

SCOM: A Scalable Content Centric Network Architecture with Mobility Support

Dawei Li

Computer Science and Engineering Department
Lehigh University
Email: dal312@lehigh.edu

Mooi Choo Chuah

Computer Science and Engineering Department
Lehigh University
Email: chuah@cse.lehigh.edu

Abstract—Content centric network (CCN) enables users to retrieve contents of interest (e.g. videos, music etc.) using the content names directly. Researchers face two critical challenges in CCN design: content naming and mobility support. The content naming is a fundamental building block for CCN. Users must be able to obtain names of contents of interest fast so that contents can be correctly retrieved. Meanwhile, the rapid development of wireless technologies especially the emergence of cellular 4G networks allows users to retrieve content on the move. However, since mobile users may frequently change their attachment points (e.g. 4G, WiFi, WiMax etc.), the CCN architecture must be able to handle mobility effectively so that a user can still get the content when being connected to a different attachment point. In this paper, we propose SCOM, a scalable content centric network architecture with mobility support, to solve those two issues. SCOM adopts an improved keyword-based naming resolution method to enable faster name retrieval. Users could retrieve names of contents of interest by sending keyword queries to the corresponding Content Resolution Servers (CRS). For mobility, SCOM provides efficient solutions for both keyword-based queries, and content retrievals for users on the move. Via large scale simulation studies using real internet topology, we demonstrated that our solutions are effective in terms of shorter query response times and faster content retrievals during mobility.¹

Index Terms—future internet, naming, mobility, content centric network, scalability

I. INTRODUCTION

The Internet today is more like a huge content storage where users retrieve all kinds of interested contents such as videos, musics, images, news, tweets etc. Just in a single year of 2008, the new content available for user access is roughly 500 exabytes [2]. In Van Jacobson's Google Tech Talk Speech [3], he cited several measurement results and showed that more than 99% of today's network traffic is for a machine to acquire named contents including web pages, email messages etc. Those facts illustrate the current network usage pattern: for most of the time, users only want to know what content they can get from the Internet and retrieve contents of interest as fast as possible. However, current host-based Internet architecture is inefficient for content retrieval in the sense that a user needs to establish a connection with a host serving that content before the content can be retrieved. Meanwhile, there is no automatic way for users to locate the closest copy of a content, i.e. with minimum retrieval latency, but they have to purely based on their own past experience or use the ranked list provided by searching engines. The Content Distribution Network (CDN) provides a business model for hosting different copies of contents of a particular content publisher (such as Yahoo.com) at different network locations

to minimize retrieval latency and balance traffic load, but fundamentally it is still based on a host-based network, and the ever increasing private contents (e.g. personal videos, photos) cannot enjoy this benefit.

To better meet the need of current Internet usage patterns, researchers have proposed the architecture of CCN (Content Centric Network) [1] which allows users to retrieve contents of interests with just the content names by sending "Interest Packets". Network packets (both Interest and Data packets) are transmitted purely based on the content names instead of IP addresses. CCN supports content caches in any CCN nodes so that if an interested content exists in the cache of an intermediate node, it will be retrieved and replied directly without forwarding the "Interest Packet" to the original hosting node (the content publisher). This paper and several research projects have proved that CCN can be used to efficiently deliver contents for all current network applications including those instant message (IM) Applications.

Despite these exciting results, CCN, however, still faces many open issues, two of which are content naming and mobility support. The content naming is the fundamental building block for CCN and users must be able to acquire the desired content names fast so that the content can be correctly retrieved. Most of current CCN proposals just assume users know the content names they are interested in but actually most of the time they do not (that is also why people are using Google so frequently). Therefore, the robustness of CCN requires an efficient way for users to get the names content objects of interest. Meanwhile, the rapid development of wireless technologies especially the emergence of cellular 4G network allows users to retrieve content on the move. As reported by Strategy Analytics [4], there have been more than 1 billion smart-phone users by Oct. 2012. Since mobile users may frequently change their access points (e.g. 4G, WiFi, WiMax etc.) due to the cost and availability reasons, the CCN architecture must be able to handle the mobility effectively so that users can still get the content seamlessly when connecting to a different access point.

We propose SCOM, a scalable content centric network architecture with mobility support, to deal with the two challenges. With SCOM, we have made the following contributions:

- SCOM adopts an improved keyword-based naming resolution method to enable fast name retrieval. Users could retrieve the interested content names by sending keyword queries to the corresponding Content Resolution Servers (CRS).
- SCOM provides mobility solutions for both name query

¹This project is supported by NSF NeTS Grant 1049845.

and content query to ensure the best performance for users on the move.

- We have implemented a large-scale simulation to demonstrate the performance of SCOM with real Internet topology. As far as we know, we are one of the earliest project measuring the mobility performance of large-scale content centric network.

The rest of the paper is organized as follows. In Section 2, we review the related work. In Section 3, we present the overview of SCOM architecture. In the two subsections, we describe the fast keyword-based name retrieval sub-system and illustrate the mobility support mechanisms for both name query and content query. Section 4 reports our simulation result and Section 5 concludes the paper.

II. RELATED WORK

A. Fast Naming Resolution

Several previous works have proposed solutions for fast naming resolution. DMAP [5] provides a fast "global unique content id" (GUID) to IP address resolution method by hashing a GUID to an Autonomous System (AS) number and storing its GUID-to-IP pair in the BGP router of that AS. The content owner inserts and updates the IP address of the GUID while clients retrieve the IP address with the same hash function. The underlying BGP routing protocol ensures the scalability of DMAP in the global Internet scale. DMAP reduces the overall query latency by hashing a GUID to different ASes with additional hash functions and users can always retrieve from the AS with minimum latency. However, this can increase the storage overhead of BGP routers especially those in small AS. MDHT [6] is an overlay of IP network that provides name resolution, i.e. resolving a name to an IP address. Different from DMAP, a content ID in MDHT is hashed into DHT keys of hierarchical DHT areas which reflect the underlying network topology (Access node, point of presence, AS etc.). MDHT allows content owners to control the level of insertion, e.g. the access to a company's document can be limited to only the company's network. Compared with DMAP, MDHT requires extra overhead to maintain DHT overlays and the latency of the query can be much larger if a query is propagated to DHT nodes at different levels.

