Digital Communications and Networks 5 (2019) 69–75

Contents lists available at ScienceDirect

Digital Communications and Networks

journal homepage: www.keaipublishing.com/dcan

A comparative study of name resolution and routing mechanisms in information-centric networks

Hang Liu [a,*], Koorosh Azhandeh [a], Xavier de Foy [b], Robert Gazda [c]

[a] Department of Electrical Engineering and Computer Science, The Catholic University of America, Washington, DC 20064, USA
[b] InterDigital Communications, Inc., Montreal, QC H3A 3G4, Canada
[c] InterDigital Communications, Inc., Conshohocken, PA 19428, USA

ARTICLE INFO

Keywords:
Information-centric networks
Content-centric networks
Name resolution
Name-based routing

ABSTRACT

Information-Centric Networking (ICN) is an innovative paradigm for the future internet architecture, which addresses IP network limitations in supporting content distribution and information access by decoupling content from hosts and providing the ability to retrieve a content object by its name (identifier), rather than its storage location (IP address). Name resolution and routing is critical for content retrieval in ICN networks. In this research, we perform a comparative study of two widely used classes of ICN name resolution and routing schemes, namely flooding and Distributed Hash Table (DHT). We consider the flooding-based routing in Content-Centric Networks due to its wide acceptance. For the DHT scheme, we design a multi-level DHT that takes into account the underlying network topology and uses name aggregation to further reduce control overhead and improve network efficiency. Then, we compare the characteristics and performance of these two classes of name resolution and routing through extensive simulations. The evaluation results show that the performances of these two approaches are reliant on several factors, including network size, content location dynamics, and content popularity. Our study reveals insights into the design tradeoffs and offers guidelines for design strategies.

1. Introduction

Future Internet has been envisioned to be built around information [1], instead of host-to-host connections that characterize today's internet. This new paradigm of networking is motivated by the observation that the predominant usage of networks is no longer for host-to-host communications, but for information access and content distribution. Existing IP networks require that two hosts establish a connection to transmit data, and data packets are forwarded along a path based on their destination location (IP address) in the network. This host-based communication approach does not natively support content-oriented applications and causes inefficiency in content delivery [2,3].

Information-Centric Networking (ICN) has the potential to address the inefficiency of IP networks in supporting content access. Several ICN architectures have been proposed [1–10], which share the same principle: content identity is decoupled from its location at the network layer. That is, a content object has a unique name and can be stored or cached at multiple locations in the network. The name is independent of the locations (host addresses), and is used to discover and retrieve a content object from the best location. These architectures differ from each other in the mechanisms that are involved, such as content naming, name resolution, and message routing.

Name resolution, i.e., locating a content object in the network, and request routing, i.e., routing the request message to the content source for retrieving the content object, are closely related. Specifically, there are two main classes of name resolution and routing techniques, namely flooding-based name routing and Distributed Hash Table (DHT) based name resolution. With the former [1,2], a Content Router (CR) announces the names of the available content objects to other CRs via a flooding protocol, and then a request for a content object is forwarded to the best content source(s) in the network based on the requested object name and Object IDentifier (OID). In order to improve scalability, a hierarchical naming scheme is often used to assign content objects, and certain name aggregation is performed during flooding to reduce overhead. For convenience of discussion, we refer to flooding as a name resolution scheme, that is, publishing the content availability information to all CRs by advertising, and a CR can resolve the name based on the published information and forward the request towards the content source.

With the latter [3,4], a DHT scheme [11,12] is used to map a content

* Corresponding author.
E-mail address: liuh@cua.edu (H. Liu).

https://doi.org/10.1016/j.dcan.2018.03.005
Received 1 December 2017; Received in revised form 14 February 2018; Accepted 21 March 2018
Available online 22 March 2018

OID to a CR, called the resolver, based on a hash scheme. For example, the resolver for a content OID with a hash value H(OID) is the CR with the hash value whose identifier or address is the closest to and not exceeding H(OID) in the hash space. The location information for the content object with this OID is stored at its resolver. A content object is retrieved as follows. On receiving a request with an OID, a CR queries the resolver responsible for this OID using DHT, resolves the requested OID into a network location (an IP address or a more general directive for forwarding), and then forwards the request message to the content source using the resolved address. Note that the terms, "OID," "identifier," and "name" are used interchangeably in this paper, unless otherwise stated.

