CURLING: CONTENT-UBIQUITOUS RESOLUTION AND DELIVERY INFRASTRUCTURE FOR NEXT GENERATION SERVICES

Wei Koong Chai[1], Ning Wang[2], Ioannis Psaras[1], George Pavlou[1], Chaojiong Wang[2], Gerardo García de Blas[3], Francisco Javier Ramon Salguero[3], Lei Liang[2], Spiros Spirou[4], Andrzej Beben[5] and Eleftheria Hadjioannou[6]

[1]University College London
[2]University of Surrey
[3]Telefónica Investigación y Desarrollo S.A.U.
[4]Intracom SA Telecom Solutions
[5]Warsaw University of Technology
[6]PrimeTel PLC

Email: wchai@ee.ucl.ac.uk, n.wang@surrey.ac.uk, i.psaras@ee.ucl.ac.uk, g.pavlou@ee.ucl.ac.uk, c.wang@surrey.ac.uk, ggdb@tid.es, fjrs@tid.es, l.liang@surrey.ac.uk, spis@intracom.com, abeben@tele.pw.edu.pl, eleftheriah@prime-tel.com

ABSTRACT

The next generation Internet is expected to focus more on large-scale media/content distribution rather than the communication infrastructure. In this article, we present CURLING, a Content-Ubiquitous Resolution and Delivery Infrastructure for Next Generation Services. The proposed architecture will support the realization of a future content-centric Internet that will overcome the current intrinsic constraints by efficiently diffusing media content of massive scale. We propose a holistic approach that natively supports content manipulation capabilities which encompass the entire content lifecycle, from content publication to content resolution and finally, to content delivery at Internet-wide scale. The CURLING infrastructure offers to both content providers and customers high flexibility in expressing their location preferences when publishing and requesting content respectively, thanks to the proposed *scoping* and *filtering* functions. Content manipulation operations can be driven by a variety of factors, including business relationships between Internet Service Providers (ISPs), local ISP policies, and specific content provider and customer preferences. Content resolution is also natively coupled with optimized content routing techniques that enable efficient unicast and multicast-based content delivery across the global Internet.

1.0 INTRODUCTION

The original Internet model focused mainly on *connecting machines* whereby addresses point to physical end-hosts and routing protocols compute routes to specific destination endpoints. Nowadays the Internet is primarily used for *transporting content/media,* where a high volume of both user-generated and professional digital content, e.g., webpages, movies/songs, live video streams, etc., is delivered to users who are usually only interested in the content itself rather than the location of the content sources. Human needs along with the nature of communication technologies have transformed the Internet into a new content marketplace generating revenue for various stakeholders. In fact, the Internet is rapidly becoming a super-highway for massive digital content dissemination.

In this context, many researchers have advocated a transition of the Internet model from *host-centric* to *content-centric*, with various different architectural designs proposed [1][2][3][4][5][6][7]. Many of these proposals support the key feature of *location independence,* where content consumers do not need to obtain explicit location information (e.g., the IP address) of the targeted content source *a priori,* before issuing the consumption request [1][2][3][5][7]. Nevertheless, location requirements are still demanded by both content consumers and providers. On the one hand, content providers may want their content accessed only by content consumers from a specific region (which is known as *geo-blocking*), for example BBC iPlayer, Amazon Video-on-Demand, Apple iTunes Store and Sina video services. On the other hand, content consumers may prefer content originated from specific regions in the Internet, for instance, a US-based shopper might only like to check the price of an item sold in Amazon online stores in North America rather than anywhere else in the world. Today, this is typically achieved through the user's explicit input in the URL (e.g., Amazon.com and Amazon.ca), and supported by name resolution through the standard Domain Name System (DNS) [8], with the relevant requests directed towards the specific regional web server. Similar practice can be observed in multimedia-based content access (e.g., in video on demand services), where consumers have specific requirements regarding the location/area of content sources.

In this article, we introduce a new Internet-based content manipulation infrastructure – CURLING: Content-Ubiquitous Resolution and Delivery Infrastructure for Next Generation Services. The objective is to both accurately and efficiently "hit" (or "not hit") content objects in *specific regions/areas of the Internet*, based on specific user requirements and preferences. Such a paradigm, deployed by ISPs, will allow both content providers and consumers to express their location requirements when publishing/requesting content, thanks to the embedded content *scoping/filtering* functions. In particular, instead of following the conventional DNS-like approach, where a content URL is translated into an explicit IP address pointing to the targeted content server, our proposed content resolution scheme is based on hop-by-hop "gossip"-like communication between dedicated Content Resolution Server (CRS) entities residing in individual ISP networks. Content resolution operations can be driven by a variety of factors, including the business relationships among ISPs (provider/customer/peer), content consumer preferences and local ISP policies. This resolution approach is natively coupled with content delivery processes (e.g., path setup), supporting both unicast and multicast functions. Specifically, a content consumer simply issues a single *content consumption request* message (capable of carrying his/her location preferences on the content source candidate(s)), and then individual CRS entities collaboratively resolve the content identifier in the request, in a *hop-by-hop manner,* towards the desired source. Upon receiving the content consumption request, the selected content source starts transmitting the requested content to the consumer. During this content resolution operation, "multicast-like" content states are installed along the resolution path so that the content flows back immediately upon the completion of the resolution process. This is in contrast to the current IP-based content delivery services where name resolution and content delivery are separate processes. By exploiting multicast delivery techniques, we increase the sustainability of the system in view of the expected explosion of content in the Internet.