Both DMAP and MDHT focus on resolving a content name/id to an IP address which is much like what DNS does. In content centric network, users need a way to retrieve content names. E-DMAP [8] is an improvement to DMAP by allowing users to retrieve content name/id with keywords. Content publishers not only insert GUID-to-IP mappings but also Keyword-to-GUID mappings. Like the GUID, a keyword is hashed to an IP address, and the Keyword-to-GUID mapping is stored in a server located within the AS which owns that IP address. Users can then retrieve matching GUIDs by sending keyword-based queries. Like DMAP, E-DMAP has a trade-off between storage overhead and query latency especially when a content is associated with many keywords. COPSS [7] solves the problem in a different way. It eliminates the requirement for users to know the content names but instead it includes a subscription sub-system. COPSS adds a "Subscription Table" in CCN routers and users can send a "Subscribe Packet" with a "Content Descriptor" (e.g. /sports/football) to a CCN router where the $routerID = hash(ContentDescriptor)$. When a new content is generated, the publisher adds a "Content Descriptor"

to the new content and pushes it to the corresponding CCN router which will then forward the content to subscribers. However, COPSS does not give a performance evaluation with realistic workload.

B. Mobility Support in CCN

Any mobility support design for CCN needs to address two scenarios: either a client or a publisher moves. A client move scenario is easy to handle as a client can resend an Interest Packet after it moves. Content caching feature can reduce the query latency if the requested content was previously retrieved by other users in the new location. However, if the content is not cached, the query latency can be as bad as sending a new query. [9] proposed a proactive neighbor caching so that the queried content will be forwarded to not only the client's current location but also other neighboring locations where the client is likely to visit. The proactive cache scheme makes seamless client mobility support possible but the author fails to explain how it can be integrated with the CCN architecture.

The publisher moving scenario is more complicated as the publisher needs to re-announce its content prefixes to update forwarding tables after the move which can be very costly. [16] solves this problem by introducing a "Home Repository" (HR). When a publisher moves to a new location, instead of announcing a new content prefix, it tells the HR to forward any interest packet to the new location. However, because the data packet in CCN is sent back to the user along the reverse path of the interest packet, such a solution forces the data packet to "detour" through the "HR", and hence increases the delivery delay. [10] solves the "detour" issue by adding a "Location Name" to the data packet so that after the user receives the first data packet, he can send the interest packet directly to the new location. This solution works well when a publisher moves just once, but if a publisher moves frequently, the user might experience even worse network delay as he cannot get synchronized with the newest location. For example, the user sends an Interest to retrieve a data content with the "current" location name he obtained, however, the publisher has already moved to a new location. In this case, the user needs to wait until a timeout and resends an Interest to the publisher's HR.

III. SCOM ARCHITECTURE

A. SCOM Overview

SCOM is an overlay content centric network built on top of IP network ([1] mentioned that CCN can be layered over anything, including IP itself). Each AS in the IP network has a main content resolution server (CRS) to store and forward user interest packet as in CCN and the interest packets are routed using underlying BGP routing protocol. CRSs from selected ASes also serves as the content name retrieval server (CNRS) where content publishers can publish content names/ids based on keywords as in E-DMAP [8].² A content publisher sends content advertisement packets (content names and keywords relevant to the contents) to its local CRS. The local CRS then generates relevant keyword-to-Name mappings to insert such information to the relevant CNRS whose identifier is the hash value of a keyword. Contents with the same keyword are aggregated into a single packet and the content with multiple keywords will be published to different CNRSes. The details of such insertion will be further described in Section 3.2. A

²In the paper we use content name and content id interchangeably since in that SCOM a content name can be either hierarchical or flat.

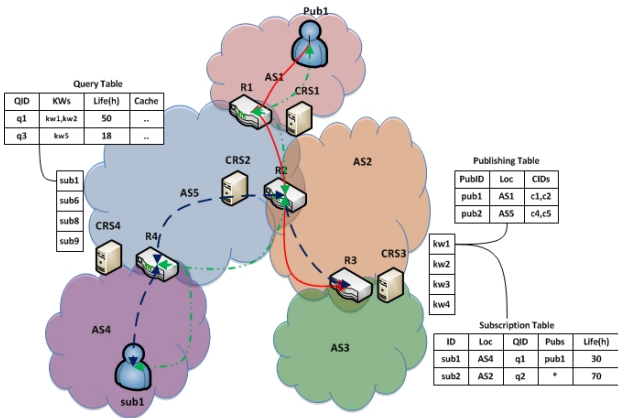


Fig. 1. Overview of SCOM. The solid red line (from Pub1 to CRS3) is the path for the publisher to publish content names with keyword; the blue dashed line (from sub1 to CRS3) is the path for the user to query the content names with keyword; the green dashed line (from sub1 to Pub1) is the path for the user to retrieve the data content.

SCOM user who does not know the names of the contents he is interested in submits a keyword-based query to its local CRS. The local CRS transforms this query into sub-queries (one for each keyword) and send these subqueries to relevant CNRSes. CNRSes that contain matching contents will send responses. Each CNRS also maintains a subscription table which serves for two purposes: (1) Subscribers will be informed of any new available contents based on their interested keywords. (2) When a publisher moves, subscribers will be informed of the change.

