cost, independent of the optimization strategies. Results show rnd as the the worst performer. This corroborates our assumption that rnd's previous results benefit from the low length AS paths and not from good opportunistic distribution trees. In contrast, we see that lat's high hop stretch does not result in a high tree cost because of optimized overlay path reuse. The highest tree cost registered by rnd is under 0.4 for high client churn. However, for average and low client churn Lcast is more than one order of magnitude better than unicast. Finally, results are not influenced by the fan-out, even for bgp, because the distribution trees obtained are shortest path and not minimum weight spanning trees.

D. Control Traffic Overhead

We split control traffic overhead in management overhead, needed for group management, and active topology discovery overhead needed to convey topology measurements. lat is the only one to employs active topology discovery.

Figures 3d and 3e show the average results for management overhead from client and ITR perspective respectively. Members seldom exchange messages with the ITR and even in the worst recorded case the count does not go above 33. From optimization strategy perspective, lat seems to be slightly more susceptible than bgp to disruptive tree optimizations whilst both require larger amounts of messages for lower fan-outs. rnd, due to the lack of optimizations, has very low communication overhead with no fan-out influence. For all three, higher client churn combined with large replication factors result in larger number of message exchanges.

Looking at the results for the ITR, it can be seen that it is also exposed to low average and instantaneous group management communication overhead. In fact, the highest instantaneous rate registered is $2.5k$ messages/s. In the worst recorded case it must exchange under $31k$ messages with an average rate that is lower than 11 messages/s. All these requirements are easily fulfillable and in light of the $3k$ member domains the results are a proof of Lcast's control plane scalability. High client churn slightly influences management overhead, especially for higher fan-outs, when it favors larger



(a) Latency stretch with 95% confidence intervals (b) ECDF for peers measured/member (c) ECDF for peer pairs measured/s

Fig. 4: *lat* simulation results

amounts of control traffic but with low peaks (Fig. 3f).

Active topology discovery on the other hand requires more involvement from the ITR. Figure 4c shows the ECDF for number of member pairs measured/s at the MS's request. Alas, not having implemented NICE's communication protocol, we can not provide the exact number of exchanged messages. However, if one considers that any measurement requires one message, although the ITR may batch requests, then the number of measured pairs can be used as a worst case estimate. On average, during more than 40% of the time spent in a simulation, the ITR does not request any measurements while in 79% to 89% of the time, less than 100 pairs are measured/s. In the worst situation, just 0.8% of the simulation time, around 72s, is spent performing more than $1k$ measurements/s. The largest number of measurements performed in a second is relatively easily sustained by a current day server.

Furthermore, we note that the ITR, being the measurements coordinator, is the only one affected by measurement requests peaks. Hence, in overload situations, it may delay and queue requests for later processing with limited or no effect for the quality of the distribution tree.

Overlay members participate on average in less than 220 measurements (see Figure 4b) and only 4% of the members exceed $1k$ latency estimations. Client churn and fan-out do not influence the number of measurements.

VII. Related Work

As a long standing academic and commercial research challenge, single-source live media streaming benefits from copious amounts of related literature. Consequently, in this section we restrict our focus to a limited set of solutions that bear similarities with Lcast.

Multicast functionality, that enables live media streaming, may be offered as either network or application layer service. But in light of IP-multicast's [35] deployment predicament, focus has gradually shifted from in network, router based solutions to end-host supported ones. Out of the application layer solutions proposed by academia [23], [36], [22], [37], [38], OMNI [22] is the closest in spirit to our proposal. OMNI requires service providers to deploy a set of proxying nodes that self organize in an overlay and forward traffic to subscribed clients. The optimizing algorithm employed is a distributed instance of MADDBST and the optimized metric is latency. In contrast, Lcast works at network-layer and is supposed to be deployed, at no additional cost, together with LISP. Moreover, Lcast's logically centralized control plane allows easy deployment of new optimization algorithms without requiring router changes.

Apart from the academic solutions, a large array [11], [5], [39], [40] of commercial ALM architectures are widely used for Internet content streaming. Being closed source their architectures are not completely understood, nevertheless their performance has often been the subject [6], [30], [32], [31] of academic scrutiny. The results have shown significant limitations of these architectures in scaling user quality of experience with the increase of client population. Lcast, being an extension of LISP, operates on domain border routers and thus builds an overlay topology that is not exposed to client churn. Furthermore, through design it avoids imposing bandwidth strain on overlay members and could assure certain performance bounds.

VIII. Conclusions

Our goal with Lcast was to devise an inter-domain multicast framework that, besides possessing a low deployment cost, is also easily configurable and scalable. The former requirement was fulfilled by our use in the inter-domain overlay of just LISP enabled domain border routers, without requiring any further support or changes in the Internet's core. But, equally important, by exposing the service to the clients by means of intra-domain multicast, an already implemented protocol, and by limiting the router overlay fan-out to low values.

