A Distributed Inter-Domain Control System for Information-Centric Content Delivery

Wei Koong Chai, *Member, IEEE,* George Pavlou, *Fellow, IEEE,* George Kamel, *Member, IEEE,*
Konstantinos V. Katsaros, *Member, IEEE,* Ning Wang, *Senior Member, IEEE*

***Abstract*—The Internet, the *de facto* platform for large-scale content distribution, suffers from two issues that limit its manageability, efficiency and evolution: (1) The IP-based Internet is host-centric and agnostic to the content being delivered and (2) the tight coupling of the control and data planes restrict its manageability, and subsequently the possibility to create dynamic alternative paths for efficient content delivery. Here we present the *CURLING* system that leverages the emerging *Information-Centric Networking* paradigm for enabling cost-efficient Internet-scale content delivery by exploiting multicasting and in-network caching. Following the *software-defined networking* concept that decouples the control and data planes, CURLING adopts an inter-domain hop-by-hop content resolution mechanism that allows network operators to dynamically enforce/change their network policies in locating content sources and optimizing content delivery paths. Content publishers and consumers may also control content access according to their preferences. Based on both analytical modelling and simulations using real domain-level Internet subtopologies, we demonstrate how CURLING supports efficient Internet-scale content delivery without the necessity for radical changes to the current Internet.**

***Index Terms*—Information-centric networking, , future Internet, software-defined networking, content delivery.**

I. Introduction

WITH the proliferation of content distribution and streaming services, the IP-based Internet is now *the* network that offers global access to digital content. Besides high-end professional or enterprise content producers and broadcasters, there are now new groups, ranging from individual amateurs to small/medium content providers that exploit the Internet for distributing their content (usually via third-party platforms)[1]. Nowadays, streaming services can often be of long duration, lasting several hours or even continuously (*e.g.*, wildlife monitoring). New Internet stakeholders [1] have also created complex relationships amongst them with different and sometimes conflicting interests. This has made the issue on efficient content delivery more challenging due to the need to consider multi-party interests [2]. Nevertheless, the inherent design focus of the Internet on inter-connecting hosts with tightly integrated control and data planes limits its flexibility and constrains its future evolution. In the current host-to-host Internet model, a content consumer must first obtain the specific network location of the content source in order to be able to access the content itself [3]. Based on the current domain name system (DNS) and IP-based access, Internet Service Provider (ISP) networks essentially act as mere 'bit pipes', delivering content flows from DNS-resolved content sources to the requesting consumers.

The inherent coupling of the control and data planes, as manifested by the integrated routing and forwarding functionalities in network layer devices, constrains network operators from efficiently managing their network resources to serve user demands, *e.g.*, by dynamic configuration and adaptation of communication end-points of content delivery paths subject to network and traffic conditions. ISPs also have limited flexibility in coping with the increasingly high network utilization. As a result, many ISPs resort to continuously upgrading their network capacity in a manner that cannot be sustainable in the long term. This is further exacerbated by the fact that networking technologies that were designed to support efficient content delivery and streaming services, such as IP multicast [4], have not enjoyed significant deployment under the current host-to-host Internet model due to inter-domain scalability and deployment problems [5].

In contrast to these inherent limitations in IP networks, especially with respect to the efficient delivery of (live) content, *information-centric networking* (ICN) has been proposed as an alternative networking paradigm [6]. In ICN, content is explicitly named and thus can be identified independent of hosting nodes. This results in *location-independence*, which enables consumers to request content from the network without having to first locate a server that hosts it. This facilitates the flexible management of the network and fine-grained control of the content discovery and delivery processes by network operators, departing from the 'bit pipe' model. Subsequently, ICN supports natively important desired features, such as multicast and in-network caching, for the efficient wide-scale distribution of multimedia content. A number of ICN architectures have been proposed in recent years, following sometimes slightly different approaches towards the overarching ICN aims [6], but all of them exhibiting key limitations. The prevailing Content-centric Networking / Named Data Networking (CCN/NDN) approach [7] still lacks scalable support for name-based routing at inter-domain level [8]. Other approaches that focused on scalable and flexible inter-domain operation provide limited compatibility with the current IP-based model and/or inefficient performance [9][10][11]. Moreover, in view of a multi-actor environment such as the Internet, information-centricity calls for the necessary flexibility to control both the discovery and the eventual delivery of information/content. A careful investigation of the current state-of-the-art clearly shows that limited attention has been paid in finding an evolutionary path for the transition towards a flexible, ICN-oriented Internet (See Section VII for further details).

In this paper, we propose an ICN-based control system, *CURLING* (*Content-ubiquitous Resolution and Delivery In-*

[1] For instance, livestream.com offers a cloud-based broadcasting platform to broadcasters of any size.

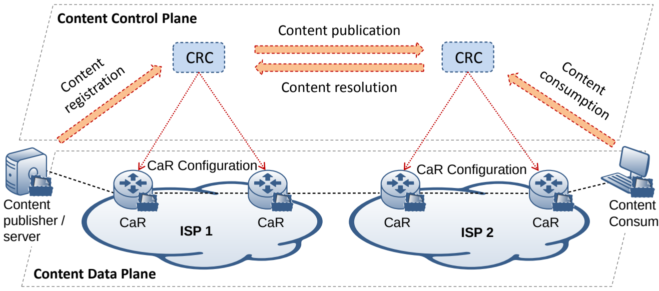


Fig. 1. Overview of CURLING.

frastructure for Next Generation Services), specifically designed for the efficient and flexible content distribution at Internet scale. Its design follows the spirit of software-defined networking (SDN), targeting the decoupling of the control and data planes [12] for enabling fine grained flow steering and enhancing the network management and control capabilities of ISPs. In CURLING, a logical central controller in each autonomous system (AS)[2] – known as a *Content Resolution Controller (CRC)* – handles incoming content requests for identifying the best possible content sources and at the same time, performs on-the-fly content-path configurations at the router level within its AS. *Content-aware routers (CaRs)*, which operate at the edges of the AS, forward content objects based on the content states, records and rules set up by the local CRC during the resolution phase. As such, CURLING is a system-of-systems comprised of an individual system within each AS (*i.e.*, the CRC). Following the ICN paradigm, end-hosts simply indicate "*what*" they want to their local CaR and CRCs that belong to different ASs coordinate with each other to support content access and delivery across AS boundaries, thereby forming a distributed, inter-domain control plane. This is achieved through gossip-like communication between ISP networks according to their business relationships (*i.e.*, provider, customer and peer). To further guide the establishment of the forwarding paths on the data plane, CRCs collect information related to content availability and popularity, content server load, border gateway protocol (BGP) / IP routing, traffic and cache conditions.

By supporting policy-based content routing, CURLING introduces significant flexibility for ISPs to support content provider and customer preferences and policies by taking into account both end-user preferences and local network management policies of ISPs. CURLING also offers a flexible way to construct inter-domain multicast trees for streaming content without relying on specific IP multicast addresses. By adopting the ICN paradigm, CURLING inherits the natural capability of in-network caching, thus providing further possibilities for improving content distribution. Although CURLING uses a gossip-like resolution mechanism similar to that of some clean-slate ICN approaches such as DONA [13], it achieves this in the overlay content control plane formed by all the CRCs (see Fig. 1) while at the same time it uses conventional IP routing underneath, maintaining backward compatibility with the current Internet and facilitating actual deployment. CURLING then only requires the enhancement of content-aware routers at domain edges so that they are able to cache content (which already happens today), keep content state for supporting reverse path streaming including multicast, and intercept and handle end-host requests for content[3]. As CURLING was designed for operation on top of the current inter-domain Internet infrastructure, it also respects business relationships between domains and adheres to BGP routing. It is noted that recent efforts have also started looking into incremental deployability of ICN (*e.g.*, [14]), often employing an SDN approach. However, the focus has been on baseline functionalities such as forwarding, caching, congestion control, *etc.*, and not on important inter-domain aspects such as the flexible and scalable *inter-domain* content discovery and delivery. CURLING fills this gap by applying the control-data plane separation principle of SDN at a global level.

CURLING was first introduced in our preliminary work [15] which described its basic design principles. In this paper, we present the complete design, formal specification and detailed evaluation of the approach. Our main contributions include:

- Design of CURLING, that is based on the ICN principles and adopts the emerging SDN approach for its instantiation for supporting Internet-wide content delivery (See Section II). The formal specifications (including pseudocodes) on both content publication and consumption algorithms are presented in Section III and IV.
- Modelling of our approach that proves the adherence of CURLING to current inter-domain routing and business relationships between domains as well as quantifying the gain from the path optimization mechanism (See Section V). The modelling also represents a tool for investigating the performance of the proposed scheme in the presence of business relationships as well as AS-level topological changes at a global scale.
- Detailed evaluation of our approach including content resolution latency, the gain from the path optimization mechanism and scalability against current DNS and another inter-domain ICN-based architectures. Our evaluations have led to a number of key observations which are reported in Section VI, such as better scalability against other inter-domain ICN-based architectures and comparable infrastructure requirements and resolution latency to the current DNS.

II. CURLING System Overview

The CURLING system follows SDN principles in physically decoupling the content control and data planes (see Fig. 1), allowing thus for fine-grained control of data flows from content publishers to content consumers.