Figure 1 is an overview of SCOM architecture. As shown in the figure, the CRS of each AS is deployed near a BGP router. The five-step content retrieval is illustrated: (1) A publisher *Pub1* publishes a data content with the keyword *kw1* at CRS3 and (2) a user *sub1* sends a keyword-based query with the keyword *kw1* and subscription time 30 hours to its local CRS (CRS4) which forwards the query packet to CRS3. (3) The matching content name and its publisher information are sent in a response packet to the user and a new entry is inserted to the subscription table. (4) The user retrieves the data content by sending a content-based interest packet to the publisher *Pub1*. Finally, (5) the publisher sends the data content to the user.

B. Keyword-based Content Name Retrieval Sub-system

The keyword-based content name retrieval sub-system of SCOM is modified from E-DMAP [8]. In E-DMAP, a data content has a global unique content id (GUID) but can have multiple keywords associated with it. There are two proposed schemes for content insertion and retrieval. The first scheme, *Independent Search and Merge (ISM)*, hashes each keyword to an IP address (or CRS number in SCOM) and insert the (Keyword, GUID) to the AS/CRS which owns that IP address. Thus, for a content with m keywords, inserting keyword-content mapping requires m insert request messages. When querying contents with a set of n keywords, the local CRS will transform the query into n sub-queries each containing a single keyword and forward each sub-query to the keyword-hashed CRS. Sub-queries can be merged if the keyword in each query is hashed to the same CRS. On receiving a sub-query, the CRS replies with a list of GUIDs with a keyword which matches those specified in the query. When all sub-queries are received, the local CRS determines the intersection

Keyword	Publisher ID	Publisher Location	A list of Content Names (CID)
---------	--------------	--------------------	-------------------------------

Fig. 2. Fields in Publishing Packet

Query ID	Client ID	Client Location (AS)	Keyword(s)
A list of wanted Publisher (* represents all publishers)			Lifetime (0 for no subscription)

Fig. 3. Fields in Query Packet

of all GUID lists and returns the final list to the user. The second scheme, *Integrated Keywords Search (IKS)*, considers the combinations of keywords when publishing them. Thus, for a content with m keywords, there are $2^m - 1$ possible combinations containing at least one keyword and will result in $2^m - 1$ insert request messages. However, the benefit of *IKS* is that there will only be one query message irrespective of how many keywords there are in a query.

In SCOM, we adopt an *ISM* like scheme because a user typically issues queries with two to three keywords at most but a content may be published with more than 20 keywords. Using the *ISM* scheme allows us to have a lower cost in inserting keyword-content mapping entries.

SCOM improves E-DMAP in several aspects:

First, SCOM requires publishers to include publisher location (expressed as an endpoint identifier which CCN routers understand) in its publish announcement packet (Figure 2). In E-DMAP, when the user receives a list of content names, he needs to select the interested content name and send a second query to search the location of that content. In SCOM, the publisher location is included in the returned matching content list so the data content can be retrieved directly from the publisher by sending an Interest packet with a content name such as */location/content_name*.

Second, SCOM supports content subscription by including query lifetime within its Query Packet (Figure 3). A CRS, where contents are published, maintains a list of keyword entries, each consists of a keyword and two pointers: *publisher* and *subscriber* pointers. The *publisher pointer* points to a *publishing table* which contains the identifiers of all publishers which publish contents with that keyword, and the *subscriber pointer* points to a *subscription table* which contains the identifiers of all subscribers who are interested in contents with that keyword (Figure 1). Each CRS also maintains a list of clients residing in the AS who has sent a query and each client has a pointer to a *query table* of all queries he has sent. The lifetime in the query table is set to be longer than that in the subscribing table to support possible mobility. The "Lifetime" should be the real subscription time plus a mobility support time. The subscription function makes sure that a user can get the latest updates (e.g. a new content or the publisher's new location) from any publishers of interest in a timely manner.

Third, in SCOM, the keyword-to-name mappings can be configured to be published only at CRSes owned by Tier1/Tier2 ASes but not all CRSes as in E-DMAP. As known by networking researchers, Tier 1/Tier 2 ASes provide connectivity services to smaller lower tier ASes. Therefore, publishing contents at these "large" ASes instead of all ASes can effectively reduce query latency. Note that for ISPs that share peer links, the queries will still go through these peer links since SCOM does not change the existing route patterns. Furthermore, to minimize the inter AS traffic among Tier1/Tier2 ASes caused by SCOM, keywords-content mappings stored in other Tier1/Tier2 ASes can be replicated so

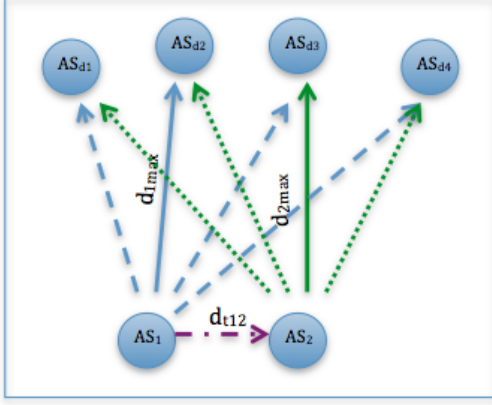


Fig. 4. A Client Moves after Issuing a Keyword-based Query

that keyword-based queries can be answered within the same Tier1/Tier2 AS.

