A Survey on Content-Oriented Networking for Efficient Content Delivery

Jaeyoung Choi, Jinyoung Han, Eunsang Cho, Ted "Taekyoung" Kwon, and Yanghee Choi, Seoul National University

ABSTRACT

As multimedia contents become increasingly dominant and voluminous, the current Internet architecture will reveal its inefficiency in delivering time-sensitive multimedia traffic. To address this issue, there have been studies on content-oriented networking (CON) by decoupling contents from hosts at the networking level. In this article, we present a comprehensive survey on content naming and name-based routing, and discuss further research issues in CON. We also quantitatively compare CON routing proposals, and evaluate the impact of the publish/subscribe paradigm and in-network caching.

INTRODUCTION

Video traffic has been and will be increasingly prevalent in the Internet. Some video content providers (CPs, e.g., YouTube and Hulu) have even begun to provide high-definition video streaming services. As the bit rate of multimedia traffic increases, the TCP/IP architecture may reveal its inefficiency in delivering time-sensitive multimedia traffic.

Another important multimedia application is multicasting/broadcasting over IP networks (e.g., IPTV). However, the endpoint-based Internet is not suitable for multicast/broadcast due to issues including multicast address assignment and complex group management. Such ineptness leads to limited deployment and complicated multicast frameworks.

At present, many voluminous contents (most of which are multimedia [1]) are delivered to numerous users by peer-to-peer (P2P) systems such as BitTorrent. In BitTorrent, for each content file there is a tracker, which informs a new peer of other peers. A peer exchanges missing parts (called chunks) of the content file with other peers. However, from a networking perspective, the delivery performance of BitTorrent is inefficient since a peer can download chunks only from a small subset of peers who may be distantly located. In general, P2P systems have limited information on peers downloading the same content and the network topology among them (e.g. proximity).

In these "content-oriented" applications/services, an end user cares not about hosts, but about contents. However, the current Internet relies on the host-to-host communication model. This mismatch leads to application/service-specific solutions, which may be costly and/or inefficient. Two representative examples are:

- Web caches and content delivery networks (CDNs) transparently redirect web clients to a nearby copy of the content file.
- P2P systems enable users to search and retrieve the content file.

To address the above mismatch, there have been studies on content-oriented networking (CON)[1] (e.g. [2–4]). They strive to redesign the current Internet architecture to accommodate content-oriented applications and services efficiently and scalably. The essence of CON lies in decoupling contents from hosts (or their locations) not at the application level, but at the network level. Note that these proposals also solve or mitigate other Internet problems such as mobility and security.

We argue that the new CON paradigm will:

- Free application/service developers from reinventing application-specific delivery mechanisms
- Provide scalable and efficient delivery of requested contents (e.g., by supporting multicast/broadcast/anycast naturally)

In this article, we classify the prior studies on CON, discuss their technical issues, and identify further research topics. After demonstrating the performance of CON proposals, we conclude this article.

CONTENT-ORIENTED NETWORKING

A CON architecture can be characterized by four main building blocks:

- How to name the contents
- How to locate the contents (routing)
- How to deliver/disseminate the contents
- How to cache the contents "in" the network

[1] *Many studies use their own terminology such as* data-oriented, content-centric, *and* content-based. *In this article, we use* content-oriented *as a generic term.*

Persistence refers to a property that once a content name is given, people would like to access the content file with the name as long as possible. For example, if the ownership of a content file is changed, its name becomes misleading with the above naming.

There are relatively many studies on the first two components, to be classified in this section. The last two topics need more investigation under CON environments, which will be discussed later. Before presenting the taxonomy, let us discuss common characteristics in CON proposals [2–4].

A CON has three characteristics distinct from IP networking. First, a CON node[2] performs routing by content names, not by (host) locators. This means two radical changes:

- Identifying hosts is replaced by identifying contents.
- The location of a content file is independent of its name.

An IP address has both the identifier and locator roles; hence, IP networking has problems like mobility. By splitting these roles, CON has location independence in content naming and routing, and is free from mobility and multihoming problems.

Second, the publish/subscribe paradigm is the main communication model in CON: a content source announces (or *publishes*) a content file, while a user requests (or *subscribes* to) the content file. In IP networking, a user should know which source holds the content file of interest (spatial coupling), and the two hosts should be associated throughout the delivery (temporal coupling) [5]. However, with the publish/subscribe paradigm, we can decouple the content generation and consumption in time and space, so contents are delivered efficiently and scalably (e.g., multicast/anycast).

Third, the authenticity of contents can easily be verified by leveraging public key cryptography. In IP networking, a host address seen by a user is irrelevant to its content name, which results in phishing and pharming attacks. For content authentication in CON, either a self-certifying content name [2, 4] or a signature in a packet [3] is used. We skip the security-related explanations here; see [2, 3] for details.

