A Multi-Level DHT Routing Framework with Aggregation

Hang Liu
InterDigital Communications LLC
781 Third Avenue
King of Prussia, PA 19406, USA
hang.liu@interdigital.com

Xavier De Foy
InterDigital Communications LLC
1000 Sherbrooke Street West
Montreal, QC H3A3G4, Canada
xavier.defoy@interdigital.com

Dan Zhang
WINLAB, Rutgers University
671 Route 1 South
North Brunswick, NJ 08902, USA
zhangdan@gmail.com

ABSTRACT

Information-Centric Networking (ICN) has recently attracted research attention, which decouples content from hosts at the network layer, and retrieves a content object by its name (identifier), instead of its storage location (host IP address) in order to address IP network's limitations in supporting content distribution. However, ICN systems face scalability and efficiency challenges in global deployments. In this paper, we propose a scalable routing and name resolution framework, called Scalable Multi-level Virtual Distributed Hash Table (SMVDHT). SMVDHT uses a combination of name aggregation and multi-level virtual DHTs to achieve scalability. A novel aggregation scheme is proposed to reduce the size and update overhead of name resolution tables, while relieving the "suffix-hole" problem encountered in traditional prefix-based name aggregation. Furthermore, SMVDHT exploits underlying intra- and inter-domain IP routing protocols to build multi-level virtual DHTs for name resolution, which is more efficient than conventional hierarchical DHT schemes and simplifies network management. We also design the new protocols to efficiently resolve the aggregated names and forward a request to the closest available copy of content via multi-level virtual DHTs.

Categories and Subject Descriptors

C.2.1 [**Computer-Communication Networks**]: Network Architecture and Design

General Terms

Design, Management

Keywords

Information-centric networking, Name-based routing, Name resolution, Scalability, Name aggregation, Multi-level DHT

1. INTRODUCTION

Information-Centric Networking (ICN), also referred to as Content-Oriented Networking (CON), has recently attracted research attention, which decouples content from hosts at the network layer, and retrieves a content object by its name (identifier), instead of its storage location (host IP address). This new networking paradigm has potential to address the inefficiency of IP networks in supporting content distribution. Several ICN architectures have been proposed [1-8], which share the same principles: a content object has a unique name and is cached at multiple locations in the network, the name is independent of the locations, and a publish/subscribe model is used to discover and retrieve the content by name. These proposals differ in the mechanisms such as content naming, routing and name resolution, etc. In particular, some designs employ flat, self-certifying names such as DONA [1], PSIRP/PURSUIT [3], NetInf/SAIL [4], and MobilityFirst [8], whereas others, e.g. CCN [2], adopt a hierarchical naming scheme with a structure like binary-encoded URLs. For content publishing, two approaches, announcing the content availability to other content routers (CRs) via a traditional flooding protocol [2] or a distributed hash table (DHT) scheme [3, 4], can be used. To retrieve a content object, the request is forwarded to the best content source(s) in the network employing either direct name-based routing on the requested object ID [2] or a name resolution process [3, 4] that resolves the ID into a network location (an IP address or a more general directive for forwarding).

ICN systems face scalability and efficiency challenges in global deployments. The number of content objects is huge, and rapidly growing. Even based on the current web size, a conservative estimate is that an ICN system should be able to handle at least 10^{12} objects [9]. Moreover, it is expected that a variety of cyber-physical communication scenarios such as wireless sensors and machine-to-machine (M2M) will be integrated into the Internet applications in the near future and the number of global information objects will be increased by at least several orders of magnitude [10]. These objects may be stored at any location(s) in the Internet. They are dynamically created, replicated and deleted.