What is more, SCOM users can query content names from publishers of interests by including a list of desirable publishers in their Query Packets (Figure3). With such information, users can avoid receiving content information from unwanted publishers even spams. Figure 1 illustrates how our keyword-based content retrieval sub-system works. The red solid line denotes that a publisher *Pub1* from AS1 publishes two content *c1* and *c2* with keyword *kw1* at CRS3. The blue dashed line denotes that a user *sub1* from AS4 sends a query packet with two keywords *kw1* and *kw2*. The query packet also specifies a subscription time of 10 hours and an interested publisher *Pub1*. Since both *kw1* and *kw2* are hashed to the same AS/CRS (CRS3), only one merged query message is sent.

C. Mobility Support in SCOM

SCOM provides mobility support for both keyword-based content name queries and data content retrievals.

1) *Supporting Keyword-based Queries from Mobile Client:* A client may move to a different AS after issuing a keyword based query but before getting a response back. An example of a realistic scenario is as follows: a user submits a keyword based query for video contents matching certain keywords as he is leaving his office and wishes to receive the answer using his cellular service. Figure 4 shows a scenario when a client who originally resides in old AS (AS1) moves after issuing a query containing four keywords which is distributed to four ASes (one per keyword). Let d_{1max} denotes the maximum one-way latency of all sub-queries from the old AS (AS1), d_{2max} denotes the maximum one-way latency of all sub-queries from the new AS (AS2), and d_{t12} denotes the one-way latency between old AS and new AS. SCOM provides three solutions to deal with this mobility problem:

- **Forwarding Solution:** The client requests the CRS in the new AS to fetch the query response from the CRS in the old AS.
- **Resending Solution:** The client resends the keyword-based query to the CRS in the new AS.
- **Adaptive:** The CRS in the new AS calculates the latency of the above two solutions (with input from user and underlying topology information) and adopts the one with a smaller latency.

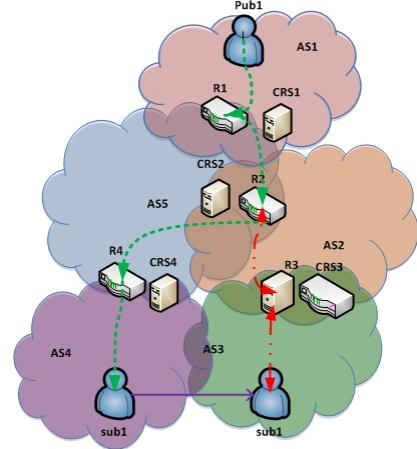


Fig. 5. Client Move Scenario: A client *sub1* moves from AS4 to AS3. The green dashed line denotes the original content delivery path; the red half-solid half-dashed line denotes the path of the remaining content segments queried and delivered from the "common point" R2/CRS2.

The latency for the **Forwarding** solution can be expressed as:

$$t_f = \max(2 * d_{1max} - t_{\Delta}, d_{t12} + t_h) + d_{t12} \quad (1)$$

(t_{Δ} is the interval between the time instant the client sent the query at the old AS and the time he moves ; t_h is the hand-off time.)

The latency for the **Resending** solution can be expressed as:

$$t_r = 2 * d_{2max} + t_h \quad (2)$$

The **Adaptive** solution will compare the latency of (1) and (2) to determine which solution should be used automatically.

Note that the proactive cache scheme [9] can be applied in addition to our solution. To incorporate this proactive cache feature into our solution, each client can include its frequently visiting locations in its interest packet, and request the relevant CRS to send responses to all visiting locations or to a particular location based on a particular time schedule. The price to pay for this feature is the added overhead of sending multiple response packets. We did not evaluate this added feature in this work but will include this evaluation in the near future.

2) *Mobility during Data Content Retrieval:* In SCOM, we assume a user can retrieve a large data content object e.g. a video stream using only one content name rather than via multiple content names, one for each video segment as described in CCN [1]. To deal with mobility during the content retrieval process, SCOM introduces a new Interest Packet for large-size content with multiple segments. Different from the traditional Interest Packet in CCN, the new Interest Packet is only consumed when the last segment of the content is received. The client can specify the "starting segment number" in the Interest Packet so that the content segments already received will not be resent.

SCOM supports both scenarios of having a client or a publisher moves during a content object retrieval. If a client moves during content retrieval, the client will send a new Interest Packet for the same content but set the "starting segment number" to be one plus the last received segment number. When the new Interest Packet reaches the first CRS on the previous content delivery path, since the content segments are cached, it can forward the remaining segments to the client's new location (as shown in Figure 5). The extra delay

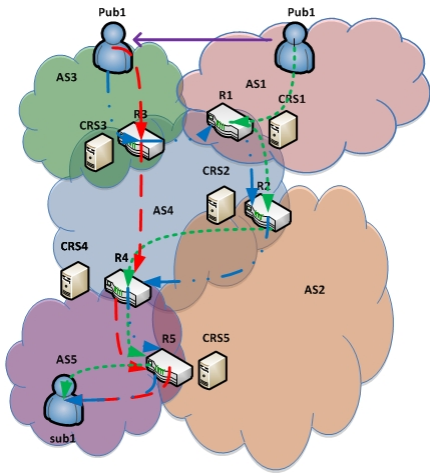


Fig. 6. Publisher Move Scenario: A publisher *Pub1* moves from AS1 to AS3. The green dashed line denotes the original content delivery path; the blue half-solid half-dashed line denotes the extended delivery path from new AS to old AS after the move; the red line denotes the new content delivery path through the “merging point” R4/CRS4.

caused by a client movement is calculated as:

$$t_c = 2 * d_{common} + t_h \quad (3)$$

(d_{common} is the one-way latency between the new AS and the common point; in Figure 5 it is the delay between R2/CRS2 and R3/CRS3)

When a publisher moves, the publisher first informs relevant CRSs of the move. Therefore, existing subscribers will be informed of this move so that they will generate new Interest Packets for unreceived or incomplete Contents. In addition, this publisher will include an optional “New Location” header in the first one or two data content packets of each request it sends at the new location. Any intermediate CRS along a content delivery path will determine if the data contents it receives from the publisher in his new location arrives via the best delivery path. If not, it will forward all non-expired interests to this publisher via the reverse of the best new delivery path. Thus, our solution allows data contents to be delivered using the shorter delivery path, avoiding the “triangular routing” or “Detour” path that other existing approaches use. For example, in Figure 6, R4/CRS4 will forward all non-expired interests to CRS2 and hence contents can be delivered using the shorter delivery path marked by the red-line.