Configurability was ensured by two design decisions: first, the separation of the control and data-planes and second, the centralization of the control-plane functions in the ITR. Member participation in the data-plane is conditioned only by the implementation of LISP functionality. However, member presence in the control plane is not required since all optimization functions are centralized in the ITR. As a result, switching between tree optimization algorithms can be done easily,

even on-line, assuring fast (re)configuration of the overlay's topology to meet operational performance requirements.

The isolation through design between local-domain and inter-domain multicast allows the separation of the overlay's router members from the churn specific to client end-hosts and thus relieves the architecture's control plane from the inherent overhead. This ensures the scaling of the architecture with the number of end-hosts however, the scaling with the number of member domains is attained through proper data and control plane design.

We evaluated three possible overlay management strategies for low latency content delivery and concluded that they are all fit for optimizing overlays with thousands of members. Several conclusions could be drawn from the analysis.To begin with, we see that AS hops are a poor estimator of latency and except for low fan-out cases, bgp and rnd behave similarly. This further leads us to conclude that rnd offers reasonable performance and therefore it can be used as an efficient, no overhead backup optimization strategy. High client churn slightly influences performance and most noticeably lightly increases management overhead. A very encouraging results is that Lcast's performance does not depend on large fan-outs and in fact, fan-outs larger that 6 offer limited benefits. Finally, we observe that lat offers very low, unicast like, latency in exchange for increased but manageable control overhead.

ACKNOWLEDGEMENTS

Many thanks to Harsha Madhyastha for the help with the iPlane latency lookup service and to Damien Saucez for his valuable comments. This work has been partially supported by the Spanish Ministry of Education under scholarship AP2009-3790, research project TEC2011-29700-C02, Catalan Government under project 2009SGR-1140 and a Cisco URP Grant.

REFERENCES

[1] C. Labovitz, S. Iekel-Johnson, D. McPherson, J. Oberheide, and F. Jahanian, "Internet inter-domain traffic," *ACM SIGCOMM Computer Communication Review*, vol. 40, no. 4, pp. 75–86, 2010.

[2] "Digital TV World Revenue Forecasts," Jun 2011. [Online]. Available: http://www.digitaltvresearch.com/products/product?id=21

[3] H. Holbrook and B. Cain, "Source-Specific Multicast for IP," RFC 4607 (Proposed Standard), Internet Engineering Task Force, Aug. 2006.

[4] C. Diot, B. N. Levine, B. Lyles, H. Kassem, and D. Balensiefen, "Deployment issues for the IP multicast service and architecture," *IEEE Network*, vol. 14, no. 1, pp. 78–88, Jan. 2000.

[5] "PPLive." [Online]. Available: http://www.pplive.com/

[6] X. Hei, C. Lian, J. Lian, Y. Liu, and K. W. Ross, "A Measurement Study of a Large-Scale P2P IPTV System," *TOM*, vol. 9, no. 8, Dec. 2007.

[7] C. Wu, B. Li, and S. Zhao, "Diagnosing network-wide p2p live streaming inefficiencies," in *INFOCOM, Proceedings IEEE*, 2009, pp. 2731–2735.

[8] D. Meyer, L. Zhang, and K. Fall, "Report from the IAB Workshop on Routing and Addressing," RFC 4984 (Informational), Internet Engineering Task Force, Sep. 2007.

[9] D. Farinacci, V. Fuller, D. Meyer, and D. Lewis, "Locator/ID Separation Protocol (LISP)," draft-ietf-lisp-23, Internet Engineering Task Force, Feb. 2012, work in progress.

[10] N. McKeown, "Software-defined Networking," Keynote Talk at IEEE INFOCOM 2009.

[11] "Sopcast P2P Internet TV." [Online]. Available: http://www.sopcast.com

[12] "Routeviews project." [Online]. Available: http://www.routeviews.org

[13] Y. Hyun, B. Huffaker, D. Andersen, E. Aben, M. Luckie, kc claffy, and C. Shannon, "The IPv4 Routed /24 AS Links Dataset - 2011-04." [Online]. Available: http://www.caida.org/data/active/ipv4_routed_topology_aslinks_dataset.xml

[14] RIPE RIS. [Online]. Available: http://ripe.net/ris

[15] H. V. Madhyastha, T. Isdal, M. Piatek, C. Dixon, T. Anderson, A. Krishnamurthy, and A. Venkataramani, "iPlane: An information plane for distributed services," in *USENIX OSDI*, Nov. 2006.

[16] "LISP Testbed." [Online]. Available: http://www.lisp4.net/

[17] R. Hinden, "New Scheme for Internet Routing and Addressing (ENCAPS) for IPNG," RFC 1955 (Informational), Internet Engineering Task Force, Jun. 1996.