A. Control plane

Each AS maintains a logically[4] centralized controller (a CRC). CRCs interface with CRCs in neighbouring ASs, and CaRs / content providers in the local domain. Inter-CRCs interfaces follow the underlying business relationships between domains (*e.g.*, provider-customer and peering relationships). The

[2] In this paper, the terms *AS* and *domain* are used interchangeably.

[3] Such functionality can be easily supported over the existing TCP/IP protocol stack *e.g.*, through TCP/HTTP proxies.

[4] Multiple synchronized mirror CRCs can be maintained for resilience against the failure of the primary CRC in the domain.

resulting set of CRCs throughout the inter-domain topology, constitute the *inter-domain CURLING control plane*. Within each domain, CRCs constitute a second level, local control plane, that focuses on the management of the local, data plane resources through a southbound interface with CaRs.

The control plane operations in CURLING supports the full content lifecycle, namely:

1) Content registration: Content providers initially register their content with their access ISPs through their local CRC. Upon registration, ISPs collect information about the name and location of the content available in their domain.

2) Content publication: The information collected via content registration is propagated in the inter-domain control plane of CRCs to denote its availability. During this process, CURLING allows content providers to explicitly specify their preferences on the network locations (reflected by IP prefixes) for content access, *scoping* their content to be accessible only by its local consumers or by domains in a given region of the Internet. One example is the BBC iPlayer application which is only accessible by subscribers to a UK-based ISP. Advanced content privacy/security mechanisms can be deployed on top of the CURLING system to support application scenarios such as location-independent access (*e.g.*, Netflix subscription). Content scoping can even be applied by the underlying ISPs to optimize inter-domain content delivery paths across domains (Details see Section V-C). This key feature allows network operators to optimize the use of their network resources.

3) Content resolution: Content consumers access the desired content by triggering the content resolution process with an object-level content request. CRCs forward resolution requests in the global control-plane until the content is located. In this process, a content consumer may specify wanted or unwanted locations as possible content sources, essentially *filtering* specific available content sources.

4) Content delivery - data plane configuration: During the content resolution process, the control plane of CRCs identifies the inter-domain path(s) for the delivery of the content to the content consumer(s). At this stage, each CRC in the resolution chain needs to configure its own underlying CaRs by installing content forwarding states for content delivery. The established data plane configuration adheres to the scoping and filtering requirements of the content providers and consumers as well as the local policies of each domain (*e.g.*, selection of ingress and egress routers). Upon successful content resolution, the end-to-end content delivery path from the resolved source to the consumer is already in place and ready for content transmission. See Section III-C for details.

The above operations enable the discovery and delivery of content across the Internet. In addition, CRCs further collect information for the support of efficient content traffic engineering. This includes routing policies information (in a similar fashion to Route Reflectors [16]) and information on the current load in the network. Moreover, in distributed realizations of CRCs, information regarding the content popularity dynamics can be exchanged. Such information can assist the formation of multicast groups or support content resolution decisions (*e.g.*, steering requests to pre-fetched content).

B. Data plane

In the data plane, a set of CaRs residing at the borders of each domain is responsible for forwarding content traffic. A southbound interface between the CRC and the CaRs is used for setting up content forwarding states on a per-content-session basis. Such configuration specifies the next-hop CaR in the direction towards the consumer. The maintenance of the state information also enables content multicast support, both within and across domains. The key difference with traditional IP multicast is that the CURLING construction and manipulation of the multicast tree in the control plane is controlled by the CRC, which is decoupled from the data plane in which CaRs act merely as content forwarders that are *agnostic* to content resolution. This physical decoupling of the control and data planes offers the flexibility of enabling different content distribution policies at the CRC. For better scalability, CRCs may also be realized as virtualized entities running in a cloud similar to Path Computation Elements (PCEs) whose role is to compute paths on behalf of nodes in the network [17][18]. In this case, issues of single point of failures and scalability can be circumvented through virtualized CRC infrastructure.

The data plane configuration is established along with the content resolution process. It is triggered by an information-centric type of interaction with content consumers. Content consumers simply indicate the desired content to their local CRC, without establishing a communication path with the content provider. By explicitly using content names, CURLING enables object-level in-network caching, which in fact enables a global ISP-operated "CDN" and is in contrast with the packet-/chunk-level caching of other ICN architectures.

III. Content Access

A. Background

As briefly mentioned, our design exploits the business relationship established amongst ISPs [19]. The conventional view of the Internet AS topology is that it is inherently hierarchical [20], [21]. The relationship between two connected ASs is generally classified as one of the following:

- provider-to-customer (p2c) (or in the reverse direction, customer-to-provider (c2p)) – customer pays the provider to obtain transit through the provider domain
- peer-to-peer (p2p) – two peering ASs both pay for the maintenance for the link between them
- sibling-to-sibling (s2s) – ASs having mutual transit agreement such as merging ISPs.

These business relationships are driven mainly by economic considerations. Two ASs having a p2p relationship may decide to share the maintenance cost of their shared link, while an AS may simply buy transit capacity from a provider (forming p2c and c2p links). Under such structure, ASs are organized into tiers. Tier-1 ASs are those without any provider, forming a fully-meshed peered topology. A customer of a tier-i AS is classified as a tier-$(i+1)$ AS. For multihomed ASs, their tier levels are determined by their lowest-tiered provider (*i.e.*, tier-$[\max(A)+1]$ where A is the set of tier numbers of all providers of the AS). Fig. 2 illustrates the hierarchical structure of the inter-domain topology.

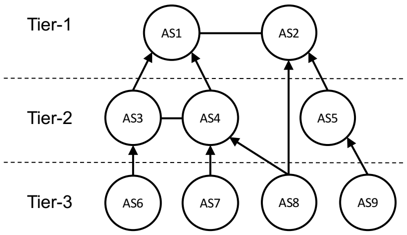


Fig. 2. Hierarchical structure of inter-domain topology: Domain AS3 and AS4 are peers while domain AS3 is customer to domain AS1

B. Content Registration and Publication

The content publication operation follows the *provider-route forwarding rule* which resolves content requests in a guaranteed manner if a content source exists (see Lemma 1 in Section V-B). It exploits the business relationship between ASs so that the requests received by a CRC in one domain should be passed only to its counterpart CRC(s) in its provider domain(s). Effectively, neighboring domains connected via peering/sibling links are excluded from the dissemination of the content publication information, so as to reach tier-1 domains that possess knowledge of all published content.

For scalability, only information of live streaming content or of popular static content is passed upwards to tier-1 domains and this amount of information can certainly be handled at the tier-1 level. This means that static content popularity needs to be monitored so that when a threshold is exceeded and a piece of content becomes "popular", relevant information is propagated upwards and this content can also be cached in domains other than the source one. When popularity drops below the threshold, this information is purged and the content can only be accessed at its source domain. The precise mechanisms for keeping track of content popularity, which are not simple because of in-network caching, are outside the scope of this paper. We see cache management approaches that consider content popularity and request locality (*e.g.*, [22][23]) to be complementary to CURLING.

The publication of a new content object consists of two stages, where a *content object* here may refer to either the name of a specific pre-recorded video in the case of video-on-demand applications, or the name of a broadcasting channel in the case of live-streaming applications:

Stage 1 – Content Registration: The content provider initiates the publication of a new content object by issuing a `Register` message to its local CRC. A new content record entry is created in its content management repository containing (1) a globally unique[5] content identifier (CID) assigned to that content and (2) the explicit location of the content (*i.e.*, the IP address of the content server).

Stage 2 – Content Publication: For the first-hop CRC that receives a direct `Register` message from the local server, a content record is created for which the `next_hop` points to the IP address of the content server. This CRC is then responsible for publishing the content globally. This is achieved through the dissemination of the `Publish` message across CRC(s) in individual domains according to the provider-route forwarding rule. The pseudocode for publishing content at intermediate CRCs is given in Algorithm 1. Each CRC disseminates a new `Publish` message towards its counterpart in the provider domain(s) until it reaches a tier-1 AS. Each CRC receiving a new `Publish` message updates its content management repository with a new record entry containing the CID and the implicit location of the content (*i.e.*, the IP prefix of the previous-hop domain from which the `Publish` message was received). Following this, each CRC effectively knows the locations of all the content within its own domain (explicitly) and those under it (its explicit or implicit customer domains) *i.e.*, within its *customer cone* [24]. However, peer domains do not know the content records of each other. If a publisher has a scoping requirement, *i.e.*, the content is allowed to be accessible only by consumers with certain IP prefixes, then the `INCLUDE` option with the prefix should be included in the `Publish` message (line 6). If an incoming `Publish` message carries a previously published (or known) CID, this `Publish` message is simply dropped (lines 2-5).