2.0 Business Model

We envision an open business environment that involves stakeholders in the Internet marketplace, ranging from single individuals or small communities to large commercial service/infrastructure providers. We present here a basic business model that can be expanded to multiple more sophisticated ones that promote increased participation and competition. We give a brief overview on the involved stakeholders and their business interactions. The following top-level roles can be envisaged:

- *Content providers* (CPs): the entities that offer content to be accessed and consumed across the Internet. These include both commercial content providers and end users who want to publish their content in the Internet;
- *Content consumers* (CCs): the entities that consume content as receivers;
- *Internet service providers* (ISPs): equipped with the CURLING content-aware infrastructure, ISPs are responsible (1) for dealing with the content publication requests from content providers, and content consumption requests from consumers, and (2) for the actual delivery of the content, possibly with quality of service (QoS) awareness.

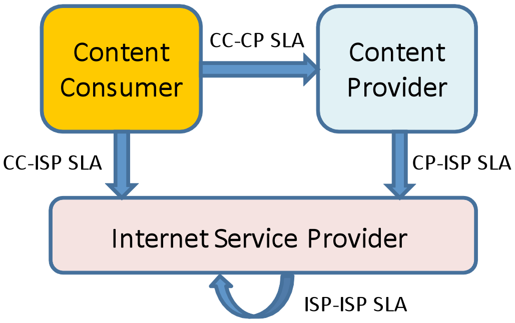

Figure 1: Business Model

Figure 1 illustrates the business model indicating the business interactions between individual roles. Since content providers rely on the underlying content-aware infrastructure owned by ISPs for having their content published across the Internet, they are expected to establish a service level agreement (SLA) involving relevant payment to the ISP for content publication services (CP-ISP SLA). In addition, since ISPs offer content searching/location and delivery services to content consumers, a CC-ISP SLA can be established. Sometimes, content consumers may need to pay content providers for consuming charged content (e.g., pay-per-view). This can be covered by the CC-CP SLA between the two. Finally, business contracts are also established between ISPs (ISP-ISP SLA), given a provider-customer or a peering relationship between them. As will be described later, a low-tier ISP needs to pay its provider ISP not only for content traffic delivery but also for "delegated" content publication/resolution services on behalf of its own customers, including directly attached content providers and consumers.

3.0 THE CURLING ARCHITECTURE

Our solution requires a form of aggregatable labels capable of being sequentially ordered to which we refer to as *content identifiers* (IDs). A content item to be published and accessed is allocated a globally unique content ID. Multiple copies of the same content which are physically stored at different sites in the Internet share one exclusive ID.

Content manipulation operations rely on two distinct entities in the CURLING architecture: (1) the Content Resolution Server (CRS) that primarily handles content publication requests, discovers the requested content and supports content delivery, and (2) the Content-aware Router (CaR) that collaborates with its local CRS(es) to enforce receiver-driven content delivery paths.

A CRS entity is present in every domain (1) for handling *local* publication requests and content consumption requests and (2) for interacting with other neighboring CRS entities for content publication/resolution across domains. Both content providers and consumers are configured to know their local CRS. The number of CRSes within each domain depends on performance and resilience considerations. Figure 2 depicts the functional view of the overall CURLING architecture; we explain the operational properties of each functional block below. The internal structure of the CRS entity consists of three logical components. The *content management* block is responsible for dealing with requests from both content providers and consumers (via CRS-CP and CRS-CC interfaces respectively), including content ID allocation and entry creation upon new content registrations, and also content ID lookup upon each content consumption request from a content consumer. A dedicated content record repository is also maintained, including not only content ID lookup information, but also ingress and egress(es) CaR(s) within the local domain for each active content session being delivered in the network. The *inter-CRS protocol component* enables the communication between neighboring CRS entities for handling inter-domain content publication/consumption requests. Finally, the *monitoring module* gathers necessary "near real-time" information on content server and underlying network conditions for supporting optimized content resolution and delivery configuration operations.

CRSes communicate with other entities via specialized interfaces as described below:

- Inter-CRS interface – it enables interaction amongst CRSes in neighboring domains especially when they cooperate in content publication and searching for a requested content across domains.