IV. PERFORMANCE EVALUATION

In this section, we evaluate the SCOM architecture via simulation studies. We modified the simulator used in DMAP [5] and E-DMAP [8] by choosing a compact Internet topology with 6456 ASes and their IP prefix allocations extracted from the DIMES database [17]. The DIMES database also provides average end-to-end latencies between each pair of ASes and within each AS which are used in our simulation. The 6456 ASes include all major ASes and own 91% of total IP addresses announced by around 26,000 registered ASes today. We focus only on the AS-level network topology in our simulation so that we can produce simulation results within reasonable run times for more realistic network scenarios.

To evaluate the effectiveness of our mobility solution, we assume the mobile clients or publishers only move between wired AS (WiFi connection) and cellular AS within the same country. We believe most of the mobile scenarios in real life

incur moving within the same country rather than between different countries. We do not consider scenarios where users move across countries e.g. passengers flying on planes or riding on trains in Europe which move from one country to another. Thus, we extracted AS numbers (Non-tier1/tier2 ASes) of seven countries. Wikipedia provides a list of mobile operators in each country [13] and we searched the AS number for mobile providers in the seven countries with Hurricane BGP Toolkit [14]. Table I summarizes the number of wired and cellular ASes.

TABLE I
NUMBER OF WIRED AS AND CELLULAR AS

Country	# of Wired AS	# of Cellular AS
USA	1877	10
UK	155	2
China	70	10
Germany	140	3
Australia	124	3
Brazil	125	4
India	81	9

A. Workload

Like E-DMAP, our workload for keyword-based name query is generated from Flickr’s pictures but we use photo tags as keywords instead of the words from the titles of photos. We have crawled 250,440 photos as our contents and retrieved a total of 99,360 unique keywords from the tags of these photos. Table II lists some basic characteristics of our content/keyword workload. The first row is about the number of keywords for each content. The average number is 5.11, the maximum is 110, and the standard deviation is 5.59. The second row represents the statistical information about content frequency, which is the number of contents matching a given keyword.

TABLE II
NUMBER OF KEYWORDS PER CONTENT & CONTENT FREQUENCY

	Average	Maximum	Std Deviation
# of keywords	5.11	110	5.59
Content freq	12.87	4323	69.81

For content query, we assume webpages are published contents, and the page view number of a page represents the popularity of a content. We obtained a total of 1000 contents with their popularity from Google’s statistics of Top 1000 sites [15]. Generally, the content popularity follows a Zipf-based distribution.

In subsequent subsections, we first evaluate the query latency and storage overhead for non-mobile scenarios. Then, we present keyword-based query and content retrieval latency results for the mobile scenarios we studied.

B. Query Latency and Storage Overhead

1) *Latency of Keyword-based Content Name Query*: First, we compare SCOM’s solution of only inserting keyword to content name mappings at Tier1/Tier2 ASes to E-DMAP’s solution of inserting such mappings at all ASes. Wikipedia provides a full list of 17 Tier1 ASes and a small portion of 18 Tier2 ASes [11][12]. In our simulation, except those 35 listed Tier1/Tier2 ASes, we assume all other ASes with a degree larger than 50 are Tier2 ASes and thus we have a total of 212 Tier1/Tier2 ASes which is 3.28% of the total number of ASes. We assume the IP addresses of all users come from other lower tier ASes but content publishers are from a random AS for both Tier1/Tier2 and lower tier ASes.

Uniformly Distributed Queries: We simulate q -keyword queries with q in the range of 1,2,3,4,5. We generate 20,000

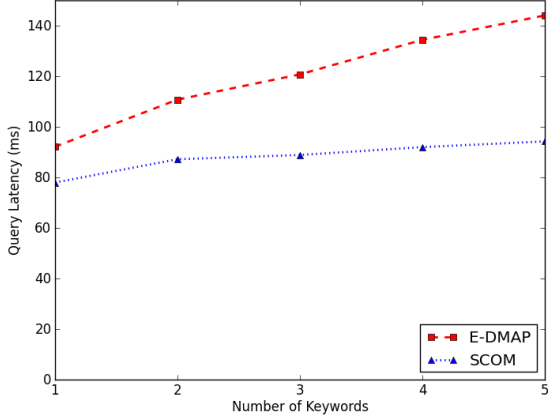


Fig. 7. Average Query Latency for Different Number of Keywords.

queries for each q assuming each query is sent from a random non Tier1/Tier2 AS. The keyword(s) for each query is randomly selected from the keywords pool and we make sure that at least one content exists for each query. Table III presents the statistics of the number of matching contents for each type of 20,000 queries. As the number of keywords increases, there are fewer matched contents.

TABLE III
NUMBER OF MATCHING CONTENTS

# of Keywords	Average	Maximum	Std Deviation
1	391.39	4323	786.82
2	19.03	848	47.44
3	9.59	233	9.19
4	8.84	77	6.54
5	8.83	68	6.36

Our simulation result is shown in Figure 7. When only Tier1/Tier2 ASes are used to store keyword to content mappings, the average query latency is reduced by 15ms (1 keyword) to 50ms (5 keywords). Meanwhile, as the number of keywords increases, SCOM only shows a small increase in the query latency while the latency incurred using the E-DMAP solution increases quickly with increasing number of keywords. As shown in the Figure, the average latency for queries with 5 keywords is only 16ms larger than queries with 1 keyword which is much smaller than the 52ms increase seen using the E-DMAP solution.