Algorithm 1 Pseudocode to process `Publish` primitives at CRCs

```
Publish – the received Publish message
INCLUDE_Prefix – the scoping option (prefixes) included in the corresponding Publish message
My_Prefix – the domain prefix where the CRC resides
CID – content identifier included in the Publish message
in_link – the CRC-CRC link the Publish message is received
out_links – all CRC-CRC links excluding in_link
next_hop – next-hop CRC towards the original content publisher
 1: if (Publish received) then
 2:   if duplicate then
 3:     drop(Publish);
 4:     return;
 5:   end if
 6:   if (My_Prefix $\subseteq$ INCLUDE_Prefix or INCLUDE_Prefix==Null) then
 7:     next_hop = in_link;
 8:     new(content_record, CID, next_hop);
 9:     for all (out_link==‘provider’) do
10:       send(Publish, out_link);
11:     end for
12:   else if (My_Prefix $\nsubseteq$ INCLUDE_Prefix) then
13:     drop(Publish);
14:     return;
15:   end if
16: end if
```

C. Content Resolution

A consumer initiates the content resolution process at the content control plane via a `Consume` message with the desired content identifier. The primary resolution procedure (See Algorithm 2) follows the same provider-route forwarding rule, *i.e.*, the `Consume` message is forwarded to its provider domain(s) if the local CRC cannot find the content in its own repository. For ASes having multiple providers, the `Consume` message may be forwarded to all its providers (*i.e.*, *broadcast resolution mode*, lines 24–26) or to one of its providers based on some preset settings or simply at random (we refer this as the *random resolution mode*, lines 27–30). For a tier-1 ISP domain (*i.e.*, ASs without any provider) that is unaware of the content location, the request is broadcasted to all its peering and sibling domains until the `Consume` message is

[5] Uniqueness can be established via cryptographic hashing of the content.

Algorithm 2 Pseudocode to process Consume primitives at CRCs

```
Consume – the received Consume message
INCLUDE_Prefix – the scoping option (prefixes) included in the corresponding Consume message
EXCLUDE_Prefix – the filtering option (prefixes) included in the corresponding Consume message
My_Prefix – the domain prefix where the CRC resides
CID – content identifier included in the Consume message
in_link – the CRC-CRC link the Consume message is received
out_links – all CRC-CRC links excluding in_link
next_hop – next-hop CRC towards the original content publisher
 1: if (Consume received) then
 2:     if (content record for CID found) then
 3:         if (My_Prefix $\subseteq$ INCLUDE_Prefix and My_Prefix
            $\not\subseteq$ EXCLUDE_Prefix) then
 4:             get(next_hop);
 5:             send(Consume, next_hop);
 6:         else
 7:             drop(Consume);
 8:             return;
 9:         end if
10:     else
11:         if (INCLUDE_Prefix != null) then
12:             send(Consume, based on BGP route toward
                INCLUDE_Prefix);
13:         else if (get("provider")==null) then
14:             if (in_link=="peer") then
15:                 send(Error, in_link);
16:                 return;
17:             else
18:                 for all (out_link=="peer") do
19:                     send(Consume, out_link);
20:                 end for
21:             end if
22:         else if (get(provider)!=null) then
23:             if (local_policy=="broadcast") then
24:                 for all (out_link=="provider") do
25:                     send(Consume, out_link);
26:                 end for
27:             else if (local_policy=="random") then
28:                 next_hop=random(1, out_link=="provider");
29:                 send(Consume, next_hop);
30:             end if
31:         end if
32:     end if
33: end if
```

delivered to a source holding the requested content (lines 14–21). If the content is not found after the entire resolution process (*e.g.*, content has been removed or request does not satisfy the preferences of the consumer or content provider), an Error message is returned to the consumer through reverse-path forwarding to indicate a resolution failure (line 15).

We define two distinct content resolution stages:

Stage 1 – Uphill: the forwarding of a Consume message 'upwards' along the provider-route until it reaches a domain whose CRC has the record entry for the requested CID.

Stage 2 – Downhill: the forwarding of the Consume message from the domain whose CRC recorded the requested CID 'downwards' to the content server that hosts the content.

During the uphill stage, a CRC may apply its local policy to selectively forward it to the CRC of its neighboring domain(s). This is especially the case for *multihomed* domains. For instance, a multihomed domain may forward the Consume request to the provider that incurs the least content delivery path length within the local domain. Such a 'hot-potato' resolution policy minimizes the interior gateway protocol (IGP) distance (or the hop count) between the local ingress CaR and the egress CaR within its own network.

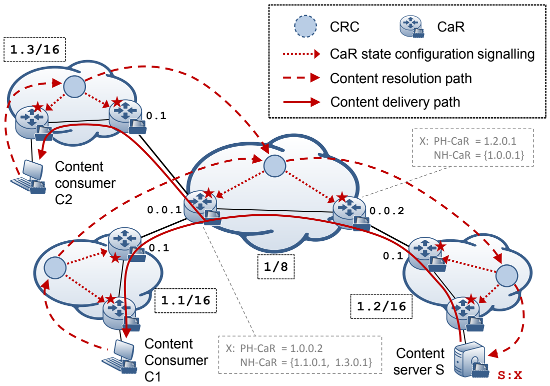


Fig. 3. Basic content delivery path construction.

Regarding consumer preferences of content sources, CURLING provides three options for content resolution:

Scoping allows a consumer to indicate the preferred ISP network(s) as the source domain of the requested content. The INCLUDE option in the Consume message is used to convey the IP prefixes from which the consumer would like to receive the content[6]. Since explicit IP prefixes for a candidate content source is carried in the Consume message, the corresponding resolution process becomes straightforward: each intermediate CRC only needs to forward the request towards the targeted IP prefix(es) directly according to the underlying BGP routes, and not following the default provider-route forwarding rule. In case multiple inter-domain routes are available towards a specific prefix, the most explicit one will be followed, as consistent with today's inter-domain routing policy. A special use of the scoping function is to be applied by a CRC for the purpose of inter-domain route-optimization once the content source has already been resolved (see Section IV-B for details).

Filtering is complementary to scoping. Instead of specifying the preferred networks, the consumer indicates unwanted domains as potential content sources. It is supported via the EXCLUDE option in Consume messages. In contrast to the scoping scenario in which a Consume message is explicitly routed towards the desired IP prefix(es) according to the BGP route, in the filtering case, a request is routed based on the business relationship between domains, following the provider-route forwarding rule.

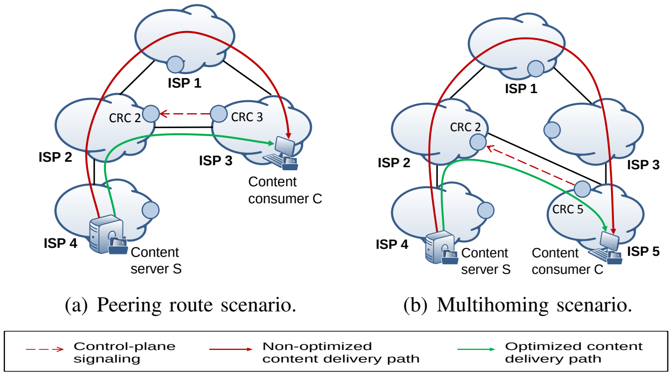
The **Wildcard** mode is the default option for Consume messages. It indicates that the consumer does not have a preference on the content source. Similar to the filtering scenario, the resolution on wildcard content requests follows the provider-route forwarding rule.

IV. Content Delivery

A. Basic Content Delivery

We follow a state-based approach to enforce content delivery paths. It lends itself to the realization of receiver-driven

[6] It is not always required that content Consumers know the actual IP prefix of the domains they prefer. Their local CRCes may translate the "region information" (domain names) into IP prefixes through standard DNS services.



(a) Peering route scenario.

(b) Multihoming scenario.

Fig. 4. Scenarios for optimizing inter-domain content delivery paths.

inter-domain multicast, but with the physical decoupling of the control plane at the CRC side from the data plane at the CaR side. The path construction follows a 'breadcrumbs' paradigm in which the resolution process at the control plane leaves a trail of states at the data plane to allow the content to be delivered back to the requesting user along the reverse of the content resolution path (cf. Definition 3 in Section V-C). It is worth stressing that CRCs do not directly constitute the content delivery paths, in which case configuration interaction between the CRC and local ingress or egress CaRs is necessary.

As described in Section III-C, `Consume` requests are resolved through a sequence of CRCs according to either the business relationships between ISPs (in wildcard and filtering modes) or the BGP reachability information on the scoped source-prefix (in scoping mode). In both cases, once a CRC has forwarded the content `Consume` request to its next-hop counterpart, it needs to configure the local CaRs that will be involved in the content delivery. A CaRs configuration is realized by a mapping between the resolved content identifier and the IP address of the next-hop CaRs. It is noted that the established configuration refers only to the involved CaRs allowing the transparent, information-agnostic forwarding by intermediate routers, based on IP primitives. In case of failed content resolution, content states temporarily maintained at CaRs can either passively time-out or be actively torn-down by the local CRC. The determination of ingress or egress CaRs for each content `Consume` request is based purely on the BGP reachability information across networks.

Let's take Fig. 3 for illustration. Content consumer C1 (attached to domain 1.1/16) has requested a live-streaming content X from server S (attached to domain 1.2/16). The content delivery path for X traverses from 1.2/16 to 1.1/16 via 1/8 and each of the corresponding ingress or egress CaRs is associated with a content state (indicated by $\star$ in the figure) that is maintained for the delivery of this content. Each content state at the CaR side includes a unique *previous-hop-CaRs (PH-CaR)* pointing to the neighboring CaR in the direction of the server, while a set of next-hop-CaRs (NH-CaRs) lead towards one or multiple content consumers. This is similar to the style used in the maintenance of incoming and outgoing interface information in conventional IP multicast. Within each domain, the communication between the logically connected ingress and egress CaRs can be achieved either by establishing intra-domain tunnels that traverse non content-aware core IP routers, or natively through the content-centric network routing protocols [7]. The states are configured by the local CRCes during the content resolution phase.