As DHT does not require flooding during the content publishing process and a CR only needs to inform the necessary resolvers about the names of the content objects it provides, there seems to be a significant decrease in bandwidth overhead compared to the case in which flooding is required. However, DHT requires additional actions during the content retrieval process – whenever a CR receives a request for a content object by its OID from a client, the CR has to send a name resolution query to the corresponding resolver to obtain the current location information of the requested object before it can send the request to the target CR. On the other hand, flooding of content availability information in a network potentially wastes the bandwidth; however, it saves the need for name query and reduces the latency. It therefore remains unclear which approach is better under what conditions, causing a lasting debate between the two camps.

It is important to understand the design tradeoffs and compare the respective advantages of the different design choices. However, there is a lack of quantitative comparison for flooding-based name routing and DHT-based name resolution mechanisms. In this study, we identify the fundamental differences between these two approaches and investigate their design tradeoffs under different network scenarios. We conduct a performance comparison and try to understand how different factors, such as network size, content location changes, and content request rate, affect the performances of the two types of schemes. We consider the flooding-based routing in Content-Centric Networks (CCN) [1,2] due to its wide acceptance. For the DHT scheme, we design a Scalable Multi-level Virtual Distributed Hash Table (SMVDHT) scheme that takes into account the underlying physical network topology and uses name aggregation to further reduce the control overhead and improve efficiency. The initial design of SMVDHT was reported in Ref. [6]. In this research, we will enhance it and use it to perform a comparative study. Our work reveals insights into the tradeoffs of different approaches and offers guidelines for ICN name resolution and routing design.

This paper is organized as follows: Section 2 discusses and analyzes flooding-based name routing. Section 3 describes the DHT-based name resolution. In Section 4, the evaluation results are presented to compare and understand the performance tradeoffs of these two approaches. Section 5 concludes the paper.

2. Flooding-based routing

A content object is uniquely identified by its name and is stored in one or more servers or repositories. For flooding-based name routing, content availability is announced to all the CRs in the network. CCN [1] is a well-known ICN network architecture that uses such a flooding model. CCN also uses a hierarchical naming scheme for content where the name of a content object is formed by a sequence of components, such as Component_1/Component_2/ … /Component_n. Each component is a variable-length binary value. It is like a binary encoded Web URL, e.g., cnn.com/videos/title1.mpg. A component can also be added to represent the content version. In addition, a large object can be divided into chunks or packets. A component can be added to represent the chunk number, e.g., /chunk1. A CR may announce a content prefix, e.g., /cnn.com/videos if it connects to a repository that stores all or some of the /cnn.com/videos files, which indicates that it can reach some of the content with the prefix "/cnn.com/videos". Conventional intra- and inter-domain IP routing protocols, such as OSPF and BGP, can be enhanced to distribute CCN content prefixes within a domain or across domains as described in Ref. [1].

The CRs in CCN maintain a Forwarding Information Base (FIB), Pending Interest Table (PIT), and Content Store (CS). When a CR receives a content prefix announcement, it will create an entry for that prefix in the FIB pointing at the interface from which it receives the announcement. To retrieve a content chunk in CCN, a user sends an Interest, i.e., a content request containing the requested content chunk name. Once an Interest is received on some interface, *inf*, a CR will (i) check the local CS and return a copy if the requested content chunk is present; (ii) if not, it will check if the PIT has an entry for this content chunk and if there is an entry, it adds *inf* to the entry. Otherwise, it creates a new entry indicating *inf* for this Interest in PIT and forwards the Interest on the interface(s) indicated by the FIB for that chunk name. CCN also uses the prefix longest match to forward the Interest similar to that in IP. However, the forwarding behavior of CCN is different from that in IP, as there may be multiple announcements of the same prefix. In IP, a router forwards all matching traffic towards *a* destination via an interface. In CCN, a CR router can forward an Interest to *all* matching prefix announcers through multiple interfaces, because a prefix announcement indicates that the announcer may reach some of the content with that prefix [1].