CONTENT NAMING

We classify naming schemes in CON into three categories: hierarchical, flat, and attribute-based.

Hierarchical Naming — CCN [3] and TRIAD [6] introduce a hierarchical structure to name a content file. Even though it is not mandatory, a content file is often named by an identifier[3] like a web URL (e.g. /www.acme.com/main/logo.jpg), where / is the delimiter between components of a name. Thus the naming mechanism in the proposals can be compatible with the current URL-based applications/services, which may imply a lower deployment hurdle. Its hierarchical nature can help mitigate the routing scalability issue since routing entries for contents might be aggregated. For instance, if all the contents whose names start with www.acme.com are stored in a single host, we need a single routing entry (to the host) for these contents. However, as content files are replicated at multiple places, the degree of aggregation decreases. For instance, if popular contents are increasingly cached by in-network caching, the corresponding routing entries that have been aggregated should be split accordingly. Note that components in a hierarchical name (e.g., www.acme.com and logo.jpg) have semantics, which prohibits *persistent* naming. Persistence refers to a property that once a content name is given, people would like to access the content file with the name as long as possible. For example, if the ownership of a content file is changed, its name becomes misleading with the above naming.

Flat Naming — To avoid the above shortcomings, DONA [2] and PSIRP [4] employ flat and self-certifying names by defining a content identifier[4] as a cryptographic hash of a public key. Due to its flatness (i.e., a name is a random looking series of bits with no semantics), persistence and uniqueness are achieved. However, flat naming aggravates the routing scalability problem due to no possibility of aggregation. As flat names are not human-readable, an additional "resolution" between (application-level) human-readable names and content names may be needed.

Attribute-Based Naming — CBCB [7] identifies contents with a set of attribute-value pairs (AVPs). Since a user specifies her interests with a conjunction and disjunction of AVPs, a CON node can locate eligible contents by comparing the interest with advertised AVPs from content sources. It can facilitate *in-network searching* (and routing), which is performed by external searching engines in the current Internet. However, it has drawbacks such as:

- An AVP may not be unique or well defined.
- The semantics of AVPs may be ambiguous.
- The number of possible AVPs can be huge.

NAME-BASED ROUTING

CON should be able to locate a content file based on its name, which is called name-based routing. The prior studies can be classified depending on whether there is a systematic structure in maintaining routing tables of CON nodes.

Unstructured Routing — Like IP routing, this approach assumes no structure to maintain routing tables; hence, the routing advertisement (for contents) is mainly performed based on flooding. CCN suggests inheriting IP routing, and thus has IP compatibility to a certain degree. Therefore, CCN might be deployed incrementally with current IP networking. CCN just replaces network prefixes (in IP routing) with content identifiers, so the modification of IP routing protocols and systems may not be significant. Just as network prefixes are aggregatable in IP routing, so are hierarchical content identifiers in CCN routing. However, as a content file is increasingly replicated or moved, the level of aggregation diminishes. Moreover, the control traffic overhead (i.e., the volume of announcement messages whenever a content file is created, replicated, or deleted) would be huge.

Structured Routing — Two structures have been proposed: a tree and a distributed hash table (DHT). DONA is the most representative tree-based routing scheme. Routers in DONA form a hierarchical tree, and each router main-

[2] *A* CON node *refers to a node that performs CON functionalities like content routing and caching, while a* node *may indicate an IP router as well as a CON node.*

[3] *In this article, a name and an identifier are used interchangeably.*

4 *They also add a label (which has semantics) into a content identifier; however, the label can be interpreted only by end-points (i.e. publishers and subscribers), not by in-network nodes.*

	Naming	Naming advantages	Routing structure	Routing scalability	Control overheads
CCN	Hierarchical	Aggregatable, IP compatible	Unstructured	N (best) C (worst)	High
DONA	Flat	Persistent	Structured (tree)	C	Low
PSIRP	Flat	Persistent	Structured (hierarchical DHT)	$logC$	Low
TRIAD	Hierarchical	Aggregatable, IP compatible	Unstructured	N	High
CBCB	(attribute, value) pairs	In-network searching	Source-based multicast tree	2^A	High

Table 1. *Taxonomy of CON proposals in terms of naming and routing criteria is summarized. Note that the routing scalability of each proposal is in proportion to either* N, C, A, *or its logarithm/exponential. Here,* N, C, *and* A *are the numbers of publisher nodes, contents, and attributes in the entire network, respectively.*

tains the routing information of all the contents published in its descendant routers. Thus, whenever a content file is newly published, replicated, or removed, the announcement will be propagated up along the tree until it encounters a router with the corresponding routing entry. This approach imposes an increasing routing burden as the level of a router becomes higher. The root router should have the routing information of all the contents in the network. Since DONA employs non-aggregatable content names, this scalability problem is severe.