The maintenance of content states allows inter-domain multicast content delivery since a content is requested via its name and any network element having a copy of the content can serve the request. Referring still to Fig. 3, assume that content consumer C2 (attached to domain 1.3/16) issues a `Consume` request for the same content X. Upon receipt of the request, the local CRC forwards it to its provider counterpart in domain 1/8, as it is unaware of the direction toward the content source. Since the CRC in 1/8 knows that the content flow for X is being injected into the local network via the originally configured egress CaR 1.0.0.1, it then updates its outgoing NH-CaR list by adding a new egress 1.3.0.1 leading towards C2. Thus, a new branch is established from CaR 1.0.0.1 which is responsible for delivering the content to C2 without any further content resolution process (*i.e.*, the `Consume` message from C2 is not forwarded onwards from 1/8).

B. Inter-domain Route Optimization

Once the content source is identified, there may be opportunities to route content along shorter paths than the original content resolution path. Inter-domain path optimization is achieved by leveraging (1) peering and (2) multihoming routes. In Fig. 4(a), the initial content delivery path is via ISP 1, following the provider-route forwarding rule specified in Section III. However, when ISP 3 realizes that a peering route is available towards ISP 2, a shorter AS path that bypasses ISP 1 can be constructed. In Fig. 4(b), assume that ISP 5 (where the content consumer is attached) applies the *random* resolution mode and happens to follow the sub-optimal provider-route via ISP 3 during the initial content resolution. In this case, optimized content delivery paths can also be constructed by performing resolution following the route via ISP 2. In both cases, follow-up content resolution via alternative shorter paths can be performed *after* the actual content source has been identified as the result of the initial resolution.

A CRC interacts with its local CaRs (omitted in Fig. 4 for clarity) for route optimization if possible. Although CRCs do not directly participate in content traffic forwarding as CaRs, they have the knowledge on inter-domain routing similar to a route reflector (RR) in BGP. Once a CRC noticed that the content flow with source address belonging to a remote prefix has been injected into the local domain, and it also knows from the BGP routing information that there exists a shortcut path (either via a peering route or via an alternative provider-route), it may issue a new *scoping-based* `Consume` message to initiate the path optimization process. Specifically, this `Consume` message, carrying the IP prefix containing the resolved content source, is sent in the alternative shortcut direction. Consider Fig. 4(a): once CRC 3 detects a peering-route towards the resolved content source, it issues a new `Consume` message with an `INCLUDE` option containing the IP prefix of ISP 4, and sends the request to CRC 2 via the peering route. CRC 2 installs the content state at its border CaR linking with ISP 3. Upon issuing the request, CRC 3 also installs content state at its new ingress CaR which peers with ISP 2. Such content configurations by CRC 2 and CRC 3 follow the specifications described in Section IV-A. Once the content flow is injected into ISP 3 via the peering domain, CRC 3 will tear down the original content delivery path by sending a tear-down request to its counterpart in ISP 1, followed by deletion of content states at the corresponding CaRs along the original delivery path.

V. Modelling

A. Preliminaries

The Internet AS topology can be represented as a Type-of-Relationship (ToR) graph [25]. Let $G = (V, E, R)$ be a ToR graph with $V = v_1, ..., v_{|V|}$ nodes and $E = e_1, ..., e_{|E|}$ links with each link associated with a relationship, $R = \{p2c, c2p, p2p\}$. The network is thus of size $|V|$ with $|E|$ links.

We base our study on real Internet topologies. Recent research indicated that peering relationships are increasing [26][27]. To account for this ongoing evolution of the Internet topology, we complement our analysis and evaluation with results based on the uniform recursive tree (URT). For Internet AS topology, [28] has shown empirical matching properties between Internet topology measurements and the properties of the URT. Being intrinsically a tree, URTs naturally lend itself to the modeling of the Internet AS topology with consideration of the business relationships of the different ASs. Starting with one root node, a URT is constructed by randomly connecting a new node to an existing one. The union of the shortest paths between all node pairs, $G_{\cup SPT}$, will be exactly the same as the underlying URT substrate since the clustering coefficient, with such construction is zero.

For this study, we make two modifications to the original URT. First, we introduce a constraint to limit the number of tiers allowed for the URT to reflect the real Internet topology of having a low number of tiers. Second, in addition to the links included in the construction of the URT, we randomly add links between two nodes at the same level to act as peering links. With these modifications, we add to the URT the small-world effect [29] found in the Internet topology, since with the added peering links forming triangles in the network, the clustering coefficient is no longer zero. Furthermore, in our modified URT (mURT), each link is defined with an attribute denoting its type (*i.e.*, $R = \{p2c, c2p, p2p\}$). The aim is not to realistically model the *current* Internet but rather to allow us to validate our modeling results in a controllable manner that takes into account the evolution of the Internet, in terms of the continual increase of peering relationships between ASs.

B. Modeling Content Access

Following this model, we can formally define the content publication and resolution paths as specified in Sections III-B and III-C.

Definition 1 (Content Publication Path): [7] Given a content publication message, $Publish_k$, originating from domain v_i, then the corresponding content publication path, $p_k^{pub} = p_k^{pub(c2p)}$ where $p_k^{pub(c2p)}$ is the sequence of c2p links (uphill) from v_i to a Tier-1 AS, $\langle v_i, v_1\rangle, \langle v_1, v_2\rangle, ..., \langle v_{j-1}, v_j\rangle$ where $\langle v_x, v_y\rangle$ denotes the link between node v_x and v_y.

From the sequence, v_j must be a Tier-1 AS. If v_i is a Tier-1 AS, then $v_i = v_j$ and $p_k^{pub} = 0$. Let $|p|$ denote the length of path p, then $|p_k^{pub}|_{max} = D(G_{\cup p^{pub}})$ where $D(G)$ is the diameter of graph G and $G_{\cup p^{pub}}$ is the resultant shortest path tree of the union of all possible content publication paths.

Definition 2 (Content Resolution Path): Given a content request req_k, from domain v_i, then the corresponding content resolution path, $p_k^{res} = p_k^{res(c2p)} + p_k^{res(T1p2p)} + p_k^{res(p2c)}$.

$p_k^{res(c2p)}$ is the sequence of c2p links (uphill) from v_i towards a Tier-1 AS, $\langle v_i, v_1\rangle, \langle v_1, v_2\rangle, ..., \langle v_{j-1}, v_j\rangle$ where $\langle v_x, v_y\rangle$ denotes the link between nodes v_x and v_y. $p_k^{res(T1p2p)}$ is always empty except if v_j is a Tier-1 AS and does not hold the corresponding content record. In this case, $p_k^{res(T1p2p)} = \langle v_j, v_{j+1}\rangle$ is the one-hop peering link where v_{j+1} holds the content record. $p_k^{res(p2c)}$ is the sequence of p2c links (downhill) from v_j (or v_{j+1} if $p_k^{res(T1p2p)} \neq \emptyset$) towards a destination that can satisfy req_k. $p_k^{res(c2p)} = \emptyset$ if the requested content is found at the origin AS, v_i, indicating that either the content is hosted within v_i (in this case, $p_k^{res(p2c)} = p_k^{res(T1p2p)} = \emptyset$ or it is hosted in one of customer ASs of v_i (only $p_k^{res(T1p2p)} = \emptyset$)).

Lemma 1: Based on the provider-route forwarding rule, a content resolution path, p_k^{res}, relating a req_k to any source-destination pair, v_i and v_j in G, is guaranteed to exist if G is connected and all Tier-1 ASs form a full mesh.

Proof 1: According to the provider-route forwarding rule, the resolution path, $p_k^{res(c2p)}$ will reach a Tier-1 AS *iff* the requested content of req_k is not found along the c2p link-chain. Since Tier-1 ASs have peering relationship with all other Tier-1 ASs, then in the worst case, req_k will be resolved at Tier-1. By definition, the last hop of $p_k^{res(c2p)}$ and the first hop of $p_k^{res(p2c)}$ is either a common AS or two peered Tier-1 ASs. Hence, there is a valid path from v_i to v_j since all ASs must have a valid path to Tier-1. □

Lemma 2: Let $G = (V, E, R)$ be a type-of-relationship (ToR) instance of the Internet AS topology. Then, the expected longest resolution path (worst case) for a req_k is

$$|p_k^{res}|_{max} = 1 + \frac{2\,|V|}{|V|-1}\sum_{n=2}^{|V|}\frac{1}{n} \tag{1}$$

for any non-common source-destination pair ($v_i \neq v_j$) in G.

Proof 2: Recall that, by definition, a generic resolution path consists of three components: $p_k^{res(c2p)}$, $p_k^{res(T1p2p)}$ and $p_k^{res(p2c)}$. Hence, the hop count of the resolution path is

$$|p_k^{res}| = \left|p_k^{res(c2p)}\right| + \left|p_k^{res(T1p2p)}\right| + \left|p_k^{res(p2c)}\right|. \tag{2}$$

We can rewrite this equation as

$$|p_k^{res}|_{max} = 2E\left[\left|p^{|V|}\right|\right] + \left|p_k^{res(T1p2p)}\right|, \tag{3}$$

where $E\left[\left|p^{|V|}\right|\right]$ is the expected number of hops from an arbitrary chosen node in G of size $|V|$ to the Tier-1 AS following

[7] We assume the wildcard mode for all definitions, lemmas and theorems.

our provider-route forwarding rule. In [28], Section 16.3, it is shown that for any non-common node pair,

$$E\left[\left|p^{|V|}\right|\right] = \frac{|V|}{(|V|-1)}\sum_{n=2}^{|V|}\frac{1}{n}.$$

The p_k^{res} is the longest when $p_k^{res(c2p)}$ reaches a Tier-1 AS and req_k is still unresolved but needs further downhill resolution. Thus, $p_k^{res(T1p2p)} \neq \{\}$ and $\left|p_k^{res(T1p2p)}\right| = 1$. Substituting $\left|p_k^{res(T1p2p)}\right|$ and $E\left[\left|p^{|V|}\right|\right]$ into (3), we obtain (1). □

C. Modeling Content Delivery

We first provide the formal definitions of the inter-domain delivery paths.