- CRS-CP interface – it connects content servers owned by content providers with CRSes, and allows content providers to publish content, optionally with *scoping* (see next section) requirements on potential content consumers. This interface is also responsible for passing information on server load conditions to a CRS for enabling optimized content resolution operations.
- CRS-CC interface – it connects content consumer devices with the CRSes and allows consumers requesting and receiving content with scoping/filtering preferences on candidate content sources.
- CRS-CaR interface – This bi-directional interface allows a CRS to actively configure relevant CaRs for each content session (e.g., content state maintenance). It also gathers necessary information from the underlying network that will be used for optimized content resolution processes.

A CaR is the network element that is able to natively process content packets according to their IDs. In general, it is not necessary for every router in the network to be a CaR, and typically CaRs are planted at the network boundary as ingress and egress points for content delivery across ISP networks. The function of CaRs will be specified later with the description of content delivery process.

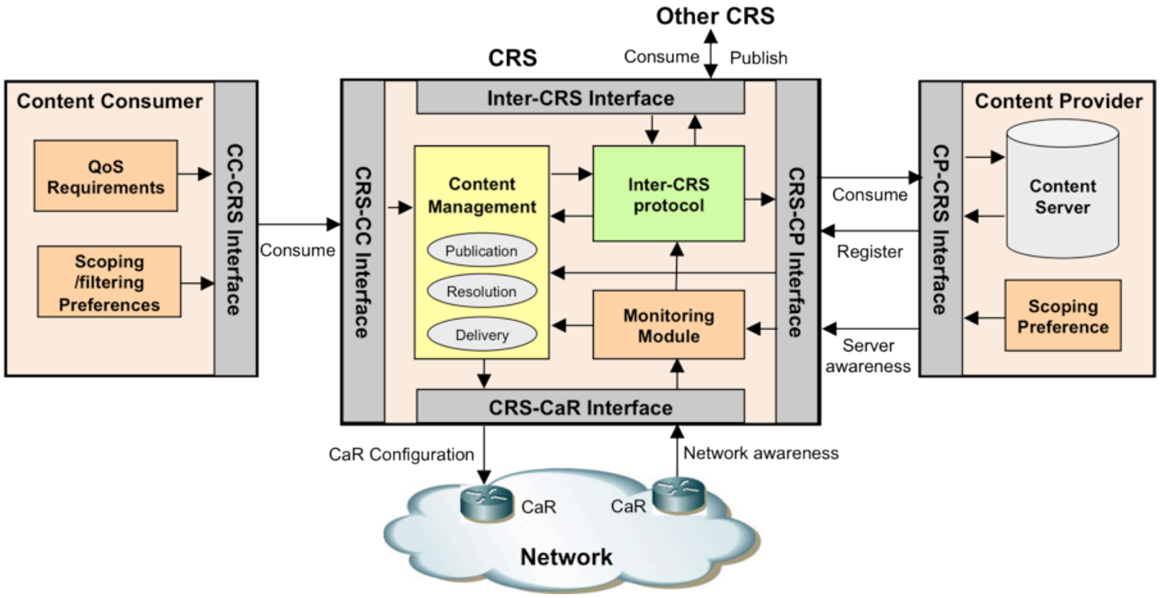

Figure 2: High-level architecture of the hop-by-hop hierarchical content resolution approach

4.0 Hop-by-hop Hierarchical Content-based Operations

Fundamentally, we envision the following three-stage content operation lifecycle: *publication*, *resolution* and *delivery*. The task of content resolution is to (1) identify the desired content source in the Internet according to the requested content ID and optionally content consumer preferences, and (2) actually trigger the content transmission by the selected content server. Once the content server starts the transmission of the content upon receiving the content consumption request, the content delivery function is responsible for enforcing the actual delivery path back to the consumer. Content publication and resolution can be operated based on various factors, including business relationships between domains (ISP-ISP SLA), inter-domain BGP routing configurations, local ISP policies and content provider/consumer preferences.

4.1 CONTENT PUBLICATION

Content publication is the process of making content available across the Internet. It consists of two stages.

Stage 1: Content Registration – It begins with the content provider notifying the local CRS that a new content is now available via a `Register` message. In the case where multiple copies of the same content are available at different locations, the content provider is responsible for informing the local CRS(es) of each content server hosting that specific content copy. Upon reception of the `Register` message, the CRS registers this content by creating a new record entry in its local content management repository containing (1) a globally unique content ID assigned to that content, and (2) the *explicit location* of the content (i.e. IP address of the content server).

Stage 2: Content Publication Dissemination – Once the content is registered to a CRS, this CRS is responsible for publishing it across the global Internet to ensure successful discovery of the content by potential consumers. This is achieved through the dissemination of the `Publish` message across CRSes in individual domains according to their business relationships. A `Publish` message is created by the CRS where the content is actually registered by the content provider. By default, each CRS disseminates a new `Publish` message towards its counterpart in the *provider domain(s)* until it reaches a tier-1 ISP network. Each CRS receiving a new `Publish` message updates its content management repository with a new record entry containing the content ID and the *implicit location* of the content (i.e. the IP prefix associated with the neighboring domain from where the `Publish` message has been forwarded). Following this rule, each CRS effectively knows the locations of all the content within its own domain (explicitly) and those under it (its customer domains, implicitly). Peer domains, however, will not know the content records of each other.