To improve scalability, in CCN [2], the authors proposed to use hierarchical names with prefix-based aggregation in name-based routing information flooding, similar to the address aggregation in IP routing. If all the content objects whose names start with "example.com" are stored in a node, a single route announcement is needed, and CRs only need to maintain a single routing state for these objects in order to route a request to this node. However as content objects are cached or replicated at multiple places, such prefix-based aggregation becomes less effective. A caching node or CR may not have all the content objects with a given prefix. For example, suppose there are a total of N (e.g. $N=20000$) content objects for a prefix "example.com," and a node only stores M ($M=10000$) of them. If the prefix-based aggregation is used to avoid M routing states and associated routing update overhead, a lot of information will be lost. This is because a routing announcement with prefix-based aggregation can only express "some of the content objects with this prefix (example.com) may be reached via me." We refer to this as a

ICN'12, August 17, 2012, Helsinki, Finland.

suffix hole. Suffix holes introduce uncertainty in locating a particular content object and reduce routing efficiency. In the worst case, a CR needs to forward all requests matching a prefix to all the nodes that announce this prefix and receives multiple responses on multiple interfaces [2]. Furthermore, it is difficult to use the prefix-based aggregation on non-structured flat names. To address the weaknesses of prefix-based aggregation, in this paper, we first propose to use the prefix and the *digest of suffixes* for aggregation. The digest is generated from the suffixes of the aggregated object names using Bloom filters. The aggregation degree is controlled based on the content popularity to trade off between routing state reduction and information loss.

Another way to achieve good system scalability is to use a DHT scheme, instead of simple flooding, in which a participating node only maintains the location information for a portion of content objects. The nodes collaboratively provide a lookup service for name resolution. Some natural questions to ask are: *(1) can we combine the two techniques, aggregation and DHT, in an ICN system design for better scalability? (2) How can we do it?* Interestingly, all DHT-based ICN designs such as PSIRP/PURSUIT [3, 6] and NetInf/SAIL [4, 7] adopt flat, self-certifying names without aggregation. Moreover, they assume the name-location mappings obtained from the DHT are accurate and a content host can be reached by simply following the address provided by the resolution process. Given a huge and rapidly increasing number of content objects, some tradeoffs among bandwidth, storage, and computing resources need to be made in the design to achieve better scalability. The second contribution of this paper is to propose a scalable content routing and resolution framework, called Scalable Multi-level Virtual Distributed Hash Table (SMVDHT), which uses a combination of name aggregation and multi-level DHT techniques to improve ICN scalability. New protocols and procedures are designed to efficiently resolve the aggregated names and forward a request to the closest copy of content via multi-level DHTs.

We notice that most state-of-the-art DHT-based ICN designs such as PSIRP/PURSUIT [3, 6] and NetInf/MDHT/SAIL [4, 7] simply adopt the DHT schemes developed for peer-to-peer overlays such as Chord [12], Hierarchical Rings [13], and Canon [14]. They assume the DHT systems are independent of the underlying IP routing/forwarding layer and the node membership is very dynamic with frequent churns. These DHT schemes have their own bootstrapping and maintenance procedures although their hierarchical structure may reflect the underlying network topology. In addition, the existing designs clearly separate the name resolution process and the forwarding process into two steps. They are not very efficient on a global Internet scale.

We argue that the IP-based Internet infrastructure will not be thrown away. ICN would not completely replace IP just like IP would not replace Ethernet. Furthermore, the infrastructure routers running IP routing protocols can help design more scalable and efficient name resolution mechanisms. In contrast to the P2P environments with dynamic peer churns, the infrastructure routers are relatively stable in ICNs, while content object locations change frequently due to cache replacements. Based on the above observations, we propose to exploit the underlying intra- and inter-domain IP routing protocols to build multi-level virtual DHTs for name resolution, making name resolution as an integrated part of the routing and forwarding process. By virtual, it means that nodes do not run the dedicated DHT bootstrapping and maintenance protocols. IP routing is used to form multiple one-hop DHTs at different levels corresponding to the Internet hierarchy and optimize forwarding path. This also improves ICN evolvability and deployability.