Definition 3 (Unoptimized Content Delivery Path): Given a content resolution path, p_k^{res} for a req_k, the unoptimized content delivery path, p_k^{del}, is simply the reverse path of p_k^{res}. Formally, $p_k^{del} = \overleftarrow{p_k^{res}}$. These paths are constructed following the provider-route forwarding rule. Effectively, except if $p_k^{res(T1p2p)} \neq 0$, the peering links in the topology are not used since the content record information is not propagated via the peering links for scalability considerations.

Based on Definition 3, $\exists p_k^{del}$ if $\exists p_k^{res}$, meaning that if a request is resolvable, the corresponding content is deliverable.

Definition 4 (Optimized Content Delivery Path): The optimized content delivery path for a req_k is $p_k^{del*} = p_k^{del(c2p)} + p_k^{del(p2p)} + p_k^{del(p2c)}$ where the path follows our optimization rules and the resultant p_k^{del*} always abides to the valley-free routing property[8].

Lemma 3: For any content delivery path with either $p_k^{res(c2p)} = \emptyset$ or $p_k^{res(p2c)} = \emptyset$ for its corresponding resolution path, it holds that $\left|p_k^{del}\right|_{min} = \left|p_k^{res}\right|$.

Proof 3: The lemma states that a content delivery path of any req_k *cannot* be further optimized (*i.e.*, $p_k^{del} \equiv p_k^{del*}$) if either $p_k^{res(c2p)} = \emptyset$ or $p_k^{res(p2c)} = \emptyset$ for its corresponding resolution path. A resolution path with $p_k^{res(p2c)} = \emptyset$ (*i.e.*, no downhill hop in the resolution path) implies that the requested content is hosted within one of the provider domains along the provider-route. As such, the consideration of peering links along this path can never form a shorter path than the original resolution path. On the other hand, $p_k^{res(c2p)} = \emptyset$ (*i.e.*, no uphill hop in the resolution path) *iff* the requested content record is found at the origin domain where the content request is issued. In this case, the resultant delivery path is either exactly the publication path (*i.e.*, $p_k^{del} \equiv p_k^{pub}$) or coincides with the first hops of the p_k^{pub}. In both cases, the path length is already shortest and optimal. □

For any req_k where $\exists! p_k^{del}$, then the path is already optimal (*i.e.*, $p_k^{del} \equiv p_k^{del*}$). Now, we can state the following theorem.

Theorem 1: All paths constructed following the CURLING specifications are valid valley-free paths.

Proof 4: An AS path is said to have the valley-free property when it satisfies the conditions that

Cond. 1. no two or more peering links are used, and;

Cond. 2. once a path uses a p2c or p2p link, the rest of the path must follow p2c links [30].

A content publication path, by definition, consists of c2p links only. Thus, it adheres to the valley-free property.

A content resolution path, by definition, consists of three ordered components as follows: $p_k^{res(c2p)}$, $p_k^{res(T1p2p)}$ and $p_k^{res(p2c)}$. Only the $p_k^{res(T1p2p)}$ component of the path uses a p2p link and since this is a Tier-1 p2p link, it can only be a maximum of one link (satisfying condition 1, above). Furthermore, we see that p2c links are used only at the last leg of the path. Therefore, it also satisfies condition 2. Since by definition, $p_k^{del} = \overleftarrow{p_k^{res}}$, then the unoptimized content delivery path is also a valid valley-free path.

Finally, an optimized content delivery path is created based on BGP reachability information. Since BGP routes are valley-free, then the optimized paths are also valley-free. □

Next, we model the expected gain that can be obtained through our path optimization mechanism and validate it through both synthetic topologies and real Internet topologies in Section VI. Content requests are assumed to arrive in the network exogenously. The set of content requests is denoted by $REQ = req_1, \ldots, req_K$. We use *gain* as an indicator of hop-count-reduction after the optimization operation. To enable direct comparative study across different topologies and settings, we define the *aggregate optimization gain* as the ratio of the total hop count before and after optimization. Formally,

$$gain, g = \frac{\sum_{k=1}^{K}\left|p_k^{del}\right|}{\sum_{k=1}^{K}\left|p_k^{del*}\right|}. \tag{4}$$

The gain, then, for a specific content request, req_k, is defined as $g_k = \frac{\left|p_k^{del}\right|}{\left|p_k^{del*}\right|}$. We note that any requests for the same content that originate from a common node will follow the same path towards the content (*i.e.*, $\left|p_k^{del}\right|$ before optimization and $\left|p_k^{del*}\right|$ after optimization).

Theorem 2: Given a graph, $G = (V, E, R)$, representing a ToR instance of the Internet AS topology, then the boundary of the gain for any request req_k, is $1.0 \leq g_k \leq \left|p_k^{del}\right|_{max}$.

Proof 5: The lower bound of g_k is 1.0 which is achieved when p_k^{del} cannot be optimized (*i.e.*, $p_k^{del} = p_k^{del*}$).

To obtain the upper bound of g_k, we need to construct a scenario when $\left|p_k^{del}\right|$ is at the maximum while $\left|p_k^{del*}\right|$ is at the minimum following the forwarding and optimization specifications previously described. We invoke Lemma 2 where $\left|p_k^{res}\right|_{max} = 1 + \frac{2|V|}{|V|-1}\sum_{n=2}^{|V|}\frac{1}{n} = \left|p_k^{del}\right|_{max}$ is a function of the topology size, $|V|$. On the other hand, $\left|p_k^{del*}\right|_{min} = 1$, *i.e.*, when the content record can be found a single hop away but only connected through a p2p link. Therefore, the upper bound is $g_k = \frac{\left|p_k^{del}\right|_{max}}{\left|p_k^{del*}\right|_{min}} = \left|p_k^{del}\right|_{max}$. □

Our investigation revealed that the gain is sensitive to the network structure and thus, a specific index describing the topology as a whole (*e.g.*, graph diameter, spectral radius, graph efficiency [31]) cannot estimate the gain. For example, two networks may have the same spectral radius but obtain different mean gain. The gain is specifically dependent on the valid paths allowed in the graph. Note that these are usually not the shortest paths (*e.g.*, those computed via Dijkstra's

[8] We use superscript asterisk (x^*) to indicate the state after optimization.

algorithm) since in our approach, the path constructions and optimization follow specific rules and restrictions.

To approximate the expected gain of our path optimization mechanism, we found that the crucial factor is the change in the distances from each node in the graph to the rest of the nodes when comparing the corresponding delivery paths before and after the optimization. To model this, we exploit the concept of closeness centrality, C_c, which is often used as a measure of efficiency of the network [32].

$$C_c(v_i) = \frac{|V|-1}{\sum_{j=1}^{|V|} dist(v_i, v_j)} \tag{5}$$

where $dist(v_i, v_j)$ is the distance in hop count from v_i to v_j. Unlike the convention, this distance is not computed as the shortest path, but rather, following our content delivery rules specified previously. We postulate that the expected gain can be approximated by considering the mean contribution of the change of closeness centrality of individual node after optimization. The modeled gain, g' is approximated as follows:

$$g' = \frac{1}{|V|} \sum_{i=1}^{|V|} \frac{C_c^*(v_i)}{C_c(v_i)} \tag{6}$$

where $C_c^*(v_i)$ is the closeness centrality of node v_i after optimization (*i.e.*, $C_c^*(v_i) = (|V|-1)/[\sum_{j=1}^{|V|} dist(v_i, v_j)^*]^{-1}$).

The union of the content delivery paths, $G_{\cup p^{del}}$ following the provider-route forwarding rule is essentially a spanning tree with no cycle. Thus, its average degree, $\overline{d} = \frac{1}{|V|}\sum_{i=1}^{|V|} d_i = 2 - \frac{2}{|V|}$ (*i.e.*, lower bound of a connected graph) and the number of links, $|E| = |V| - 1 = \underline{|E|}_{min}$. As such, the average degree of the optimized $G^*_{\cup p^{del}}$, $\overline{d^*}$, is always greater than $\overline{d}$. Increased connectivity may create new valid valley-free routes. Thus, $C_c^* \geq C_c$ since the closeness centrality is inversely proportional to the distance. This guarantees a positive gain with lower bound of 1.0 (*i.e.*, $g' \geq 1.0$); conforming to the lower bound of Theorem 2.

VI. Performance Evaluation

A. Simulation Setup

We evaluate the content access and delivery performance of CURLING using the Lord of the Links (LOTL) [33] dataset. From this, we extract various sub-topologies with a single root at AS-701 (Verizon) consisting of ToR graphs of sizes ranging between 200 to 1,600 ASs. We preserve all the p2c, c2p, and p2p links, but ignore the rarely-occuring s2s links.