[18] V. Fuller, D. Farinacci, D. Meyer, and D. Lewis, "LISP Alternative Topology (LISP+ALT)," draft-ietf-lisp-alt-10, Internet Engineering Task Force, Dec. 2011, work in progress.

[19] L. Jakab, A. Cabellos-Aparicio, F. Coras, D. Saucez, and O. Bonaventure, "LISP-TREE: A DNS Hierarchy to Support the LISP Mapping System," *Selected Areas in Communications, IEEE Journal on*, vol. 28, no. 8, pp. 1332 –1343, october 2010.

[20] F. Coras, A. Cabellos-Aparicio, J. Domingo-Pascual, F. Maino, and D. Farinacci, "Locator/ID Separation Protocol (LISP)," draft-coras-lisp-re-00, Internet Engineering Task Force, Jul. 2012, work in progress.

[21] S. Shi, J. Turner, and M. Waldvogel, "Dimensioning server access bandwidth and multicast routing in overlay networks," in *NOSSDAV, Proceedings ACM*, 2001, pp. 83–91.

[22] S. Banerjee, C. Kommareddy, K. Kar, B. Bhattacharjee, and S. Khuller, "Construction of an efficient overlay multicast infrastructure for real-time applications," in *Proc. IEEE INFOCOM*, 2003, pp. 1521–1531.

[23] S. Banerjee, B. Bhattacharjee, and C. Kommareddy, "Scalable application layer multicast," in *SIGCOMM, Proceedings ACM*, 2002.

[24] "SopCast traffic captures used to evaluate Lcast." [Online]. Available: http://www.cba.upc.edu/lcast/traces

[25] M. Faloutsos, P. Faloutsos, and C. Faloutsos, "On power-law relationships of the internet topology," in *SIGCOMM*, 1999.

[26] X. A. Dimitropoulos, D. V. Krioukov, and G. F. Riley, "Revisiting Internet AS-level Topology Discovery," in *PAM*, 2005, pp. 1–13.

[27] G. Huston, "AS6447 BGP routing table analysis report." [Online]. Available: http://bgp.potaroo.net/as6447

[28] H. V. Madhyastha, E. Katz-Bassett, T. Anderson, A. Krishnamurthy, and A. Venkataramani, "iPlane Nano: path prediction for peer-to-peer applications," in *NSDI, Proceedings USENIX*, 2009, pp. 137–152.

[29] K. Sripanidkulchai, B. Maggs, and H. Zhang, "An Analysis of Live Streaming Workloads on the Internet," in *IMC*, 2004.

[30] L. Vu, I. Gupta, J. Liang, and K. Nahrstedt, "Measurement and modeling of a large-scale overlay for multimedia streaming," *QSHINE*, 2007.

[31] E. Veloso, V. Almeida, W. Meira, and A. Bestavros, "A hierarchical characterization of a live streaming media workload," *TON*, vol. 14, no. 1, pp. 133–146, Feb. 2006.

[32] T. Silverston, L. Jakab, A. Cabellos-Aparicio, O. Fourmaux, K. Salamatian, and K. Cho, "Large-scale measurement experiments of P2P-TV systems insights on fairness and locality," *Signal Processing: Image Communication*, vol. 26, no. 7, pp. 327 – 338, 2011.

[33] S. Savage, A. Collins, E. Hoffman, J. Snell, and T. Anderson, "The end-to-end effects of internet path selection," *ACM SIGCOMM Computer Communication Review*, vol. 29, no. 4, pp. 289–299, 1999.

[34] C. Lumezanu, R. Baden, N. Spring, and B. Bhattacharjee, "Triangle inequality and routing policy violations in the internet," in *PAM*. Springer, 2009, pp. 45–54.

[35] S. Deering, "Host extensions for IP multicasting," RFC 1112 (Standard), Internet Engineering Task Force, Aug. 1989, updated by RFC 2236.

[36] Y. hua Chu, S. Rao, S. Seshan, and H. Zhang, "A case for end system multicast," *Selected Areas in Communications, IEEE Journal on*, vol. 20, no. 8, pp. 1456 – 1471, oct 2002.

[37] D. Tran, K. Hua, and T. Do, "A peer-to-peer architecture for media streaming," *Selected Areas in Communications, IEEE Journal on*, vol. 22, no. 1, pp. 121 – 133, jan. 2004.

[38] M. Castro, P. Druschel, A.-M. Kermarrec, and A. Rowstron, "Scribe: a large-scale and decentralized application-level multicast infrastructure," *Selected Areas in Communications, IEEE Journal on*, vol. 20, no. 8, pp. 1489 – 1499, oct 2002.

[39] "UUSee." [Online]. Available: http://www.uusee.com/

[40] X. Zhang, J. Liu, B. Li, and Y.-S. Yum, "Coolstreaming/donet: a data-driven overlay network for peer-to-peer live media streaming," in *INFOCOM, Proceedings IEEE*, vol. 3, march 2005, pp. 2102 – 2111.