Non-uniformly distributed Queries Next, we study non-uniformly distributed queries since we are interested in understanding whether non-uniformly distributed queries result in similar latency results. Instead of assuming each keyword-based query is sent from a random AS, we assume that the AS owning more IP addresses will receive more queries since it is likely to have more users. In this study, we study the latency performance for each country. We assume 50% of the total queries come from wired ASes and 50% come from cellular ASes in each country. For each country, we generate a total of 20,000 queries. All queries have 5 random keywords and we make sure that at least one content can be matched for each query.

The average query latency for each country is summarized in Table IV. Typically, the retrieval latency for wired ASes is smaller than their cellular counterparts except for the case in India where they are comparable. Furthermore, two scenarios stand out: the large cellular latency for Brazil and Australia.

The Brazilian Cellular ASes has an average query latency of over 700ms. We investigated and found that there are two cellular ASes in Brazil: AS26615 (Tim Cellular S.A.) and AS26599 (Vivo S.A.) with a very large first-hop latency (latency to a neighboring AS). For the Australia scenario, AS4804 (Singtel Optus Pty) owns more than 70% of mobile IP addresses in the country and its minimum first-hop latency is 284ms. This leads to a much larger average query latency for the cellular ASes in Brazil and Australia entries seen in Table IV.

TABLE IV
QUERY LATENCY OF WIRED AS AND CELLULAR AS WITHOUT MOBILITY

Country	Wired AS (ms)	Cellular AS (ms)
USA	71.76	85.99
UK	71.52	59.73
China	79.00	92.64
Germany	64.09	102.29
Australia	321.29	577.57
Brazil	130.61	717.18
India	99.12	97.69

2) *Latency of Content Retrievals*: In this subsection, we study how different content caching schemes impact the latency of content retrievals. The following 3 schemes are evaluated, namely (i) all contents are cached, (ii) top 30 most popular contents are cached, and (iii) no caching. Furthermore, we also explore two caching strategies: (a) a retrieved content is cached at all ASes along its delivery path, (b) a retrieved content is only cached at Tier1/Tier2 ASes along its delivery path.

In this simulation study, we assume 1000 content objects are published and each object is published at a random AS (all tiers) in US. Each content has a size that can fit into one data packet. The popularity of the 1000 contents follow a Zipf-based distribution. Then, we generate 100,000 queries for these 1000 contents, assuming each query is being sent from a random AS among non-Tier1/Tier 2 ASes in US. The most popular content will be requested more frequently according to the Zipf distribution.

Our results are tabulated in Table V. In Table V, "All Cached" means all retrieved contents are cached while "Top 30 cached" means only the top 30 most popular contents are cached; "All CRS" means the contents are cached on all ASes along the delivery paths while "Tier1/Tier2" means they are only cached if the ASes along the delivery path are Tier1/Tier2 ASes. The results show that (1) content caching can significantly reduce the overall content query latency; (2) caching content at all ASes results in a much smaller query latency than caching at only Tier1/Tier2 ASes; (3) caching only the popular content can reduce the latency from 52.3ms (without caching) to 14.5ms. Caching all contents reduce this latency further to 8ms.

TABLE V
CONTENT QUERY LATENCY WITH DIFFERENT CACHE SCHEMES AND CACHE LOCATIONS

Scheme	Location	Avg Latency	% from Cache
All Cached	All CRS	8.01ms	94.49%
Top 30 Cached	All CRS	14.51ms	78.55%
All Cached	Tier1/Tier2	38.91ms	40.3%
Top 30 Cached	Tier1/Tier2	43.32ms	26.75%
No Cache	No Cache	52.31ms	0%

C. Storage Analysis

Here, we analyze the storage requirement for our SCOM solution. We first consider the storage requirement for

storing keyword-to-content mappings. Since the number of Tier1/Tier2 ASes only constitutes 3.28% of the total number of ASes, one possible concern for SCOM is whether SCOM will cause too much storage overhead on Tier1/Tier2 ASes. In DMAP or E-DMAP, a keyword or GUID is hashed to an IP address and therefore the AS with more IP addresses are likely to incur larger storage cost. For the network topology used in our simulation studies, the total number of announced IP addresses is about 2.5 billion while the Tier1/Tier2 ASes own about 1.07 billion which is 42.6% of the total announced IP addresses. This means, on the average, each Tier1/Tier2 AS's storage overhead with SCOM will be only 2.34 times of the storage overhead in E-DMAP. In addition, using the same assumption as the DMAP paper where such mappings are stored in every AS, each AS only requires 173Mbits with 5 billion mappings (with each mapping requiring 352 bits). Even with 2.34 times more storage, the storage cost is still modest. Therefore, our SCOM solution is still practical and useful for it significantly reduces the query latency. To further reduce the storage overhead on a single CRS, a Tier1/Tier2 AS may deploy multiple CRSes using a Distributed Hash Table (DHT). When the master CRS receives an insertion request, a second hash function is applied to the keyword to determine which specific CRS to store the new mapping.