We introduce the concept of *scoped publication* to allow publication of content only to specific areas in the Internet as designated by the content provider. This feature is able to natively support regionally-restricted content broadcasting services such as BBC iPlayer and Amazon VoD that are only available within the UK and the US respectively. We achieve this through the `INCLUDE` option embedded in the `Register/Publish` messages where the content provider specifies a scoped area in the Internet, e.g., only the IP prefix associated with the local ISP network where the content is registered. A special case of *scoped publication* is the *wildcard mode* (denoted by asterisk "*" symbolizing *all* domains) for which the content provider has no restrictions on the geographical location of potential consumers in the Internet.

Figure 3 illustrates different scenarios in the publication process. It depicts the *domain-level* network topology with each circle logically representing a domain containing a CRS entity. We first assume that content provider **S1** registers a content item (assigned with ID **X1** by the local CRS in the stub domain **A.A.A)** to the entire Internet by issuing a `Register` message with a wildcard. Each intermediate CRS along the publication path creates a content entry for **X1** associated with the IP prefix of its customer domain from where the Publish message has been forwarded. For clarity, the Publish messages are omitted in the figure for other scenarios. Our approach also allows local domain policies to influence the publication process (e.g., domain **B.A** has the policy of NOT propagating content **X2** originated from the multi-homed domain **A.B.B** to its own provider). **S3** illustrates the *scoped* registration by only registering content **X3** to tier-2 domain **A.A** from this content provider. This effectively limits the access of content **X3** to domain **A.A** and its customer domain **A.A.A**. Finally, records for different copies of the same content can also be aggregated. For instance, both **S4** and **S5** host one copy of content **X4** respectively, but the two Publish messages from **B.B.A** and **B.B.B** are merged at **B.B**, in which case domain **B** only records aggregated location information (**X4→B.B**). A content consumption request for **X4** received at **B.B** can be forwarded to either **B.B.A** or **B.B.B** based on performance conditions such as content delivery path quality or server load, as will be discussed later.

The process of de-registering a content (e.g., triggered by the deletion of a content) follows the same procedure as the publication process whereby an Update message is forwarded according to the same rules detailed above.

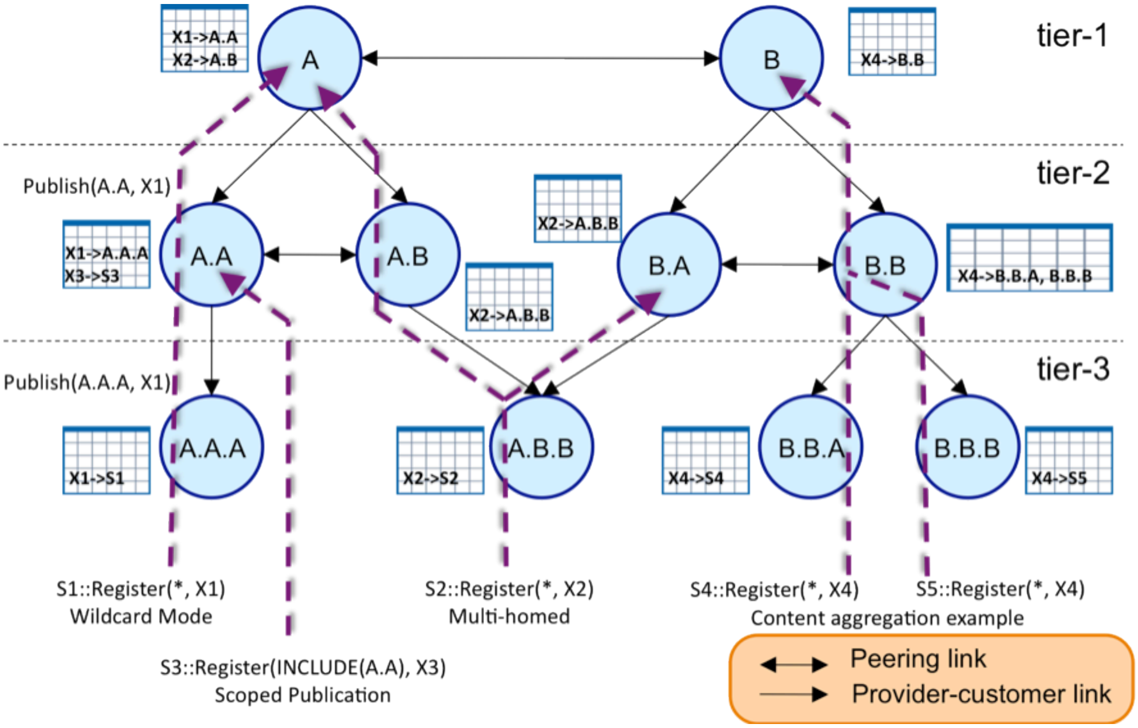

Figure 3: Content publication process.