The contributions of this paper are threefold:

- We propose SMVDHT, a new name resolution and routing framework that employs a combination of aggregation and multi-level virtual DHTs to improve ICN scalability.
- A name aggregation scheme is designed to use the prefix and the digest of suffixes to reduce the size and update overhead of name resolution tables, while relieving the suffix hole problem in traditional prefix-based aggregation.
- Multi-level virtual DHTs are constructed by fully exploiting the underlying intra- and inter-domain IP routing protocols, making the multi-level DHT-based name resolution as an integrated part of routing and forwarding.

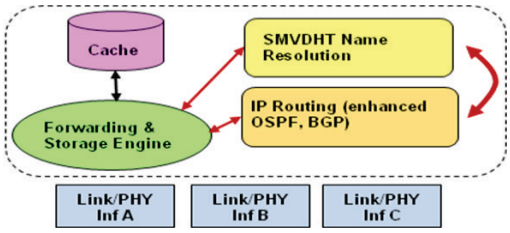

Figure 1. SMVDHT router model.

2. SYSTEM MODEL

SMVDHT defines a name resolution layer on top of the IP layer. A content router (CR) runs both IP routing and SMVDHT name resolution protocols, and also has data caching capability. Name resolution is an integrated part of the routing and forwarding process. ICN services can co-exist with other IP services such as traditional host-to-host communications. Conventional intra-domain routing, e.g. OSPF, and inter-domain routing such as BGP, are used for IP routing with certain extensions, e.g. a router can advertise its name resolution capability in its IP routing dissemination. Figure 1 illustrates a schematic of an SMVDHT content router. A host or a normal IP router can connect to an SMVDHT router as a client.

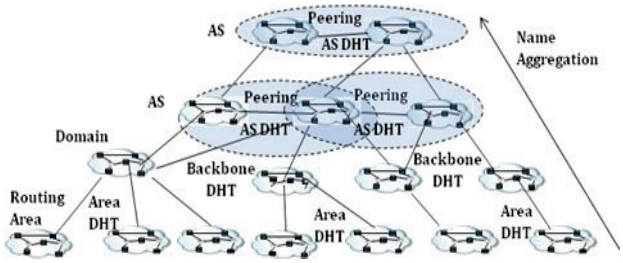

Figure 2. System model.

SMVDHT assumes that there is no change to the hierarchical structure of the current Internet infrastructure as well as the relationship between enterprise domains and Internet service providers (ISPs). This simplifies SMVDHT deployment. The multi-level virtual DHT structure reflects the underlying physical network topology. At the lowest level, the CRs in an OSPF routing area form a virtual DHT (VDHT) substrate using the link state information provided by the IP routing. Then the OSPF area

border routers form a higher-level backbone VDHT substrate for inter-area routing and name resolution. For inter-domain routing and name resolution, the multi-level VDHTs are formed using the information provided by BGP that reflects the Internet hierarchy and peering relationship among Autonomous Systems (ASes). A CR may join multiple VDHT substrates, and these VDHT substrates are interconnected to a tree-like structure but with multi-homing as illustrated in Figure 2. The location information of content objects is published in the multi-level VDHTs and at each level certain aggregation is performed based on the content popularity and VDHT level.

The integrated name resolution and IP address routing procedures provided by SMVDHT ensure that a content request is forwarded to the best or closest host(s) of the requested object by a set of "*delegated*" CRs. A response carrying the content data or an instruction to establish the content retrieval session is forwarded back to the requester along the same shortest path as the request travels so that en-route caching can be performed by intermediate CRs.

A packet carries both the IP address (location information) and content object ID (OID). The SMVDHT-related OID information is essentially inserted as a shim layer between the IP and transport headers. The location information, i.e. IP address is transient, and only the OID serves as a persistent and unique identifier for a particular content object. The IP addresses in a packet can be changed by an intermediate CR during forwarding in the network. For example, if an intermediate CR knows a better location for the requested object via name resolution, it can change the destination address of the request packet and forward the request toward the new destination. Note that the terms, "OID," "identifier," and "name" are used interchangeably in this paper, unless otherwise stated.