On the other hand, PSIRP [4] adopts hierarchical DHTs [8]. The flatness of a DHT imposes an equal and scalable routing burden among routers. If the number of contents is C, each router should have $\log(C)$ routing entries. However, the DHT is constructed by random and uniform placement of routers, and thus typically exhibits a few times longer paths than a tree that can exploit the information of network topology. Also, the flatness of a DHT often requires forwarding traffic in a direction that violates the provider-customer relation among ISPs; for instance, a customer ISP does not want to receive a packet from its provider ISP if the destination is not located inside.

Table 1 compares the CON studies with a focus on naming and routing characteristics. As CCN and TRIAD adopt hierarchical naming, content names are aggregatable and IP compatible. However, their flooding-based (i.e., unstructured) routing incurs significant control traffic. If all the contents of the same publisher are stored in a single host, their routing entries are aggregated into a single one; thus, the routing scalability is proportional to the number of publisher nodes. TRIAD considers only this case. In CCN, however, the contents can be replicated, which may split the aggregated routing entries; in the worst case, the routing burden becomes on the order of the number of contents.

DONA and PSIRP employ flat names for persistence. As they have systematic routing structures, the control traffic will not be substantial. In DONA, the root node of the tree should have the routing information of all the contents. Meanwhile, the DHT structure of PSIRP levies the routing burden of a logarithm of the number of contents on every node.

With AVP-based content names, CBCB enables in-network searching and establishes source-based multicast trees to deliver contents between publishers and subscribers. As each attribute can be selected or not in a search query, the number of routing entries of a router may be proportional to 2^A where A is the total number of attributes. Furthermore, its control overhead will be high since each new query may have to be flooded across the network.

Further Issues in CON

Multisource Dissemination

The current Internet architecture is designed with point-to-point connectivity since early stage applications rely on packet exchanges between two hosts. As the Internet becomes increasingly popular, however, new applications requiring different connectivities have emerged: one-to-many ($1{:}N$) and many-to-many ($M{:}N$).

$1{:}N$ connectivity represents content dissemination from a single source to multiple recipients; representative applications are online streaming and IPTV services. To support such applications with the point-to-point TCP/IP architecture, the Internet Engineering Task Force (IETF) standardized the IP multicasting framework, which is deployed in limited situations like a separate network for intradomain IPTV services. Compared to IP multicasting, CON accommodates $1{:}N$ connectivity naturally by the publish-subscribe paradigm in terms of content naming and group management [2, 4]. However, its link efficiency is not different from IP multicasting. Thus, let us focus on $M{:}N$ connectivity.

$M{:}N$ connectivity takes place among multiple sources and multiple recipients. There are two kinds of $M{:}N$ connectivity applications:

- M instances of $1{:}N$ connectivity (e.g., videoconference).
- M sources disseminate different parts of a content file to N recipients.

The current Internet architecture is designed with point-to-point connectivity since early stage applications rely on packet exchanges between two hosts. As the Internet becomes increasingly popular, however, new applications requiring different connectivities have emerged: one-to-many and many-to-many.

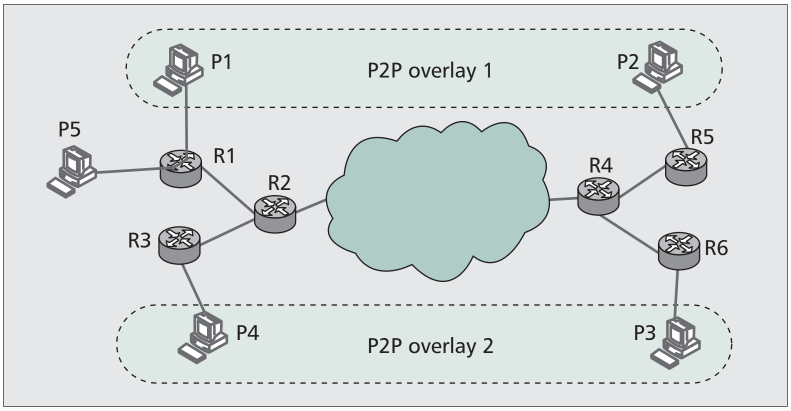

Figure 1. *Dissemination of the same file in the two separate overlays will be inefficient since peers are distantly located.*

We focus on the latter case, whose representative applications are P2P systems like BitTorrent. Another application is multi-user online gaming in which different but partially overlapping game data are transmitted to players.