4.2 CONTENT RESOLUTION

In the content resolution process, a content consumption request issued by a content consumer is resolved by discovering the location of the requested content and is finally delivered to the actual content source to trigger the content transmission. A content consumer initiates the resolution process via a Consume message containing the ID of the desired content. The primary resolution procedure follows the same "*provider route forwarding*" rule in the publication process (i.e. the Consume message will be further forwarded to its provider(s) if the CRS cannot find the content entry in its local repository). In case a tier-1 domain is not aware of the content location, then the request is forwarded to all its neighboring tier-1 domains until the content consumption request is delivered to the identified content source. If the content is not found after the entire resolution process, an Error message is returned to the requesting content consumer indicating a resolution failure.

We define two distinct content resolution stages:

- **Uphill** – the forwarding of a content consumption request from the local CRS "up" along the provider route until it reaches a domain whose CRS has the record entry for the requested content ID.
- **Downhill** – the forwarding of the content consumption request from the domain whose CRS has the record entry of the requested content ID "down" to the explicit content server that hosts the content.

Similar to the publication process, *scoping functions* can also be applied in the resolution process, either embedded in the request from a content consumer or actively issued by a CRS for route optimization purposes during the content delivery phase (see next section). Such a function allows a content consumer to indicate preferred ISP network(s) as the source domain of the requested content. Specifically, a content consumer may use the INCLUDE option in Consume messages, which carry one or multiple IP prefixes to indicate where he/she would like to receive the content[1] from. Since a set of explicit IP prefixes for candidate content source is carried in the Consume message, the corresponding resolution process becomes straightforward: each intermediate CRS only needs to forward the request (splitting required in the presence of multiple non-adjacent IP prefixes) towards the targeted IP prefix(es) directly according to the underlying BGP routes. In case multiple inter-domain routes are available towards a specific prefix, the most explicit one will be followed, as is consistent with today's inter-domain routing policy. In Figure 4, content consumer **C1** issued a Consume message for content **X1** indicating its preference for content source in domain **A**. This Consume message is then explicitly forwarded towards **A** from **B** following the underlying BGP routing, but without splitting it to **C** despite that a copy of **X1** is also accessible from **C**'s customer domain **C.A**. This scoping-based content resolution path is illustrated with the solid line in the figure.

[1] It is not always required that content consumers know the actual IP prefix of the domains they prefer but their local CRSes may be responsible for translating the "region information" (e.g. domain names) into IP prefixes through standard DNS services.