We complement our evaluations using real data with results using mURT. As mentioned, recent research has indicated that the number of peering links in the Internet has significantly increased over the years and continues to do so [26][27]. Therefore, we retain control of the level of peering links in each mURT as a tuning knob to evaluate the modeled optimization gain. In our experiments, we vary the ratio of p2p to p2c links from 0.1 to 0.5, for a single 200-node mURT. For each simulation run, we set 10,000 unique content items uniformly distributed across the topology (*i.e.*, content is not replicated). Thereafter, 10,000 content requests were simulated for each run with the content popularity following the commonly accepted Zipf distribution with $\alpha = 1.0$.

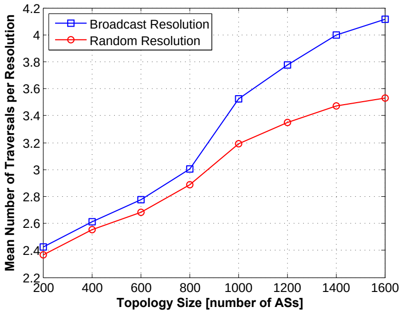


Fig. 5. Mean number of AS-traversals per content-resolution.

B. Simulation Results

We begin by showing the mean number of AS-traversals per resolution for topologies of different sizes in Fig. 5. We observe that the mean number of AS-traversals under the broadcast resolution mode is consistently higher than the random resolution mode, by an average of 8.4%. The difference in mean traversals between them increases with topology size – at 1,600 nodes, the increase of the broadcast scheme over the random scheme is as much as 16.5%.

While the broadcast resolution mode exhibits more AS-traversals, it yields shorter delivery paths. Fig. 6 compares the mean hop-counts of the unoptimized delivery path (the resolution path) with the optimized delivery path, for both the broadcast and random modes, for (a) all simulated content resolution paths, and (b) only content resolution paths for which a shorter delivery path exists. From Fig. 6(a), on average, the broadcast mode consistently finds shorter content delivery paths across all topologies. For the 1,600-AS topology, the unoptimized delivery path length of the broadcast mode is on average 6.1% lower than that of the random mode. The percentage decrease in optimized delivery path length is 4.4%, showing that after the optimization, the resulting path length under both modes converges to a small extent.

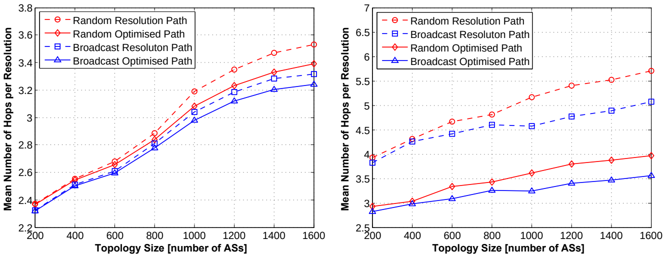
Fig. 6(b) shows that the optimized delivery path length of each resolution scheme is on average 29.0% hops less than the respective unoptimized path length. Comparing both modes, we observe again that the broadcast mode exhibits shorter delivery paths than the random mode, by an average of 7.4% and 7.3% for unoptimized and optimized paths respectively.

Fig. 7 shows the cumulative density function of the hop count of the 1,600-AS topology for the four aforementioned mode combinations. From this figure, we observe that for the broadcast mode, 90% of path lengths are less than 4.7-hops in the unoptimized case, and less than 4.4-hops in the optimized case. These values are less than those for the random mode, in which 90% of path lengths are less than 5.2-hops in the unoptimized case, and less than 4.8-hops in the optimized case.

We model the content resolution latency, τ_k^{crl}, as

$$\tau_k^{crl} = 2\tau_{inter} p_k^{res} + \tau_{lu}(p_k^{res}+1) + 2\tau_{intra} \sum_{a=1}^{p_k^{res}+1} (1 + \lfloor \log d_a \rfloor) \tag{7}$$

where τ_{inter} and τ_{intra} are the inter- and intra-AS-hop propagation delays, τ_{lu} is the CRC content table look-up time and d_a is the degree of domain a (*i.e.*, the number of direct neighbours of domain a). The term $1 + \lfloor \log d_a \rfloor$ is the mean number of



(a) (b)

Fig. 6. Mean number of hops per resolution for different topology sizes for (a) *all* content resolution paths; (b) *optimizable* content resolution paths.

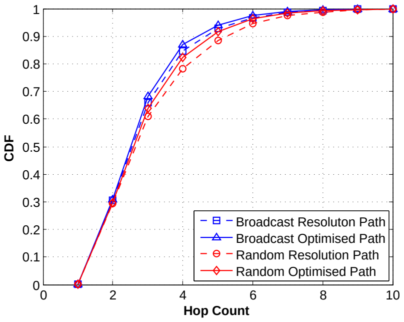
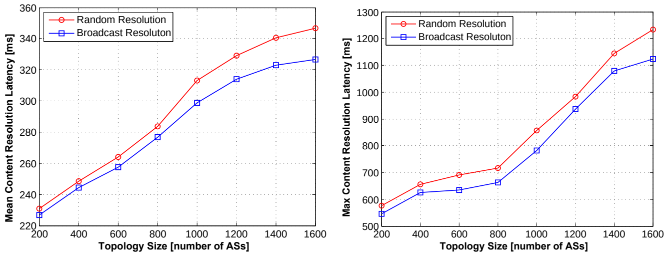


Fig. 7. CDF of hop-count for a 1,600-node topology.



(a) (b)

Fig. 8. Content resolution latency: (a) mean latencies, (b) maximum latencies.

TABLE I
NUMBER OF 96 GB RAM SERVERS REQUIRED TO HOLD CONTENT RESOLUTION STATES IN RAM [36].

	CURLING		DONA	
Type	Average	Median	Average	Median
Tier-1	2621	2703	4375	4375
Large ISP	121	14	1606	1868
Small ISP	2	1	683	5
Stub	1	1	90	1

intra-domain hops between CaRs in domain a [34]. The values of τ_{inter} and τ_{intra} are assumed to be constant, with values of 34ms and 2ms, respectively [34]. τ_{lu} is set to 5ms.

Fig. 8 shows plots of (a) the mean and (b) the maximum content resolution latency for the various topology sizes. Both the mean and maximum resolution latencies of the broadcast mode are lower than those of the random mode across all topology sizes, by approximately 3.5% and 6.8%, respectively. This follows from the fact that delivery paths of the broadcast mode are shorter than those of the random mode. In absolute terms, we observe the respective mean and maximum content resolution latency for a 1,600-AS topology to be 346ms and 1234ms for the random mode, and 326ms and 1122ms for the broadcast mode which are well within the bounds of tolerance typically reported in the literature [35].

C. Scalability

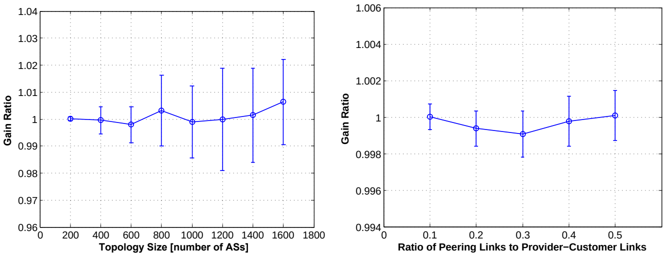
Adopting an information-centric approach, the CURLING design faces significant scalability challenges related to the support of the enormous information namespace, especially the support for content resolution at a global scale results in huge amount of content resolution state at the control plane. We pay particular attention to this issue. In [36], we showed that the design decision of CURLING to not exchange content resolution state over peering links has a dramatic reduction effect on the resulting resource requirements. We illustrate this in Table I, which shows the number of 96GB RAM servers[9] required to maintain content resolution state at each AS tier, for a total global volume of 10^{13} content items, considering 42-byte sized entries [36]. Focusing on the global scalability of CURLING, here we refer to the entire inter-domain topology. To this end, we adopt the AS classification used in [36], so as to characterize the different types of ASs. This classification is based on the customer cone size. Stub networks have no more than 4 customer networks, Small ISPs have a cone size between 5 and 50, Tier-1 ASs are the highest level of the inter-domain hierarchy that do not act as customers for another AS, and Large ISPs have a larger cone than small ISPs but are not Tier-1 members. The table also compares the number of servers required for the case of DONA. Considering that only a small subset of ASs in the Internet belongs to the Tier-1 and large ISP categories, it follows that CURLING requires the deployment of only a few small size data centers across the inter-domain topology, and only limited resources in the rest of the Internet, for the realization of its control plane.

At Internet-wide scale, this translates to a requirement for approximately 159K servers, against an approximate of 6.4M servers for DONA. It is worth noting that the current DNS-based resolution employs more than 32M DNS resolvers [37]. As shown in our previous work [38], these scalability features of CURLING, come at the cost of an average 16% increase of inter-domain resolution path lengths and a corresponding 2.78-fold increase of lookup processing overheads, against DONA. However, still the expected resolution latencies (see Fig. 8(a)) are in the same order of magnitude with DNS resolution latencies [37], even though, contrary to current DNS operation, our analysis does not include caching of resolution states.