When a Data packet is received, the CCN router (i) returns the packet over all interfaces with the pending Interests, as indicated by the corresponding PIT entry, and deletes that entry; (ii) if appropriate, it stores a copy of the packet in the local CS. Fig. 1 illustrates the message routing in a CCN network. Content routers advertise the reachability of the content at sources S1 and S2. User U1 issues an Interest for a content chunk to its access CR1, which forwards the Interest to CR3. CR3 will forward the Interest towards two potential source nodes S1 and S2 along two paths. Each Data packet is returned over the reverse paths: one from S1 to CR3 and the other from S2 to CR3. CR3 will forward one copy of Data to CR1, which delivers Data to U1. It is possible that a CR will simultaneously receive the Interests for the same content chunk from multiple users. For example, in Fig. 1, CR3 receives two Interests from U1 and U2 for a given content chunk; it will forward only one copy of the Interest towards the source. The source will return the corresponding Data to CR3. CR3 will forward a copy of Data towards U1 and another copy towards U2. CCN can thus naturally support multicast and multihoming, and bring popular content closer to users through caching.

3. DHT-based name resolution

Several DHT-based name resolution schemes have been proposed for ICN networks. In Ref. [9], a Global Name Resolution Service (GNRS) was designed, which employs a 1-hop DHT scheme for the dynamic binding of names to addresses. To publish a binding between the content object ID and the addresses of the object hosts, the ID is hashed with one or more hash functions in order to derive the network addresses of one or more GNRS routers, which will act as the resolvers to store the binding. The same 1-hop hashing scheme is used to forward the query for an ID to the resolving GNRS routers that hold the binding for this ID in order to resolve the query. However, this 1-hop DHT scheme is not locality-aware, and the resolvers may be far away from the requesters, which lowers the

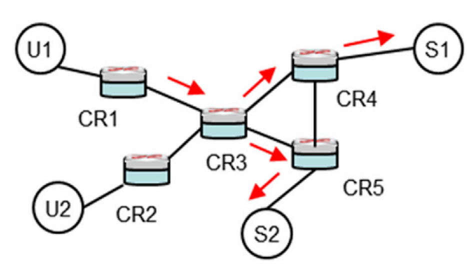

Fig. 1. Interest routing in CCN networks.

network efficiency. Moreover, it is used only for flat names without aggregation.

SAIL [4], an EU FP7 ICN project employs a multi-level DHT scheme for name resolution [8], in which multiple DHT areas are arranged in a nested, hierarchical structure, reflecting the underlying network topology. Another EU FP7 project, PURSUIT [3,16], uses rendezvous nodes to match a subscription to a publication in a domain, and inter-domain name resolution is handled by a network of rendezvous nodes that are also organized as a hierarchical DHT structure. However, these designs are based on the DHT algorithms developed for peer-to-peer overlay networks [11–14], which assume that the DHT systems are independent of the underlying routing/forwarding layer, and the node membership may be dynamic with frequent churns. They have their own mechanisms for discovering, bootstrapping, and maintaining the DHT, which result in complexity and inefficiency. In addition, the name aggregation only happens at the top level and a global resolution exchange system managed by an independent third party is used for inter-domain resolution, which causes extra overhead. Further, the existing designs clearly separate the name resolution process and the forwarding process into two steps, which are not efficient in a multi-level scale.

We argue that the IP-based Internet infrastructure will not be thrown away. ICN would not completely replace IP just as IP would not replace Ethernet. Furthermore, the infrastructure routers running IP routing protocols can help design more efficient and scalable 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 and make name resolution a unified part of the routing and forwarding process. By virtual, it means that nodes do not run the dedicated DHT bootstrapping and maintenance protocols [12]. IP routing is used to form and maintain multiple one-hop DHTs at different levels corresponding to the Internet hierarchy and optimize the forwarding paths.

We design a SMVDHT scheme, which improves upon the existing DHT schemes. SMVDHT uses a combination of name aggregation and multi-level virtual DHTs for efficient and scalable name resolution. The aggregation reduces the size and updating overhead of the name resolution tables. Multi-level virtual DHTs are constructed by fully exploiting the underlying physical network topology and routing protocols so that they are more efficient than conventional hierarchical DHTs and simplify network management. In addition, the name resolution is integrated in the routing and forwarding process. We also design the new protocols and procedures required to efficiently resolve the aggregated names and forward requests to the closest available copy of content via multi-level DHTs, both within a domain and across domains.

3.1. Multi-level DHT

SMVDHT defines a name resolution layer on top of the IP layer. A CR runs both IP routing and SMVDHT name resolution protocols and has data caching capability. In this way, 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 SMVDHT content router can advertise its name resolution capability in its IP routing dissemination. Fig. 2 illustrates a schematic of an SMVDHT router. A host or a normal IP router can connect to a SMVDHT router as a client.