3. NAMING AND AGGREGATION

SMVDHT framework can accommodate both self-certifying flat names and hierarchical names. To address the suffix-hole problem, SMVDHT applies Bloom filters [11] to generate the digest of the suffixes for the aggregated OIDs and a CR announces both the prefixes and the suffix digests for its content objects. Bloom filters are a computationally efficient hash-based scheme with controllable error probability.

Let's first assume that the self-certifying flat names as proposed in DONA [1] are used, and the names have the form *P:L*, where *P* is a cryptographic hash of the principal's public key and *L* is a flat label. If a CR has the location information for *N* content objects whose names start with the same *P* value, it can use a summary OID (sOID) to represent these objects. The sOID has the form *P:digest(L)*, where *digest(L)* is generated by Bloom filters from the value *L* of the *N* aggregated names, $digest(L) = Bloom\text{-}Filter\{L_1, \ldots L_N\}$. Thus, the CR can just use a single sOID to publish the location information of these objects, and other CRs only need to maintain one routing state for this sOID. By using the sOID in the publishing process, it means "I have the location information for the content objects with this prefix, and the digest of the suffixes in their names equal to *digest(L)*.

Assume hierarchical names are used, for example, in the form of */example.com/movie/title* [2]. For *N* objects whose names start with the same prefix */example.com/movie*, an sOID, */example.com/movie/digest(titles)*, can be used to represent these objects. Similarly an sOID */example.com/digest(categories/titles)* can be used to represent the objects with the common prefix */example.com*, but different categories and titles.

To query whether an OID exists, there should be a match between the queried OID and an sOID, i.e. the corresponding prefix should be exactly the same and the digest in the sOID should give a positive match to indicate that the corresponding suffix element in the queried OID is likely to be present. Note that with Bloom filters, false positives are possible, but false negatives are not. It is possible that a collision in the digest occurs if the corresponding bits in the digest have been set during the insertion of other elements. The digest then incorrectly indicates an element is present. However, the false positive probability can be controlled by designing appropriate filters and limiting the number of aggregated elements that are added to a digest. Given a filter, a publishing CR can flexibly control the aggregation degree based on the popularity of the content objects or the distance to the content location to balance between the network resources needed for maintaining routing states and the false positive probability. For example, no aggregation is performed in publication for the content objects residing in the local network domain, given that users may request local content with higher probability. But a domain gateway router publishes the summary OIDs of its content objects to outside domains. The aggregation degree, i.e. the number of suffix elements added to a digest, can be limited to control the false positive probability. When the number of elements exceeds the limit, the elements are divided into groups. Each group generates a digest. A domain gateway can publish an sOID with a prefix and multiple digests, for example, $P:digest_1(L):digest_2(L):digest_3(L)$, each digest generated from a group of suffixes to be aggregated. One learns from the published sOID that requests for the content objects with this prefix and suffix digest may be served by this domain.

Since an sOID carries the prefix and suffix digest of the aggregated OIDs, it helps mitigate the suffix-hole problem while achieving routing scalability. In next section, we'll present how to handle the false positives in the name resolution and routing process.

4. NAME RESOLUTION AND ROUTING

A fundamental problem in content retrieval is to locate the closest copy of a particular content object on a global Internet scale, no matter where it resides. SMVDHT uses the underlying IP routing to build multi-level topology-aware one-hop virtual DHTs to provide a distributed name resolution service and efficiently locate content objects. In contrast to the existing DHT-based ICN designs [3, 4, 6, 7], SMVDHT integrates name resolution into the routing and forwarding process. Particularly, aggregation is used in higher level virtual DHTs to reduce control overhead and state requirements for better scalability. The SMVDHT routing and resolution procedures ensure the shortest path routing and local one-hop name resolution to achieve efficiency. In this section, we first describe SMVDHT intra-domain routing and then discuss inter-domain routing.

4.1 Intra-Domain Routing

We assume that the infrastructure routers in a network domain run a link-state IP routing protocol, e.g. OSPF. The link-state protocol provides a network topology so that a router knows the existence of all other routers in the same routing area (an OSPF domain can be divided into multiple areas and the topology information may be aggregated before disseminating to other areas). OSPF can be

extended to enable CRs to announce their name resolution and caching capabilities.