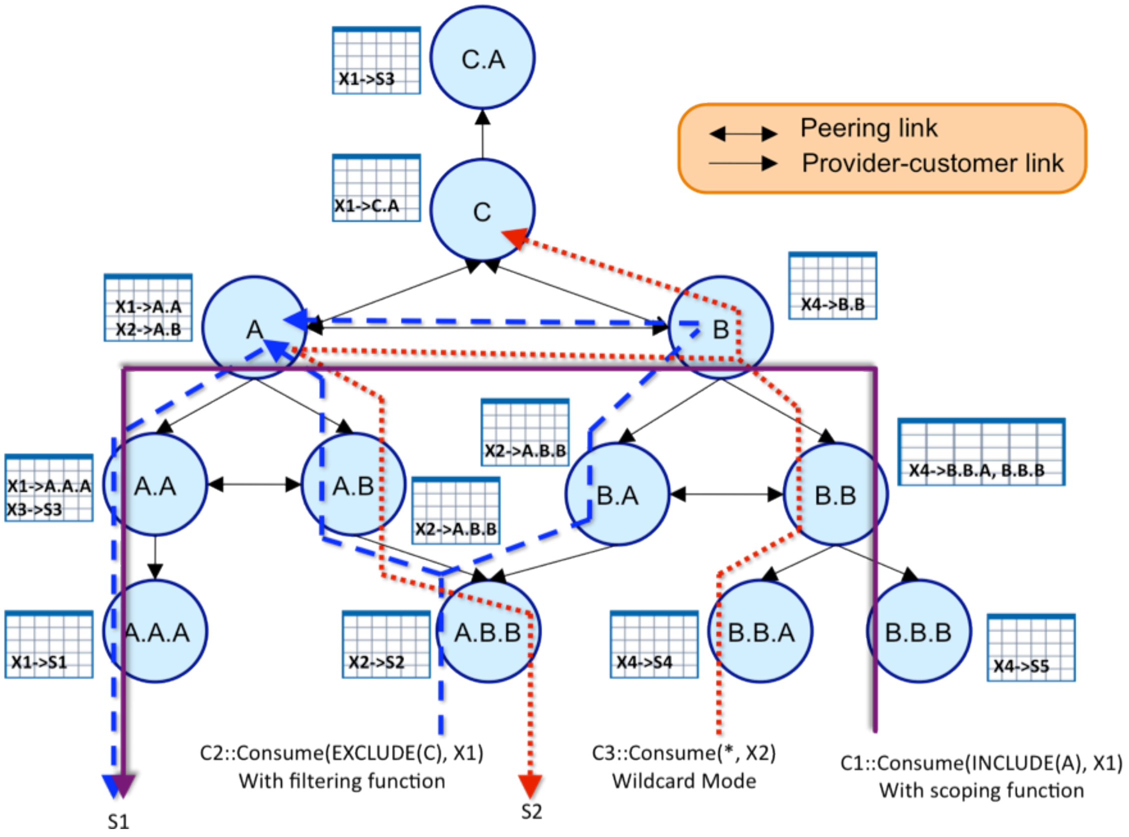

Figure 4: Content resolution in scoping, filtering and wildcard modes

The *filtering* function in content resolution operations has complementary effect to *scoping*. Instead of specifying the preferred networks, the content consumer has the opportunity to indicate unwanted domains as possible sources of the desired content. The filtering function is enabled via the `EXCLUDE` option in `Consume` messages. It is important to note the fundamental difference in resolving content consumption requests with *scoping* and *filtering* functions. In contrast to the *scoping* scenario in which a content consumption request is explicitly routed towards the desired IP prefix(es) according to the BGP route, in the *filtering* case, each request is routed based on the business relationship between domains (similar to content publication operations). Consider again Figure 4 with content consumer **C2** requesting **X1** with the exclusion of domain **C**. Since it is multi-homed, the request is sent to both its providers **A.B** and **B.A** (see the dashed line in the figure). However, at the tier-1 level, domain **C** is excluded when resolving this request even though a copy of content **X1** can be found in the customer domain of **C**.

A wildcard in a content consumption request can be regarded as a special case whereby the content consumer does not have preferences on the geographical location of the content source. The wildcard-based resolution is illustrated in Figure 4 via the request from consumer **C3** for content **X2** (dotted lines). We see that **B** splits the request to both **A** and **C** at the tier-1 level. Since only **A** has the record entry for **X2**, the request is resolved downhill to **S2**.

Through these illustrations, we show that *bi-directional location-independence* can be achieved in the sense that neither content consumers nor providers need to know *a priori* the explicit location of each other for content consuming. In particular, content consumers may include *implicit* content scoping/filtering information when requesting content. The content resolution system then automatically identifies the server in the desired "area" that hosts the content. On the content provider side, when a content is published, scoping can be applied such that the content can only be accessed by consumers in the designated area in the Internet. As we will show in the following section, thanks to the multicast-oriented content delivery mechanism, the content server is not aware of the explicit location of the active consumers of that content.

We conducted simulation experiments based on a real domain-level Internet sub-topology rooted at a Tier-1 ISP network (AS7018). This 4-tier network topology, extracted (with aggregation) from the CAIDA dataset [9], contains 5500 domains and 14714 inter-domain links, with explicit business relationship between neighboring nodes. Content sources and consumers were randomly distributed in the domains of this topology. According to our results, the average length of the combined uphill and downhill content resolution paths between individual content consumers and resolved content sources is 4.4 domain-level hops, i.e. the content is on average 4.4 ASes away according to the resolution paths. This is a very good result and also consistent with the general observation that Internet inter-domain sessions are of similar length based on the inter-domain BGP routing that is driven by the business relationships between ISP networks that follow the power-law Internet topology.

4.3 CONTENT DELIVERY

According to our design, content delivery paths are enforced in a receiver-driven multicast manner that needs state maintenance based on content IDs. As previously described, content consumption requests from consumers are resolved through a sequence of CRSes residing in individual domains 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 CRS has passed the content consumption request to its next hop counterpart in the neighboring domain, it needs to configure the local CaRs that will be involved in the delivery of content flows back from the potential server. Specifically, once a CRS receives a content consumption request from its counterpart in the previous hop domain and forwards it towards the next hop CRS, it needs to correspondingly install the content ID state at the local egress and ingress border CaRs connecting to the two neighboring domains[2]. The determination of ingress/egress CaRs for each content consumption request is purely based on the BGP reachability information across networks. Within each domain, the communication between the non-physically 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 [1]. As a result, the actual domain-level content delivery path is effectively the reverse path followed by the delivery of the original content consumption request. It is worth

[2] In case of a failed content resolution, content states temporally maintained at CaRs can be either timed-out or explicitly torn down by the local CRS.

mentioning that CRSes do not directly constitute the content delivery paths, in which case the configuration interaction between the CRS and local ingress/egress CaRs is necessary.