SMVDHT assumes that there is no change to the hierarchical structure of the current Internet infrastructure as well as to 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

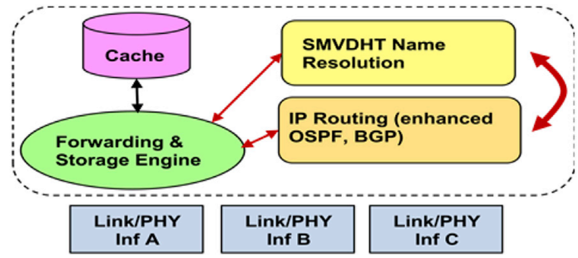

Fig. 2. SMVDHT router model.

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, multi-level VDHTs are formed using the information provided by the BGP, which 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 form a tree-like structure, but with multi-homing as illustrated in Fig. 3. The location information of content objects is published in the multi-level VDHTs, and at each level, certain aggregation is performed as described in detail in Section 3.3.

The integrated name resolution and IP address routing procedures provided by SMVDHT ensure that a content request is forwarded to the best or closest source(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 while being forwarded 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.

3.2. Intra-area routing

A fundamental problem in content retrieval is to locate the closest copy of a specific content object in the network, no matter where it resides. 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 (note that an OSPF domain can be divided into multiple areas and the topology information is aggregated before disseminating to other areas). We also extend OSPF to enable CRs to announce their name resolution and caching capabilities.

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 sends the corresponding LO to the LR. The LR will store the LO. A content host without routing capability simply uses its associated CR as a proxy to publish the location information of the content objects that it stores.

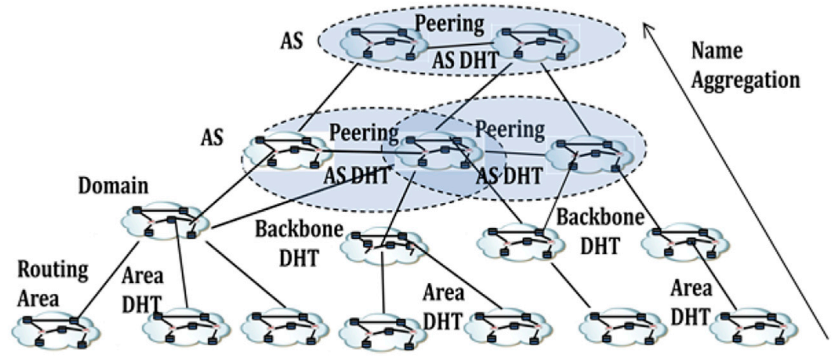

Fig. 3. Hierarchal network model.

Although possible, the content names are not aggregated within its own routing area, since the CRs should have the capability to handle local content and the content is more likely to be accessed by local users.

To map an OID to an LR, a cryptographic hash function [15] 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, but not exceeding, *H(X)* in the hash space. Since every CR knows how to reach other CRs in an area via OSPF, a CR can use *H(X)* as the key to finding the resolver and directly retrieving the corresponding LO when receiving a request. This forms a one-hop DHT to maintain the content location information in a routing area. In our design, the one-hop virtual DHT is formed and maintained using the link state information provided by the underlying OSPF.

To maintain the freshness, a publishing CR periodically updates the LOs of its content objects to the mapped LRs. If a CR fails, or if 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 if 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 that the 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 DHT 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.

3.3. Inter-area routing

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 DHT. The mapping identifier for an ABR is generated by hashing its router ID or the interface IP address connected to the backbone.

The published names of the content objects can be aggregated in the backbone DHT for scalability. An LR inside a routing area may aggregate its LOs with the common prefix in their OIDs into a summary Location Object (sLO) and send the sLO to the ABR of its routing area. An sLO contains the attributes *{sOID, publisher, scope, timeout}*, where sOID is a prefix. Note that the publisher here is the LR's IP address.

An ABR keeps the sLOs received from its internal routers in its local cache. It may receive multiple sLOs from its internal routers; it aggregates them further, combining a collection of 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 the outside know and to keep the internal host IP addresses private.

An ABR publishes its sLOs by mapping the sOID to a responsible ABR on the backbone DHT (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 if 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 DHT for robustness. An sLO is stored at multiple mapped ABRs. The query can be done by using any one of the sLO keys.