Next, we consider the storage requirement for content caching. Table VI shows the summary of content caching for each of the cache schemes and cache locations we explored in the previous subsection. We can draw the following conclusions from the result. (1) If all content are cached, on the average, only 2.7% or 6.3% of the total 1000 contents need to be cached. Even in the worst case only about 50% of the 1000 contents need to be cached. Simple cache management scheme e.g. Least Recently Used can be used if there is insufficient storage for all contents that need to be cached. (2) If only popular contents (e.g. Top 30%) are cached, the worst case storage requirement for each CRS is very small (like only 10% of total published contents. Our results in Table V show that caching only top 30% popular contents can provide good content retrieval latency. Thus, we believe that the storage requirement for content caching is a manageable engineering design issue.

TABLE VI
STORAGE OVERHEAD WITH DIFFERENT CACHE SCHEMES AND CACHE LOCATIONS (# DENOTES THE NUMBER OF CACHED CONTENT IN AN AS)

Scheme	Location	Cached AS	Avg #	Max #
All Cached	All CRS	1368	27.1	540
Top 30 Cached	All CRS	944	2.22	10
All Cached	Tier1/Tier2	119	63.1	509
Top 30 Cached	Tier1/Tier2	77	2.92	8

D. Mobility Result

Here, we study how SCOM mobility support solution performs in two mobility scenarios: (a) when mobile users move from wired to cellular services and (b) they move from cellular to wired services.

1) *Mobility Support for Keyword-based Queries:* In this study, we assume each keyword-based query contains 5 random keywords, and we make sure that the keywords are chosen such that at least one content can be matched for each query. We assume that each user that generates a query moves within the same country. Thus, we generate 20,000 such queries for each country: 10,000 queries, each of which started from a randomly chosen wired AS and moved to a cellular AS,

another 10,000 queries with each which started from a cellular AS and moved to a wired AS. As in the per-country query latency simulation study, we assume that the AS owning more IP addresses are more likely to receive more queries. In the simulation, we set the hand-off time t_h to 100ms.

We simulated two scenarios:

Scenario 1: The users move immediately after sending queries. In this case, the t_Δ in equation (1) is 0.

Scenario 2: The users move after d_{1max} . In this case, the t_Δ in equation (1) is d_{1max} .

The simulation result for Scenario 1 is summarized in Table VII.

TABLE VII
THE PERCENTAGE OF QUERIES THAT **FORWARD** SOLUTION PERFORMS BETTER THAN **RESENDING** SOLUTION ($t_\Delta = 0$)

Country	Move from Wired AS	Move from Cellular AS
USA	79.12%	38.01%
UK	84.47%	99.57%
China	74.34%	67.78%
Germany	93.85%	47.98%
Australia	70.72%	40.66%
Brazil	89.05%	13.85%
India	59.75%	53.70%

Comparing the latency results we obtained for Scenarios 1&2 (whether users move immediately or move after d_{1max} , we found no obvious difference. This is because d_{1max} is mostly smaller than t_h , i.e. 100ms (referring to Table IV), and hence the equation (1) can be simplified to:

$$t_f = 2 * d_{t12} + t_h \quad (4)$$

for most countries except Australia and Brazil. This means that if d_{t12} is smaller than d_{2max} , the **Forward** solution is better; otherwise, the **Resending** solution has a better performance.

TABLE VIII
AVERAGE QUERY LATENCY WITH MOBILITY WHEN MOVE FROM WIRED AS TO CELLULAR AS AND $t_\Delta = 0$ (MS)

Country	Forward	Resend	Adaptive
USA	175.40	185.99	169.93
UK	132.12	159.73	127.24
China	181.68	192.63	176.13
Germany	182.51	202.28	179.69
Australia	666.01	677.57	631.52
Brazil	780.85	817.17	770.64
India	180.90	197.69	170.92

TABLE IX
AVERAGE QUERY LATENCY WITH MOBILITY WHEN MOVE FROM CELLULAR AS TO WIRED AS AND $t_\Delta = 0$ (MS)

Country	Forward	Resend	Adaptive
USA	175.06	171.76	166.74
UK	131.79	171.52	131.77
China	183.00	178.99	166.64
Germany	182.89	164.09	152.89
Australia	763.98	421.29	380.63
Brazil	1051.26	230.61	223.83
India	179.96	199.11	172.26

From the result, we also found that when a user moves, no single solution (whether **Forward** or **Resending**) is optimal. For example, when Brazilian users moves from a wired AS to a cellular AS, the **Forward** solution performs better than the **Resending** solution for most of the queries because of a large first hop latency associated with a cellular AS; while when the users moves from a cellular AS back to a wired AS, **Resending** solution has a better performance. Therefore, the **Adaptive** solution introduced in Section 3 should be deployed to achieve

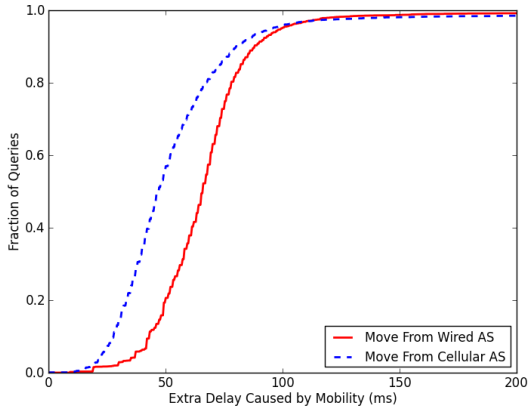


Fig. 8. Extra Delay Caused by Client Movement.

the best performance. Table VIII and Table IX summarize the average query latency for all 3 solutions when t_{Δ} is 0, i.e. users move immediately after sending query, for both mobility scenarios.

2) *Mobility Support for Content Retrieval*: In this subsection, we explore how much extra delay our SCOM mobility support solution incurs for scenarios where a client or a provider moves during a content retrieval process. We also want to compare our solution with the DETOUR solution.