D. Model Validation

We validate the modeled path optimization gain against the simulated gain (with 99% confidence interval) by computing the ratio of the two gains as $gain\ ratio = g/g'$. Thus, the best match is $g/g' = 1.0$ (*i.e.*, both the modeled and simulated gain are equal). Fig. 9(a) shows the gain ratio for topology size of 200 to 1,600 ASs. All modeled gains closely agree with the simulated gain (*i.e.*, all $g/g' \approx 1.0$). The worst case is just 0.64% off the best case. We also observe that topology size does not affect our model. In the previous section, we already showed that there is higher probability that there exists a shorter path than the basic delivery path originally constructed via the provider-route forwarding rule in larger topologies. The

[9] This assumption is conservative on purpose as such amount of memory is already provided in virtual machine instances from major cloud operators (*e.g.*, https://aws.amazon.com/ec2/instance-types/) *i.e.*, physical servers can be equipped with more memory resulting in much smaller data centers.



(a) (b)

Fig. 9. The gain ratio obtained in (a) real Internet sub-topology and (b) mURTs with different number of peering links.

main factor contributing to this is the increased number of p2p links. From the figure, we can also see that our model (*i.e.*, Eq. 6) can predict the achieved gain well within the confidence interval for different topology sizes. This is also true even for the 200-node topology where the interval is very small.

We further exploit the mURTs to validate our model against increasing number of peering links in the topology. We show in Fig. 9(b) for a 200-node topology by using the ratio of p2p-to-p2c links as a tuning knob. Increasing the ratio of p2p-to-p2c links lead to increased optimization gain. The figure verifies that our model provides reliable and accurate gain prediction regardless of the number of peering links. The worst case is just 0.09% off the ideal match.

VII. Related Work

In this section, we review and compare main ICN approaches against CURLING. Much of ICN research is centred around four main approaches: NDN [7], DONA [13], PSIRP [39], and NetInf [40]. CCN/NDN, DONA and PSIRP are radical approaches that were initially designed to replace IP, while NetInf was designed to be IP-compatible. Regarding this aspect, CURLING is by design evolutionary, promoting backward compatibility with IP-based network. Recent CCN/NDN developments have yielded approaches for supporting ICN functionalities using IPv4 or even Ethernet encapsulation [41]. Other efforts have led to the establishment of NDN overlay testbeds [42].

Apart from DONA, which also considered inter-domain setting, these main solutions mostly focused on intra-domain functionalities, with limited consideration on Internet-wide scale implications. For information-centricity at such scale, name resolution is of paramount importance, not only in terms of scalability, but also regarding inter-domain relationships in the resolution process and the resulting impact on the data forwarding plane. CURLING focuses exactly on these aspects. CURLING brings both a scalable [38] and flexible name resolution system, supporting intelligent content discovery (rather than hosts) at a global scale with significant impact on traffic steering capabilities in the data plane (*i.e.*, scoping and filtering). As such, CURLING takes the first steps towards an information-centric, Internet-wide management and control plane with minimal requirements on the underlying data plane.

In NDN, *hierarchical* naming is used. Resolution messages are directed along the interface with the longest-matched name-prefix. DONA uses *flat* content naming; if the routing table has no record of the requested content ID, the content request is forwarded by a Route Handler (RH) node present in each AS to a provider or peer AS, thus adhering to BGP domain routing policies. CURLING uses a similar resolution approach to DONA when it comes to BGP routing, but forwards content requests only to provider ASs and includes more advanced resolution and delivery features, such as scoping and filtering, that can lead to shorter and targeted delivery paths. In PSIRP, clients resolve content names by contacting a *rendezvous point* which replies with a Bloom filter message containing all the links to reach the destination from the requester, effectively through source routing. Finally, NetInf uses hierarchical distributed hash table (DHT) for resolving and transferring data. A recent study [36] investigated the feasibility of using Bloom filters for ICN-oriented name resolution to improve scalability of the system and found that configuring the Bloom filter to suit all domains in the Internet is impossible. CURLING does not rely on such enhancement. Our design, from the onset, considers this scalability issue by minimizing the necessary information propagation (*e.g.*, as compared with DONA [38]). It can be further extended to consider context-awareness as described in [43].

Several works have built on the intelligence and flexibility brought by SDN to support ICN (*e.g.*, dynamic definition of flexible header field matching rules). These efforts focused on enabling ICN routing and forwarding mechanisms on an intra-domain level, targeting the configuration of SDN-enabled forwarding devices. The early efforts [44][45] employ OpenFlow extensions to map between name-based routing and forwarding and host-based domains in an overlay fashion. A similar approach in [46] integrates intra-domain name resolution with packet header re-writing at the SDN controller. The solution proposed in [47] adds a CCNx daemon and wrapper function in legacy SDN switches to translate between CCN and TCP/IP forwarding primitives, requiring substantial changes in the forwarding fabric. Backwards compatibility is preserved in [48], where an overlay approach is proposed, building on a separate ICN control plane handling name resolution, routing and eventually forwarding configurations on the underlay. Reed et al. [49] took a different approach in enabling ICN-inspired intra-domain source routing in legacy SDN/OpenFlow domains. In contrast to all these, CURLING builds on the SDN paradigm at an *inter-domain* level. To the best of our knowledge, this is the first effort to couple information-centricity with the management flexibility of SDN at an Internet-scale, enabling a series of advantageous features such as path optimization, scoping and filtering.

VIII. Summary and Conclusions

We presented *CURLING*, an ICN-based distributed control system for efficiently delivering multimedia content and services at Internet scale. It supports inter-domain content multicasting while following the SDN principle whereby the content control-plane and the content data-plane are decoupled. This approach can be deployed over the current IP-based Internet with the only change to the networking infrastructure being the deployment of, still IP-based, content-aware routers at domain boundaries. CURLING offers highly flexible content management by individual stakeholders in the Internet marketplace. Specifically, content publishers and consumers may apply unified scoping and filtering modes to express their content access preferences on specific locations in the Internet. The underlying ISPs can also control content sessions based on their own policies and detected opportunities based on the underlying BGP routes (*e.g.*, *broadcast* vs. *random* mode and post-resolution path optimizations). Through our modeling, we proved that the proposed content resolution based on ISP business relationships is always able to locate the content source if it exists. In addition, AS-level content delivery paths, which are the results of both initial content resolution and path optimization, are guaranteed to be valley-free, conforming to the fundamental Internet inter-domain routing principles.

To comprehensively evaluate the performance of CURLING, we conducted our simulation experiments based on a subset of the real Internet topology rooted at one typical tier-1 ISP. Our key observation is that, even the initial 'blind' (*i.e.*, without knowledge of the location of the targeted content source) content resolution, the resulting content delivery path at the AS-level is still very short. Such results are consistent with the common observation of short AS-path length in BGP routing. Furthermore, follow-up path optimization based on multihoming and peering routes may further improve the path length for content delivery. Analytical modeling based on the modified URT topology also supports the accuracy of our simulation results. In addition, we found that CURLING is directly comparable to the current DNS in terms of infrastructure requirements and resolution delays.

References

[1] A. Antonopoulos *et al.*, "Shedding light on the Internet: Stakeholders and network neutrality," *IEEE Commun. Mag.*, vol. 55, no. 7, pp. 216–223, 2017.

[2] A. Beben *et al.*, "Multi-criteria decision algorithms for efficient content delivery in content networks," *Ann. of Telecommun.*, vol. 68, no. 3-4, pp. 153–165, 2013.

[3] B. Ahlgren *et al.*, "A survey of information-centric networking," *IEEE Commun. Mag.*, vol. 50, no. 7, pp. 26–36, 2012.

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

[5] C. Diot *et al.*, "Deployment issues for the IP multicast service and architecture," *IEEE Network*, no. 1, pp. 78–88, January 2000.

[6] G. Xylomenos *et al.*, "A survey of information-centric networking research," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 2, pp. 1024–1049, 2014.

[7] V. Jacobson *et al.*, "Networking named content," in *ACM Int'l Conf. on Emerging Netw. Experiments and Technologies, (CoNEXT)*, 2009.

[8] D. Perino and M. Varvello, "A reality check for content centric networking," in *ACM Workshop on Information-centric Networking*, 2011.

[9] C. Dannewitz, M. D'Ambrosio, and V. Vercellone, "Hierarchical DHT-based name resolution for information-centric networks," *Elsevier Comput. Commun.*, vol. 36, no. 7, pp. 736–749, 2013.

[10] K. V. Katsaros *et al.*, "On inter-domain name resolution for information-centric networks," in *IFIP Conf. on Netw.*, 2012, pp. 13–26.

[11] N. Fotiou *et al.*, "Developing information networking further: From PSIRP to PURSUIT," in *Broadband Communications, Networks, and Systems*. Springer Berlin Heidelberg, 2012, pp. 1–13.

[12] D. Kreutz *et al.*, "Software-defined networking: A comprehensive survey," *Proc. IEEE*, vol. 103, no. 1, pp. 14–76, 2015.

[13] T. Koponen *et al.*, "A data-oriented (and beyond) network architecture," *SIGCOMM Comput. Commun. Rev.*, vol. 37, no. 4, pp. 181–192, 2007.

[14] Cisco, "Mobile video delivery with hybrid ICN - IP-integrated ICN solution for 5G," White Paper, 2017.

[15] W. K. Chai *et al.*, "CURLING: Content-ubiquitous resolution and delivery infrastructure for next-generation services," *IEEE Commun. Mag.*, vol. 49, no. 3, pp. 112–120, 2011.

[16] T. Bates, E. Chen, and R. Chandra, "BGP route reflection: An alternative to full mesh internal BGP (IBGP)," RFC 4456, IETF, 2006.