Fig. 4 shows an example in which a client wants to retrieve a content object by an OID. The client sends a content *request, i.e., interest,* to its associated content router CR2 in Area 1. CR2 becomes the *Delegated* CR (*D-CR*) for this request, which 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 DHT 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 that the requested content is retrieved from the closest local host whenever possible, which reduces traffic.

If the resolution at the local LR fails because of the unknown OID, the local LR (CR4 in Fig. 4) 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 DHT. ABR1 hashes the prefix of the OID to generate the key, just as in the key generation process for 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 matching sLQ, 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 is determined by the longest prefix matching. 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

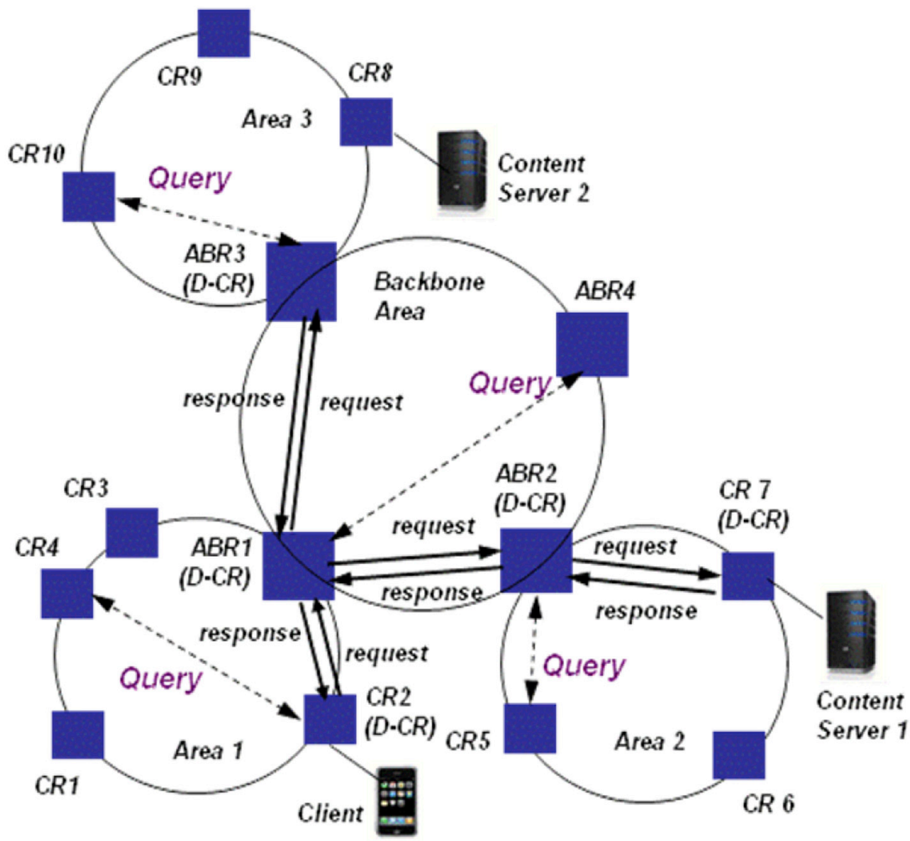

Fig. 4. Inter-area name resolution and routing.

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), which uses an exterior routing protocol to route the content request across domains (please see the next section). This approach incorporates the name resolution in the request forwarding process and 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 shown in Fig. 4, ABR1 obtains two matching sLOs by querying the responsible backbone LR ABR4. It first sends the content request to ABR3, which 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 that no matching LO is 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, which forwards the request to the content host (Content Server 1 in Fig. 4). 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 latency.

A D-CR caches the content requests as well as the responses with an 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 will become involved 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, which is similar to the PIT mechanism in CCN.

3.4. Inter-domain routing

Inter-domain routing uses a similar mechanism as that of inter-area routing described above. Multi-level one-hop virtual DHTs are formed, which reflects the Internet hierarchy, as shown in Fig. 3. The OID routing states may be aggregated at each level for scalability. One learns from the published domain-level content sOIDs that requests for the content objects with this prefix may be served by this domain. 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. The BGP CRs in the peered ASes form a one-hop DHT 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 inter-area routing to resolve the content location and forward the request to the next hop. If no matching content object is 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 multihomed. Thus, the unresolved requests are forwarded up the AS hierarchy to locate the requested object, and retrieval of the closest copy of content is ensured. If the request reaches a tier-1 AS and the requested content object cannot be found on the tier-1 DHT, 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.