Client Moving Scenario: Here we assume a content provider can be from any AS and focus on the scenario where the clients come from US ASes only. We generate 10,000 content queries each for Scenarios 1 & 2 (i.e. moving from wired to cellular and vice versa). Figure 8 plots the cumulative distribution function (CDF) of the extra delay (hand-off time not included) caused by a client movement using SCOM. The extra delay when a client moves from a cellular AS to wired AS is generally smaller than the delay when it moves from a wired AS to cellular AS or vice versa, more than 95% of content retrievals will have an extra delay of less than 100ms plus the hand-off time.

Provider Moving Scenario: Here, we also consider both publishers and users come from US ASes only. Furthermore, we assume users send content retrieval requests only from a wired non-Tier1/Tier2 AS, and publishers move from a wired AS to a cellular AS or vice versa. We compare the delay of SCOM's solution and "Detour" solution introduced in [16]. The simulation results are shown in Figure 9. As shown in the figure, with SCOM's mobility solution, almost 80% of queries have a delay less than 100ms plus the hand-off time. While the "Detour" method causes a delay of larger than 150ms (plus hand-off time) for more than 80% of content retrieval requests irrespective of whether the publisher moves from a Wired AS or Cellular AS.

V. CONCLUSIONS AND FUTURE WORK

In this paper, we present SCOM, a scalable content centric network with mobility support. As in E-DMAP, SCOM provides keyword-based queries to retrieve contents of interest to any user but SCOM achieves smaller query response times than E-DMAP by storing keyword to content mappings only in Tier1/Tier2 ASes. In addition, SCOM provides mobility support for both keyword-based queries, and content retrieval. Our large-scale simulation studies show that our mobility

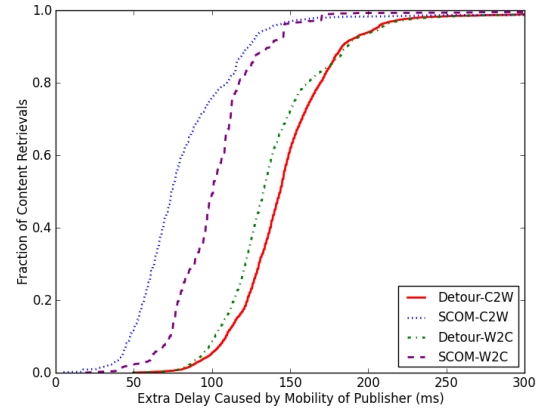


Fig. 9. Extra Delay Caused by Publisher Movement. The solid lines denote the publishers moving from Cellular AS to Wired AS (C2W). The dashed lines denote the publishers moving from Wired AS to Cellular AS (W2C).

support feature provides smaller content delivery latency than an existing solution.

Our work in this paper still has some limitations. Our simulation studies assume that the keywords included in queries are randomly chosen. We hope to find some realistic mobile queries datasets e.g. from mobile search engines which allow us to generate queries that may match a Zipf-based distribution. In addition, we hope to investigate the effect of having limited caches sizes on content retrieval latency and design algorithms to identify important contents to cache at strategic locations. Furthermore, we will address security design for our framework in the near future.

VI. ACKNOWLEDGMENTS

We wish to thank Prof. Bo Sheng for sharing the enhanced DMAP (E-DMAP) simulator.

REFERENCES

- [1] V. Jacobson et al., "Networking named content," in *CoNEXT*, 2009.
- [2] J. F. Gantz et al., "IDC - The Expanding Digital Universe: A Forecast of Worldwide Information Growth Through 2010," *Technical report*, 2007.
- [3] V. Jacobson, "A New Way to Look at Networking," *Google TechTalk*, Aug. 2006.
- [4] S. Bicheno, "Global Smartphone Installed Base Forecast by Operating System for 88 Countries: 2007 to 2017," *Strategy Analytics Wireless Smartphone Strategies (WSS) service*, 2012.
- [5] T. Vu et al., "DMap: A Shared Hosting Scheme for Dynamic Identifier to Locator Mappings in the Global Internet," in *IEEE ICDCS*, Jun. 2012.
- [6] M. D'Ambrosio et al., "MDHT: A Hierarchical Name Resolution Service for Information-Centric Networks," in *Proc. ACM SIGCOMM Wksp. Information-Centric Networking*, 2011.
- [7] J. Chen, M. Arumathurai, L. Jiao, X. Fu, and K. K. Ramakrishnan, "COPSS: An Efficient Content Oriented Publish/Subscribe System," in *ANCS*, 2011.
- [8] Y. Mao, B. Sheng, M. Chuah, "Scalable Keyword-Based Data Retrievals in Future Content Centric Networks," in *IEEE MSN*, Dec. 2012.
- [9] X. Vasilakos et al., "Proactive selective neighbor caching for enhancing mobility support in information-centric networks," in *ICN 2012*.
- [10] F. Hermans et al., "Global Source Mobility in the Content-Centric Networking Architecture," in *NoM 2012*.
- [11] http://en.wikipedia.org/wiki/Tier_1_network.
- [12] http://en.wikipedia.org/wiki/Tier_2_network.
- [13] https://en.wikipedia.org/wiki/Mobile_country_code.
- [14] Hurricane Electric BGP Toolkit, <http://bgp.he.net>.
- [15] The 1000 most-visited sites on the web, www.google.com/adplanner/static/top1000/
- [16] D. Kim et al., "Mobility Support in Content Centric Networks," in *The 2nd ACM SIGCOMM Workshop on Information-Centric Networking*, Helsinki, Finland, Aug. 2012.
- [17] O. Saleh and M. Hefeeda, Modeling and caching of peer-to-peer traffic, in *Proceedings of the Proceedings of the 2006 IEEE International Conference on Network Protocols*. Washington, DC, USA, Nov. 2006.