4.1.1 Intra-Area Routing

The content location information in a routing area is maintained using a one-hop DHT [15] that maps a content OID to a CR via a hash function and stores the corresponding content location information at the mapped CR. In our design, the one-hop DHT is formed and maintained using the link state information provided by the underlying OSPF. So we call it a virtual DHT (VDHT).

When a content router caches a content object, it creates a content location object (LO) with attributes *{OID, publisher, scope, timeout}*, where the OID is the content object ID, the publisher is the IP address of the node that publishes this LO, the scope specifies up to which level the publisher wants to make this object known (e.g. limiting the publication within the local network/routing area, the OSPF routing domain, the AS, or the Internet), and the timeout field represents the lifetime for this LO. The LO becomes invalid after it times out. The CR publishes its content LOs on the VDHT by using a hash function to map the OID to a responsible CR (called location resolver (LR)) in its routing area and send the corresponding LO to the LR. The LR will store the LO. A content host without routing capability simply uses its associated CR as the proxy to publish the location information of the content objects that it stores. Although possible, the content names are not aggregated within its own routing area since the CRs should have capability to handle local content and the content is more likely to be accessed by local users.

To map an OID to a LR, a cryptographic hash function [12] is used to assign a mapping identifier to each content OID and each CR. A CR's mapping identifier *H(Y)* is generated by hashing the CR's router ID *Y*, while the mapping identifier of a content OID *X*, *H(X)* is the value obtained by hashing *X*. The LR for the content OID *X* is thus the CR whose mapping identifier *H(Y)* is the closest to and not exceeding *H(X)* in the hash space. *H(X)* is the key for retrieving the corresponding LO and then the content object.

To maintain the freshness, a publishing CR periodically updates the LOs for its content objects to the mapped LRs. If a CR fails or a new CR is added, the underlying IP network topology changes and the LRs for some LOs may need to be changed. The CR that originally published an LO monitors the availability of its LR through the IP link-state advertisements, and will republish the LO to the new LR if needed. When a link fails or a new link is added, but the set of LRs does not change, the underlying IP routing protocol will update the link-state topology to ensure packets continue to travel along the shortest path. In this case, there is no need to take action for LO updates. To be more robust to network failures, an LO can be mapped to and stored at multiple nodes on the one-hop VDHT using multiple keys. A set of keys are generated by hashing an OID with different hash functions. The query can be done by using any one of the keys.

4.1.2 Inter-Area Routing with Aggregation

An OSPF routing domain consists of multiple areas and a backbone. Each area is connected to the backbone through its area border routers (ABR). The ABRs are connected by the backbone that provides connectivity across the areas. The topology information is aggregated across the areas, i.e. a router in an area may not know the existence of the other routers in another area. To resolve the location of a content object across routing areas, the ABRs form a backbone one-hop VDHT. The mapping identifier for an ABR is generated by hashing its router ID or the interface IP address connecting to the backbone.

The published names of the content objects are aggregated in the backbone VDHT for scalability. A LR inside a routing area aggregates its LOs with the common prefix in their OIDs into a summary location objects (sLO) and sends the sLO to the ABR of its routing area. An sLO contains the attributes *{sOID, publisher, scope, timeout}*. An sOID consists of $P{:}digest_1(L){:}digest_2(L){:}digest_3(L)$, as described in the last section. Note that the publisher here is the LR's IP address.

An ABR keeps the received sLOs from its internal routers in its local cache. It may receive multiple sLOs from its internal routers, and further aggregates them, combining a collection of the sLOs with the same prefix to generate a new sLO. Note that the publisher in the new sLO is the IP address of the publishing ABR, not the original internal routers. This indirection binding mechanism allows the locations of the objects to be moved within an area without letting outside know and to keep internal host IP addresses private.