4. Evaluation

In order to provide insights into the flooding-based and DHT-based name resolution and routing approaches, we conduct simulations using

NS3 to study the performance of CCN and SMVDHT under different network sizes, content location change rates, and content request rates. We are mainly concerned with the control bandwidth overhead due to content reachability publishing and name resolution, as well as the latency to retrieve content objects.

We consider a two-tier network with four routing areas and a backbone area. For flooding-based name routing, we use CCN implementation [1]; for DHT-based name resolution, we assume that the underlying network is IP, that is, the content names will be resolved into IP addresses, and the content routers run SMVDHT for name resolution and OSPF for IP routing. The numbers of content routers in each area are varied in order to study the impact of network size on control overhead and latency. For each number of content routers, we randomly generate 40 mesh topologies. Network control overhead and latency are calculated over the 40 topologies to obtain the mean value. For each topology, the user terminals and content servers are connected to the content routers as leaves. We also change the content request rates from the users and the content publishing rates at the servers to investigate their impact on the performance of the flooding and DHT name resolution and routing schemes. The content retrieval rates reflect the content popularity, and the content publishing rates represent how dynamic the changes in the location of the content objects are.

Fig. 5 shows the network control overhead for flooding and DHT based schemes under different numbers of content routers in each area. In this simulation, we assume that there exist ten users per content router and two content servers that are randomly connected to the content routers in an area. The content request is generated uniformly at an average rate of 100 requests per second, and the content publishing rate is 20. As expected, we see that the control bandwidth overhead significantly increases with the number of CRs for the flooding-based routing scheme, and remains relatively stable for the DHT scheme. This is because more control messages are propagated for the flooding scheme to advertise the content reachability information to all the CRs as the network size increases. However, in the case of DHT name resolution, a CR only needs to inform the necessary resolvers about the names of the content objects it provides and insert the name-address binding information regardless of the size of the network. In addition, we see that the flooding scheme costs control bandwidth overhead close to that of the DHT scheme only when the network size is very small. Otherwise, DHT incurs much less control overhead.

Figs. 6 and 7 show the behavior of control overhead of the two schemes with the content request rates when the network sizes are 20 CRs and 100 CRs per area, respectively. The publishing rate is fixed at 20 messages per second. For the DHT scheme, the control overhead increases with the increase in the content request rate, because upon receiving a content request, it needs to send a name resolution query to the corresponding resolver to obtain the current location information of the requested object before it can forward the request to the content source. The overhead of the flooding scheme is relatively stable with the content request rate, because a CR can directly route the request to the potential content source using the forwarding table already established by the content availability announcement. In addition, as expected, the control overheads increase with the increase in network size. Compared to flooding, DHT significantly decreases the control bandwidth overhead in publishing content reachability. However, DHT requires additional actions in the content retrieval process. The flooding mechanism has only "publishing" cost, whereas DHT has both "publishing" and "query" costs.

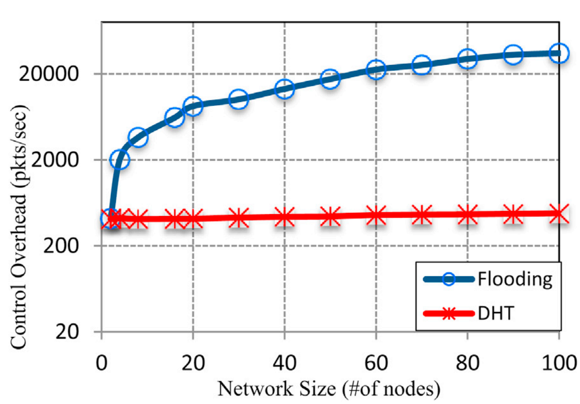

Fig. 5. Average control overhead of the flooding and DHT name resolution and routing schemes with different numbers of content routers per area.

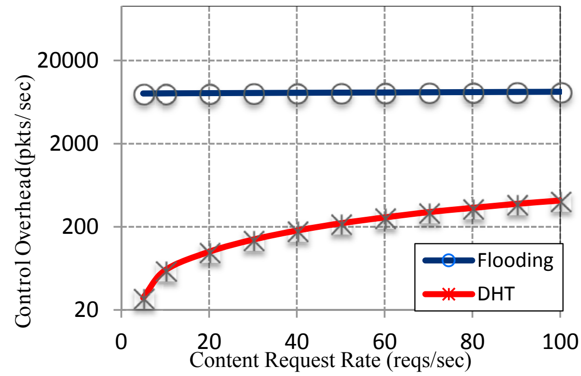

Fig. 6. Average control overhead of the flooding and DHT name resolution and routing schemes with different content request rates for small-size networks with 20 content routers per area.