Let's take Figure 5 as an example for illustration. We assume that currently content consumer C1 (attached to domain 2.1/16) is consuming live streaming content X from server S (attached to domain 1.2.1/24). The content delivery path traverses a sequence of intermediate domains, and each of the corresponding ingress/egress CaRs is associated with a star that indicates the content state maintained for content delivery. As mentioned previously, such content states are configured by the local CRSes during the content resolution phase. Now content consumer C2 (attached domain 1.1/16) issues a consumption request for the same content. Upon receiving the content consumption request, the local CRS forwards it to its provider counterpart in domain 1/8 (uphill), as it is not aware of the content source location. Since the CRS in 1/8 knows that content flow for X is being injected into the local network via the originally configured ingress CaR 1.0.0.2, it then updates its outgoing next-hop CaR list by adding a new egress 1.0.0.3 leading towards content consumer C2. As a result a new branch is established from CaR 1.0.0.2 which is responsible for delivering the content back to the new consumer C2 (the dash line), but without any further content resolution process.

The proposed content delivery operation is also supported by a routing optimization technique for path switching from *provider* routes to *peering* routes if identified. In the figure, once the CRS in domain 1.1/16 has noticed that the content flow with source address belonging to prefix 1.2.1/24 has been injected into the local domain via ingress CaR 1.1.0.1 via the provider route, and it also knows from the local BGP routing information that there exists a peering route towards the content source, it then issues a new *scoping-based* content consumption request: `Consume(INCLUDE{1.2.1/24},X)` and sends it to the CRS in domain 1.2/16 in the peering route towards the source. Upon receiving the request, the CRS in 1.2/16 will update the local CaR 1.2.0.1 by adding a new outgoing next-hop CaR 1.1.0.1. As a result, a new branch via the peering route is established towards content consumer **C2**. Once the ingress CaR 1.1.0.1 has received the content via the interface connecting to 1.2.0.1, it will prune the old branch via the provider route (the dash line). The purpose of such content delivery path optimization across domains is to effectively reduce content traffic within top-tier ISP networks and also possibly reduce the content delivery cost for customer domains. Of course, this operation is not necessary if a CRS is allowed to send content consumption requests to its peering counterparts (in addition to the provider direction) during the resolution phase. However, such an option will incur unnecessarily higher communication overhead in disseminating content consumption requests, especially when the peering route does not lead to any source that holds the requested content.

We are also interested in the actual benefit from such inter-domain routing optimization techniques for cost-efficient content delivery across the Internet, especially from the view point of tier-1 ISPs that constitute the Internet core. We used the same domain-level topology as previously described for evaluating the corresponding performance. According to our results, the content traffic (in terms of the number of media sessions) traversing the root tier-1 ISP can be reduced by a substantial 29.1% through peering route switching. This is beneficial given that less traffic traverses tier-1 domains through relatively long paths. Correspondingly, the content traffic in tier-2 ISP networks increases by 5.3%, and the overall increase of content traffic carried by lower tiers (Tier-3 and Tier 4) is 25.8%.

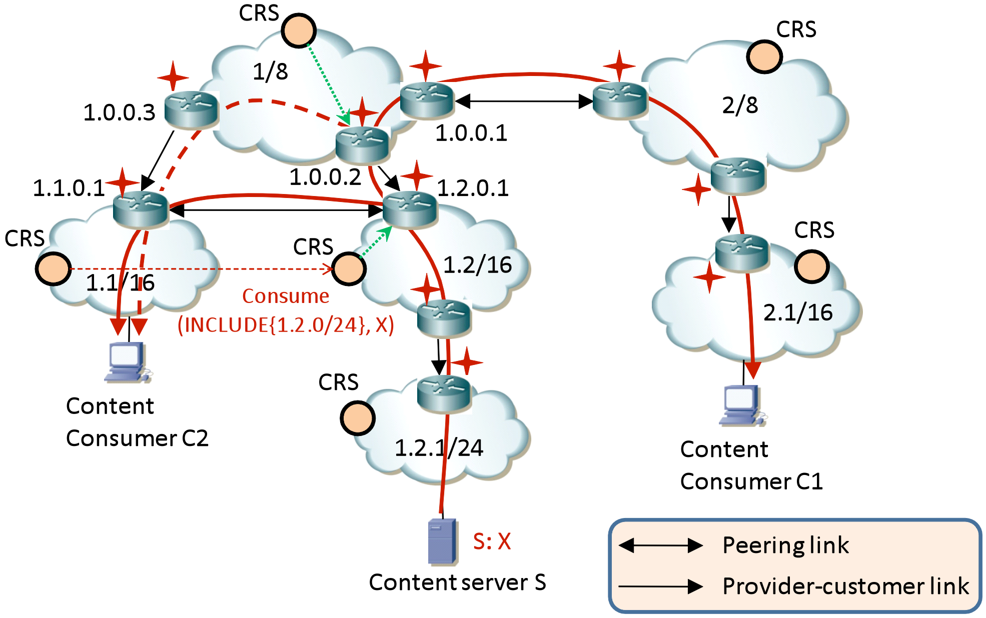

Figure 5: Multicast-based content delivery process

5.0 Discussion on Scalability

The domain-level hop-by-hop content resolution strategy presented follows a similar style to that proposed in [3]. However, through the new *scoping* and *filtering* functions, our architecture provides the necessary flexibility for both content providers and consumers to publish/request content at/from their desired area(s). The scalability of the system, thus, is dependent on the amount of content and the popularity of the content recorded within each CRS, with the most “vulnerable” CRSes being those that maintain the highest number of popular content entries. This is in contrast with intuition that the most strained CRSes will be the tier-1 ones, since content publications and requests may often not reach the tier-1 level based on our approach. Again, we take BBC iPlayer as an example where both the content publication and consumption requests are restricted only to IP prefixes from within the UK. In addition to that, local domain policies may also override the default publication route (see **S2** in Figure 3).