[17] W. Augustyn and Y. Serbest, "Service requirements for layer 2 provider-provisioned virtual private networks," RFC 4665, IETF, 2006.

[18] J. Vasseur and J. L. Roux, "Path computation element (PCE) communication protocol (PCEP)," RFC 5440, IETF, Mar. 2009.

[19] L. Gao, "On inferring autonomous system relationships in the internet," *IEEE/ACM Trans. Netw.*, vol. 9, no. 6, pp. 733–745, Dec. 2001.

[20] L. Subramanian *et al.*, "Characterizing the Internet hierarchy from multiple vantage points," in *IEEE INFOCOM*, 2002.

[21] Z. Ge *et al.*, "Hierarchical structure of the logical Internet graph," in *Int'l Symp. Convergence of IT & Communications*, 2001.

[22] V. Sourlas *et al.*, "Distributed cache management in information-centric networks," *IEEE Trans. Netw. Serv. Manage.*, vol. 10, no. 3, pp. 286–299, September 2013.

[23] Y. Li *et al.*, "How much to coordinate? Optimizing in-network caching in content-centric networks," *IEEE Trans. Netw. Serv. Manage.*, vol. 12, no. 3, pp. 420–434, September 2015.

[24] X. Dimitropoulos *et al.*, "AS relationships: Inference and validation," *SIGCOMM Comput. Commun. Rev.*, vol. 37, no. 1, pp. 29–40, Jan. 2007.

[25] G. Di Battista *et al.*, "Computing the types of the relationships between autonomous systems," *IEEE/ACM Trans. Netw.*, vol. 15, no. 2, 2007.

[26] A. Dhamdhere and C. Dovrolis, "The Internet is flat: modeling the transition from a transit hierarchy to a peering mesh," in *ACM Int. Conf. Emerging Netw. Experiments and Technologies*, 2010.

[27] C. Labovitz *et al.*, "Internet inter-domain traffic," *SIGCOMM Comput. Commun. Rev.*, vol. 40, no. 4, pp. 75–86, Aug. 2010.

[28] P. Van Mieghem, *Performance Analysis of Complex Networks and Systems*. Cambridge University Press, 2014.

[29] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks," *Nature*, vol. 393, no. 6684, pp. 440–442, Jun. 1998.

[30] F. Wang and L. Gao, "On inferring and characterizing Internet routing policies," in *ACM Internet Measurement Conf. (IMC)*, 2003.

[31] V. Latora and M. Marchiori, "Efficient behavior of small-world networks," *Phys. Rev. Lett.*, vol. 87, pp. 198 701–1–198 701–4, Oct. 2001.

[32] S. Wasserman and K. Faust, *Social Network Analysis: Methods and Applications*. Cambridge University Press, 1994.

[33] Y. He *et al.*, "Lord of the Links: A framework for discovering missing links in the Internet topology," *IEEE/ACM Trans. Netw.*, vol. 17, no. 2, pp. 391–404, 2009.

[34] J. Rajahalme *et al.*, "On name-based inter-domain routing," *Comput. Netw.*, vol. 55, no. 4, pp. 975–986, Mar. 2011.

[35] F. Nah, "A study on tolerable waiting time: How long are web users willing to wait?" *Behaviour & Inform. Technol.*, May 2004.

[36] K. V. Katsaros, W. K. Chai, and G. Pavlou, "Bloom filter based inter-domain name resolution: A feasibility study," in *ACM Conf. on Information-Centric Netw. (ICN)*, Oct 2015.

[37] K. Schomp *et al.*, "On Measuring the Client-side DNS Infrastructure," in *Proc. Conf. on Internet Measurement Conference (IMC)*, 2013.

[38] K. Katsaros *et al.*, "On the inter-domain scalability of route-by-name information-centric network architectures," in *IFIP Conf. on Netw.*, 2015.

[39] P. Jokela *et al.*, "LIPSIN: line speed publish/subscribe inter-networking," *SIGCOMM Comput. Commun. Rev.*, vol. 39, no. 4, pp. 195–206, 2009.

[40] B. Ahlgren *et al.*, "Second NetInf architecture description," 4WARD EU FP7 Project, Deliverable D-6.2 v2.0 FP7-ICT-2007-1-216041, apr 2010.

[41] "CICN Project," https://wiki.fd.io/view/cicn, accessed: 2017-06-01.

[42] "NDN," https://named-data.net/ndn-testbed/, accessed: 2018-01-15.

[43] G. Pavlou *et al.*, "Internet-scale content mediation in information-centric networks," *Ann. of Telecommun.*, vol. 68, no. 3-4, pp. 167–177, 2013.

[44] N. B. Melazzi *et al.*, "An openflow-based testbed for information centric networking," in *Future Network Mobile Summit*, July 2012, pp. 1–9.

[45] S. Salsano *et al.*, "Information centric networking over SDN and OpenFlow: Architectural aspects and experiments on the OFELIA testbed," *Comput. Netw.*, vol. 57, no. 16, pp. 3207–3221, Nov. 2013.

[46] M. Vahlenkamp *et al.*, "Enabling ICN in IP networks using SDN," in *IEEE Int. Conf. on Netw. Protocols (ICNP)*, 2013.

[47] X. N. Nguyen, D. Saucez, and T. Turletti, "Providing CCN functionalities over OpenFlow switches," Research Report, Aug. 2013. [Online]. Available: https://hal.inria.fr/hal-00920554

[48] N. van Adrichem and F. Kuipers, "NDNFlow: Software-defined named data networking," in *IEEE Conf. Network Softwarization*, 2015.

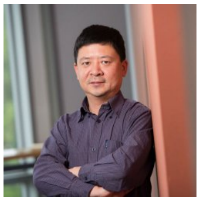
[49] M. J. Reed *et al.*, "Stateless multicast switching in software defined networks," in *IEEE Int'l Conf. on Commun. (ICC)*, May 2016.

Wei Koong Chai received both M.Sc. (Distinction) and Ph.D. degrees from University of Surrey, UK in 2002 and 2008 respectively. He currently heads the Future and Complex Networks Research Group (FlexNet) in the Department of Computing and Informatics in Bournemouth University, UK. He is also currently a Honorary Senior Research Associate at University College London (UCL). His current research interests include information-centric networking, network science, communication in cyber physical systems and resource management.

George Pavlou is Professor of Communication Networks in the Department of Electronic and Electrical Engineering, University College London, UK, since 2008, where he coordinates research activities in networking and network/service management. He holds a Diploma in Engineering from the National Technical University of Athens, Greece, and MSc and PhD degrees in Computer Science from University College London. His research interests include aspects such as network resource management, traffic engineering, quality of service, autonomic networking, network programmability and content-based networking. He has been instrumental in a number of collaborative research projects that produced significant results with real-world uptake and has contributed to standardisation activities in ISO, ITU-T and the IETF. In 2011 he received the IEEE/IFIP Dan Stokesbury bi-annual award for "distinguished technical contributions to the growth of the network management field" while in 2017 he was elected Fellow of the IEEE "for contributions to network resource management and content-based networking".

George Kamel is a software developer at the 5G Innovation Centre (5GIC), Institute for Communication Systems, University of Surrey, UK. He received his M.Eng. (honours) in telecommunications engineering and his Ph.D. in mobile communications from King's College London, UK in 2005 and 2010, respectively. He is currently working on the development of the "5GIC Exchange" system to enable the interconnection of and brokerage between different 5G test beds, and the deployment of 5G slices across them using NFV and SDN technologies. He is also involved in the design and development of packages for ETSI Management and Orchestration (MANO) to automate the creation and management of 5G slices. He is currently involved in various EU and government-funded projects related to the research and development of 5G networks. From 2011 to 2017, George was a research fellow at the 5GIC, and was involved in various EU, EPSRC, and industry-funded projects related to information-centric networking and context-aware network management.

Konstantinos V. Katsaros , Ph.D. is a Senior R&D Engineer at Intracom S.A. Telecom Solutions, Greece, active in the areas of network function virtualization and software-defined networking for 5G networks (CHARISMA, VirtuWind, 5G-VINNI 5G-PPP/H2020 R&D Projects) and information-centric networking (H2020 POINT), where he holds extensive experience (H2020 UMobile, FP7 PSIRP and PURSUIT, UK EPSRC COMIT). He has also worked on smart grids (FP7 C-DAX), inter-domain carrier services (FP7 ETICS) and multicast/broadcast service provision over cellular networks (FP6 B-BONE). He has been a Research Associate at University College London (UK), working on smart grid communications and information-centric networking, and Telecom ParisTech (France), working on cloud networking. He holds a BSc (2003) in Informatics, a MSc (2005) in Computer Science and a PhD (2010) in Computer Science, with a particular focus on information-centric networking, content distribution and mobility support, from AUEB (Greece). Dr. Katsaros has published and presented his work in numerous academic conferences and workshops, as well as scientific journals, and has received scholarships and awards from Greek institutions and international companies.



Ning Wang is a Professor at the 5G Innovation Centre, Institute for Communication Systems, University of Surrey, UK. He received his B.Eng (Honours) degree from the Changchun University of Science and Technology, P.R. China in 1996, his M.Eng degree from Nanyang University, Singapore in 2000, and his PhD degree from the University of Surrey in 2004 respectively. His research interests mainly include Future network design, Information-Centric networking (ICN), mobile content delivery, context-aware networking and Quality of Services/Experiences (QoS/QoE).