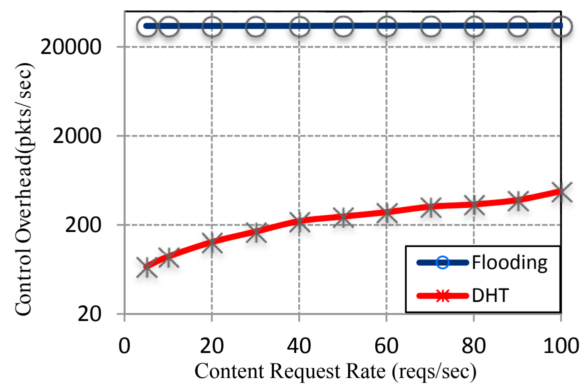

Fig. 7. Average control overhead of the flooding and DHT name resolution and routing schemes with different content request rates for large-size networks with 100 content routers per area.

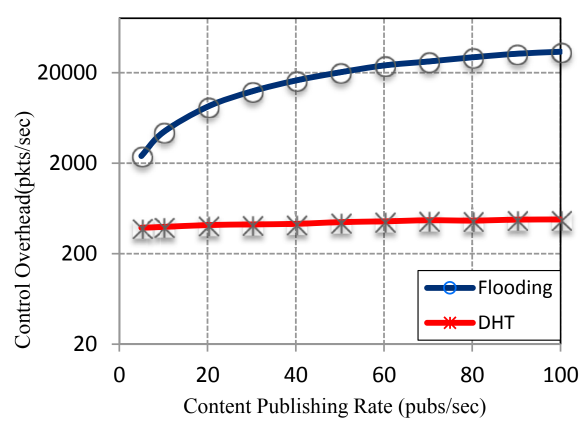

Fig. 8. Average control overhead of the flooding and DHT name resolution and routing schemes with different content publishing rates for small-size networks with 20 content routers per area.

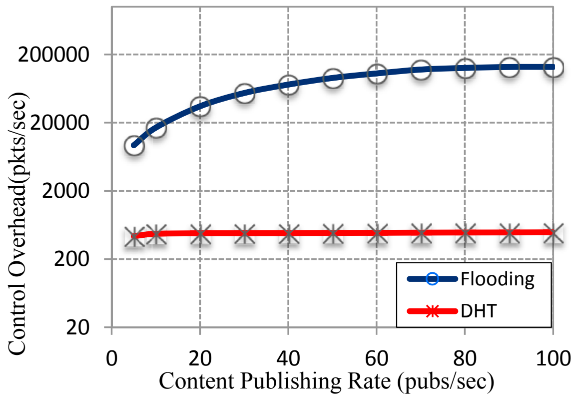

Fig. 9. Average control overhead of the flooding and DHT name resolution and routing schemes with different content publishing rates for large-size networks with 100 content routers per area.

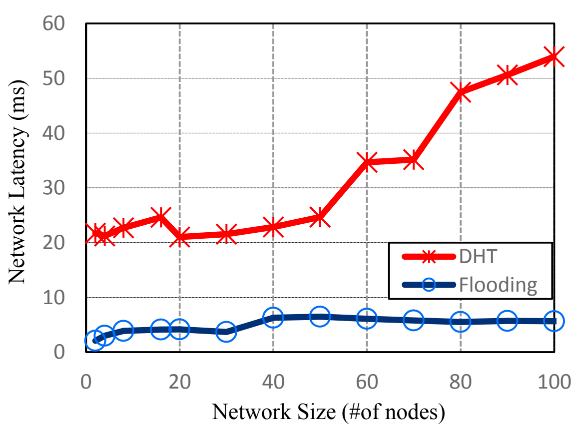

Fig. 10. Average end-to-end delay for the flooding and DHT name resolution and routing schemes.