Business incentives also present a natural load distribution mechanism for our system. We foresee ISPs charging higher publication tariffs for popular content published at higher tier domains (with tier-1 domains being the most expensive) which are able to be potentially accessed by a higher number of consumers in the Internet. This mechanism forms a business tussle from the content providers’ point of view when provision of wider access is coupled with higher monetary cost. Instead, a content provider may strategically replicate content to multiple lower-tier regional ISPs (by applying scoping functions there) in which they believe their content will be locally popular.

Finally, our system also allows aggregation in two distinct ways. First, as illustrated in Figure 3 for **S4** and **S5**, the record for copies of the same content can be merged during the publication process among CRSes. Second, a block of sequential content IDs should be allocated to *inter-related* content so that they can be published in one single process. This rule exploits the fact that a specific content provider usually offers content with some relationship with each other (e.g., all episodes of a television series, or all football matches in a World Cup event). This allows for coarser granularity in the publication process whereby the content provider can send only one `Publish` message to publish all the related content. The local CRS still assigns a unique content ID for each content, but the IDs are sequentially connected. The onwards publication process will only involve the entire block of the IDs rather than the individual content records. This alleviates higher tier ISPs from the need to know each content hosted within and under their domain. Now, instead of matching explicitly the content ID in the `Consume` message, the CRS simply checks if the content ID is within a specific range of published content. The final location of the specifically requested content is actually handled at the *last-hop* domain where inter-related content entries are locally de-aggregated.

6.0 CONCLUSIONS

The host-oriented communication paradigm of the current Internet has highlighted many architectural drawbacks for the realization of the future content-centric Internet and thus, severely hampers the development of new innovative media manipulation methods, including publication, resolution/search and delivery processes. In this article, we describe CURLING, a new content-based Internet architecture that supports inherent content operations for handling content publication, resolution and delivery. Content providers can cost-efficiently publish content based on its expected popularity in different regions by *scoping* its publication while content consumers can express their location preferences by *scoping/filtering* their content consumption requests. The processes are devised so that both sides are oblivious of their counterpart's location, resulting in a bi-directional location independence paradigm, but without sacrificing content providers' and consumers' location preferences. As far as the content delivery is concerned, the newly proposed route optimization mechanism enhances the efficiency, where possible, by using content states established during the resolution process and initiating content delivery path switching, mimicking the inter-domain multicast content delivery paradigm that has seen very slow deployment until now.

REFERENCES

[1] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, "Networking Named Content," in CoNEXT '09. New York, NY, USA: ACM, 2009, pp. 1–12.

[2] P. Jokela, A. Zahemszky, C. E. Rothenberg, S. Arianfar and P. Nikander, "LIPSIN: Line Speed Publish/Subscribe Inter-networking", in *Proc. ACM SIGCOMM '09*, Barcelona, Spain, August 2009.

[3] T. Koponen, M. Chawla, B-G. Chun, A. Ermolinskiy, K. H. Kim, S. Shenker and I. Stoica, "A Data-oriented (and Beyond) Network Architecture," in *Proc. ACM SIGCOMM '07*, Kyoto, Japan, Aug. 2007.

[4] P. Francis and R. Gummadi, "IPNL: A NAT-extended Internet Architecture," in *Proc. of ACM SIGCOMM '01*, pp. 69-80, San Diego, CA, USA, Aug. 2001.

[5] I. Stoica, D. Adkins, S. Zhuang, S. Shenker and S. Surana, "Internet Indirection Infrastructure," in *Proc. of ACM SIGCOMM '02*, pp. 73-86, Pittsburgh, PA, USA, Aug. 2002.

[6] D. Clark, R. Braden, A. Falk and V. Pingali, “FARA: Reorganizing the Addressing Architecture,” in *Proc. of ACM SIGCOMM FDNA Workshop*, Aug. 2003.

[7] M. Caesar, T. Condie, J. Kannan, K. Lakshminarayanan, S. Shenker and I. Stoica, “Routing on Flat Labels,” in *Proc. of ACM SIGCOMM ’06*, pp. 363-374, Pisa, Italy, Sept. 2006.

[8] P. Mockapetris, “Domain Names – Concepts and Facilities”. Internet Engineering Task Force (IETF) Request for Comments (RFC), RFC 1034, November 1987

[9] The Cooperative Association for Internet Data Analysis (CAIDA) dataset, online at: http://www.caida.org/research/topology/#Datasets