A nice characteristic of Bloom filers is the linearity of their union operation. To combine two digests, the two digest values are simply ORed to obtain a new digest value. To control the false positive probability, if the total number of aggregated elements in the combined digest is over the aggregation limit, we don't combine the digest. Instead, these digests are concatenated in the sOID. For the LOs with the common prefix but different scopes, multiple sLOs are generated, each with a scope. An LO with a scope of local routing area is not aggregated and published outside the routing area.

An ABR publishes its sLOs by mapping the sOID to a responsible ABR on the backbone VDHT (the backbone LR). The key of an sLO is the hash value of the prefix in the sOID. For routing freshness, an internal LR will periodically send the sLOs to its ABR, and the ABR periodically updates its sLOs to the responsible backbone LRs. If an ABR fails or a new ABR is added, the publishing ABR will republish the sLOs that need to be moved to the new backbone LRs. Similarly, multiple keys can be used to map an sLO to multiple ABRs in the backbone VDHT for robustness. An sLO is stored at multiple mapped ABRs. The query can be done by using any one of the keys.

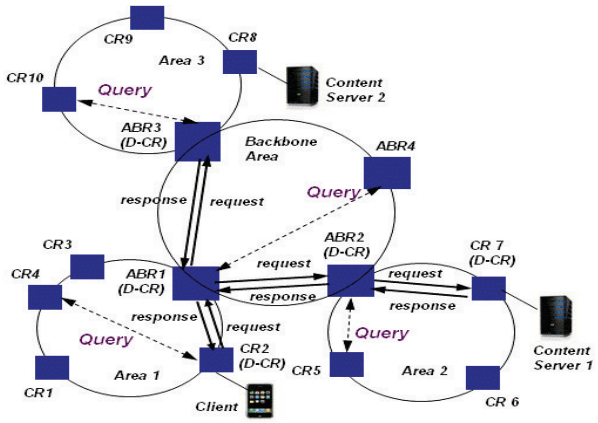

Figure 3. Inter-area name resolution and routing.

Figure 3 shows that a client wants to retrieve a content object by an OID. The client sends a content *request* to its associated content router CR2 in Area 1. CR2 becomes the *delegated* CR (*D-CR*) for this request that is responsible for resolving the content location and forwarding the request to the next hop. CR2 first sends a *query* message to the responsible LR in its own routing area, CR4, for resolving this OID based on the key. Note that a *query* message is different from a *request* message. The query is sent from a D-CR to a resolver on the same VDHT level locally, and the request is sent from the original content requester to the content host. If the local LR can resolve the OID, it sends the matching LO to the requesting D-CR. The D-CR then forwards the request to the content host. If there is more than one matching LO (multiple hosts), the choice is based on the D-CR's policy, e.g. forwarding the request to the closest one. This ensures to retrieve the requested content from the closest local host whenever possible and reduces inter-area traffic.

If the resolution at the local LR fails because of the OID unknown, the local LR (CR4 in Figure 3) sends a response to inform the D-CR (CR2) that the requested object is not found in Area 1. The D-CR then forwards the request to its area border router ABR1. ABR1 also becomes the D-CR for this request. Note that the OID routing states are aggregated in the backbone VDHT. ABR1 hashes the prefix of the OID to generate the key just as in the key generation process at the sLO publishing, and then sends a query to the responsible backbone LR, ABR4 , based on the key.

If one or more sLOs matched to this OID are available, the backbone LR ABR4 replies with all the matching sLO(s). The forwarding rule at a D-CR is as follows. If there is only one matching sLO, the content request is sent to the next-hop CR, i.e. the publisher of this sLO using IP routing. If there is more than one, based on the policy, the request is sent to all the matching sLO publishers in parallel or to one of the matching sLO publishers. If the request is sent to one of the matching sLO publishers, the choice of the publisher to send to is also determined by the policy. Due to routing aggregation, false positives may happen. If a delegated ABR router receives an error message after forwarding the request to a matching sLO publisher, it will send the request to the publisher of another matching sLO. This procedure continues until the content object is located or all the matching sLOs are tried. If the requested content object is not located after trying all the matching sLOs, the delegated ABR router forwards the request up to the autonomous system boundary router (ASBR) that uses an exterior routing protocol to route the content request across domains (please see next section). This approach allows the intermediate D-CRs to resolve the routing uncertainty hop-by-hop, using local queries to find and try different candidate paths. It is more efficient than the end-to-end approach.