Figs. 8 and 9 show the behavior of control overhead for both the schemes with the content publishing rates when the network sizes are 20 CRs and 100 CRs per area, respectively. The content request rate is fixed at 100 requests per second. The control overhead of the flooding scheme increases with the content publishing rate, because the content availability is disseminated to all the CRs in the network during publishing. The control overhead of the DHT scheme is relatively stable with the change in the content publishing rate, because only the corresponding resolver needs to be updated about the content name-address binding information.

Fig. 10 shows the end-to-end content retrieval latency for the two schemes. The latency of the DHT scheme is higher than that of the flooding scheme, because a CR needs to query the resolver to obtain the content location before it can forward the request to the content source during the content retrieval process.

5. Conclusions

In this study, we evaluated the performance of two classes of ICN name resolution and routing techniques, namely flooding and DHT, under various network sizes, content publishing rates, and content request rates. In the comparative study, we identified the key differences between these two approaches and examined the costs introduced by their different behaviors. Compared to flooding, DHT significantly decreases the control bandwidth overhead in publishing content reachability, especially when the network size is large and the content publishing rate is high. However, DHT requires additional query in the content retrieval process. The flooding mechanism has only “publishing” cost, whereas DHT has both “publishing” and “query” costs. In addition, the latency of the DHT scheme is higher than that of flooding, due to the query process. We believe that our study provides valuable insights regarding design tradeoffs and that it can provide a guideline for designing ICN name resolution and routing mechanisms.

References

[1] V. Jacobson, et al., Networking Named Content, CoNEXT ’09, New York, NY, 2009.
[2] W. Shang, A. Afanasyev, L. Zhang, The Design and Implementation of the NDN Protocol Stack for RIOT-OS, GLOBECOM Workshop on Information Centric Networking Solutions for Real World Applications, December 2016.
[3] N. Fotiou, P. Nikander, D. Trossen, G.C. Polyzos, Developing information networking further: from PSIRP to PURSUIT, Lect. Notes Ins. Comp. Sci., Soc. Inf. Telecommun. Eng., LNICST 66 (2012) 1–13.
[4] Scalable and Adaptive Internet Solutions (SAIL). http://www.sail-project.eu/.
[5] T. Koponen, et al., A Data-oriented (And beyond) Network Architecture, SIGCOMM ’07, 2007, pp. 181–192.
[6] H. Liu, X. De Foy, D. Zhang, A multi-level DHT routing framework with aggregation, in: Proceedings of ACM SIGCOMM ICN'12, August 2012, pp. 43–48. Helsinki, Finland.
[7] K. Visala, et al., An Inter-domain Data-oriented Routing Architecture, Workshop on Rearchitecting the Internet, 2009.
[8] M. D'Ambrosio, C. Dannewitz, H. Karl, V. Vercellone, MDHT: a Hierarchical Name Resolution Service for Information-centric Networks, SIGCOMM Workshop on ICN, 2011.
[9] D. Raychaudhuri, K. Nagaraja, A. Venkataramani, Mobilityfirst: a robust and trustworthy mobility-centric architecture for the future internet, ACM SIGMOB - ILE Mob. Comput. Commun. Rev. 16 (3) (2012) 2–13.
[10] J. Chen, S. Li, H. Yu, Y. Zhang, D. Raychaudhuri, R. Ravindran, H. Gao, L. Dong, G. Wang, H. Liu, Exploiting ICN for realizing service oriented communication in IoT, IEEE Commun. Mag. 54 (12) (Dec. 2016) 24–30.
[11] I. Stoica, R. Morris, D. Liben-Nowell, D.R. Karger, M.F. Kaashoek, F. Dabek, H. Balakrishnan, Chord: a Scalable Peer-to-peer Lookup Protocol for Internet Applications, IEEE/ACM Trans. on Networking, Feb. 2003.
[12] P. Ganesan, K. Gummadi, H. Garcia-Molina, Canon in G Major: Designing DHTs with Hierarchical Structure, ICDCS, March 2004.
[13] A. Gupta, B. Liskov, R. Rodrigues, Efficient Routing for Peer-to-peer Overlays, NSDI, March 2004.
[14] A. Mislove, P. Druschel, Providing Administrative Control and Autonomy in Peer-to-peer Overlays, IPTPS’04 workshop, 2004.
[15] T.M. Cover, J.A. Thomas, Elements of Information Theory, Wiley-Interscience, 1991.
[16] D. Trossen, G. Parisis, Designing and realizing an information-centric internet, IEEE Commun. Mag. 50 (4) (July 2012) 60–67.