In the example of Figure 3, ABR1 obtains two matching sLOs by querying the responsible backbone LR ABR4. It first sends the content request to ABR3 that is the publisher of the first matching sLO. ABR3 sends a query to the responsible LR in Area 3, CR10. CR10 replies with an error message to indicate no matching LO found. ABR3 then returns an error message to the sender of the request, ABR1. ABR1 sends the request to the publisher of the second matching sLO, ABR2. ABR2 queries the responsible LR in Area 2, CR5, and obtains the matching LO. It sends the request to the matching LO publisher, CR7 that forwards the request to the content host (Content Server 1 in Figure 3). The response from Content Server 1 will be forwarded by the same set of D-CRs hop-by-hop to the original requester along the same path (in reverse direction) as the request travels, given the underlying IP routing provides symmetric forward and reverse paths. Alternatively, ABR1 can send the request to all the matching sLO publishers in parallel to reduce delay.

A D-CR caches the content requests as well as the responses with expiry time. As an optimization, when a new request is received, it checks whether there is a match with one of the previous and unexpired requests or responses in the cache. If the new request matches a cached response, the D-CR can directly send the response to the sender of the request. If the new request matches a cached request, this D-CR is in the process of resolving the location of the requested content object. It does not forward the new request and simply waits for the response.

4.2 Inter-Domain Routing

Inter-domain routing uses a similar mechanism as inter-area routing described above. Multi-level one-hop VDHTs are formed which reflects the Internet hierarchy as shown in Figure 2. The OID routing states are aggregated at each level for scalability. A publisher can control the Bloom-filter aggregation degree based on the content popularity and the VDHT level. More suffixes are aggregated to generate a digest if these content objects are less likely to be requested at a higher level VDHT. ASes may form parent-child or peer relationships in the hierarchy. A content router running both BGP and name-based routing is called a BGP CR. BGP CRs in peered ASes form a one-hop VDHT for resolving content locations in these peered ASes. When a BGP CR receives a content request from a descendant router inside its AS, it becomes the delegated CR for this request and uses the same procedures as in the inter-area routing to resolve the content location and forward the request to the next hop. If there is no matching content object found in the peered ASes, the delegated BGP CR forwards the request to its parent (i.e., its provider). A BGP CR uses its policy to choose the parent if it is multi-homed. Thus, the unresolved requests are forwarded up the AS hierarchy to locate the requested object. This ensures to retrieve the closest copy of the requested content. If the request reaches a tier-1 AS and doesn't find the requested content object on the tier-1 VDHT, the delegated tier-1 BGP CR returns an error message. The response will be forwarded by the same set of delegated CRs hop-by-hop to the original requester along the same shortest path as the request traveled. The delegated BGP CRs can enforce the AS policies just as in IP routing.

5. PRELIMINARY EVALUATION

In order to provide a few first-order insights, we present a preliminary and simplified analysis of the proposed Bloom filter-based aggregation to the system scalability. As part of our future work, we plan to conduct a more thorough study on the SMVDHT routing and name resolution framework, and on the impact of OID aggregation (not only Bloom filter-based but also other aggregation techniques) to the ICN scalability.

For a Bloom filter with m bits in the array, k hash functions, and n aggregated elements, the false positive probability is approximately given as [11]

$$p_f = \left(1 - \left(1 - \frac{1}{m}\right)^{kn}\right)^k \quad (1)$$

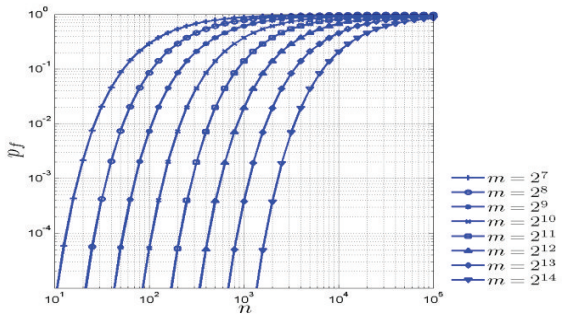

Figure 4. False positive probability of Bloom filter aggregation.

Figure 4 shows the false positive probability p_f as a function of the number of aggregated elements n in the filter and the filter size m, assuming an optimal value of k is used. The probability of false positives decreases as m increases, and increases as n increases. Note that sOIDs are only used in the content location publishing process, and are not carried in data packets. Thus there is great flexibility in designing the filter length to meet the requirements of false positive probability and the maximum number of elements to be aggregated in a filter. For example, 1000 suffix elements can be inserted in a 1024-byte long Bloom filter to generate a digest of the suffixes in an sOID (reducing the number of OID routing states by 1000 times), but the false positive probability is no more than 3 x 10^{-4}.

6. CONCLUSIONS AND FUTURE WORK

This paper proposes SMVDHT, a new name resolution and routing framework, which uses a combination of name aggregation and multi-level virtual DHTs to improve ICN scalability. A content router can publish both the prefix and the digest of suffixes to reduce the size and update overhead of name resolution tables, while relieving the "suffix-hole" problem encountered in traditional prefix-based aggregation. New protocols are designed to efficiently resolve the aggregated names and forward a request to the closest copy of content via multi-level virtual DHTs. For future work, we plan to prototype the proposed system and conduct extensive experiments to evaluate its performance and compare it with other name-based routing and resolution schemes.

7. REFERENCES

[1] T. Koponen, et al., "A Data-Oriented (and Beyond) Network Architecture," SIGCOMM '07, 2007, pp. 181–92.

[2] V. Jacobson et al., "Networking Named Content," CoNEXT '09, New York, NY, 2009, pp. 1–12.

[3] http://www.fp7-pursuit.eu/PursuitWeb/.

[4] Scalable and Adaptive Internet Solutions (SAIL). http://www.sail-project.eu/.

[5] D. Cheriton and M. Gritter, "TRIAD: A New Next-Generation Internet Architecture," Technical report, January 2000.

[6] K. Visala, et al., "An Inter-Domain Data-Oriented Routing Architecture," Workshop on Rearchitecting the Internet, 2009.

[7] M. D'Ambrosio, C. Dannewitz, H. Karl, V. Vercellone, "MDHT: A Hierarchical Name Resolution Service for Information-centric Networks," SIGCOMM Workshop on ICN, 2011.

[8] D. Raychaudhuri, "MobilityFirst Vision & Technical Approach Summary," MobilityFirst External Advisory Board Meeting, Feb 2011.

[9] A. Ghodsi, et al., "Information-Centric Networking: Seeing the Forest for the Trees," HotNet 2011.

[10] J. F. Gantz, et al., "The diverse and exploding digital universe: An updated forecast of worldwide information growth through 2011," IDC White Paper, March 2008.

[11] A. Broder and M. Mitzenmacher, "Network Applications of Bloom Filters: A Survey," Internet Mathematics, Vol. 1. No. 4, PP. 485-509, 2005.

[12] I. Stoica, et al, "Chord: a scalable peer-to-peer lookup service for Internet applications," SIGCOMM 2001.

[13] A. Mislove and P. Druschel, "Providing administrative control and autonomy in peer-to-peer overlays," IPTPS'04 workshop, 2004.

[14] P. Ganesan, K. Gummadi, H. Garcia-Molina, "Canon in G major: designing DHTs with hierarchical structure," ICDCS, March 2004.

[15] A. Gupta, B. Liskov, and R. Rodrigues, "Efficient Routing for Peer-to-Peer Overlays," NSDI, March 2004.