Substantiating M:N connectivity requires application/service-specific overlays or relay mechanisms in the current Internet. However, CON can disseminate[5] contents more efficiently at the network level by spatial decoupling of the publish/subscribe paradigm and content awareness at network nodes.

Figure 1 illustrates how inefficiently a content file is distributed by current P2P operations. Suppose two P2P overlays are formed, and the peers in both overlays wish to download the same file. Unfortunately, peers in each overlay are distant; hence, the throughput is poor, which happens frequently in reality. This is because P2P systems are application-level solutions that cannot exploit network topology information. In contrast, CON can efficiently disseminate a content file among subscribers since CON nodes (R2 and R4) will help them download the content file also from the other overlay.

Disseminating a content file from multiple sources is tightly coupled with name-based routing. That is, in order to exploit multiple sources in disseminating the same content in CON, each CON node may have to keep track of individual sources of the same content (e.g., CCN and DONA). In this case, a CON node can seek to retrieve different parts of the requested content in parallel from multiple sources to expedite dissemination. To the best of our knowledge, there is no prior study on this *multisource dissemination* at a networking level. Depending on round-trip times and traffic dynamics of the path to each source, the CON node should dynamically decide/adjust which part of the content file is to be received from each source.

Another relevant issue is what routing information should be stored and advertised by each CON node for multiple sources of the same content. For instance, suppose a CON node receives routing advertisements from two sources of the same content, and learns that one source is close and the other source is very far. It may not be useful to announce both of the sources since retrieving data from the far source would be inefficient. The more sources of the same content the CON node learns of, the more selectively it may have to propagate the routing information of the sources. How to design a routing protocol for multiple sources, both intradomain and interdomain, is a crucial issue.

In-Network Caching

The advantages of in-network caching in an ISP may be twofold:
- To reduce the incoming traffic from neighbor ISPs to lower the traffic load on its cross-ISP links[6] (and hence its expense for transport link capacity)
- To improve the delay/throughput performance by placing the contents closer to their users

The latter has the same rationale as CDNs. The usefulness of caching is already proven by the commercial success of CDNs. In-network caching is also attractive to content providers (CPs) since it can mitigate the capital expense on their content servers. We believe an ISP with in-network caching capability can also offer CDN-like businesses to CPs if the majority of potential subscribers to the CPs are connected to the ISP (e.g., [9]).

Considering the above incentives, it is viable to cache popular content files in CON nodes (or their corresponding storage servers); other studies (e.g., [2–4]) also suggest introducing in-network caching. While there are well studied caching policies such as least recently used and least frequently used replacement policies at individual nodes, the performance of in-network caching can be further improved by coordinating multiple CON nodes in a distributed fashion. There have been recent studies on *distributed caching* (e.g., how to locate caching points [9] and how to cache contents [10]). However, as they assume IP networking, their work is limited in that
- Only a single source (or cache) delivers the content file to a subscriber.
- Limited topologies (e.g. tree) or places (e.g. point of presence) are taken into account.

Thus, we need to reformulate the distributed caching problem in CON environments; for instance, multisource dissemination and general network settings may have to be considered.

Another (maybe smaller) topic is how to design a signaling protocol among CON nodes to support distributed caching. For instance, a routing protocol may have to be extended to facilitate coordinated caching among CON nodes without significant signaling traffic overhead. The major design issue would be that the more frequently the content files are replaced in a cache, the more routing information may have to be advertised to enhance the network-wide caching performance.

Performance Evaluation

Using ns-2, we evaluate:
- The effect of routing structures on the resolution time (or delay) to locate a content file
- How much traffic load can be reduced by in-network caching network-wide

[5] *Even though CON does not care whether a single or multiple recipients subscribe to a particular content file, we implicitly use "deliver" for 1:1 connectivity and "disseminate" for 1:*N *and* M:N *connectivities.*

[6] *By reducing incoming traffic from its provider ISP, its connection fee may also be lowered.*

In addition to tree and DHT structures, we introduce a new routing structure: two-tier. We first explain how the network topology is constructed and content requests are generated for experiments. Then we describe the two-tier routing structure, followed by the simulation results.

End hosts, which publish or subscribe to content files, are collocated with CON nodes for simplicity. Using GT-ITM models, we generate a physical transit-stub topology, where a single transit domain connects 10 stub domains. There are 310 nodes total whose links have 10 Gb/s bandwidth capacity, among which 100 nodes in the stub domains are selected as CON nodes. There are 10 nodes in the transit domain, which do not serve as CON nodes since they normally have higher traffic loads (typically with higher link capacities) and hence may not be suitable for CON operations (e.g., in-network caching).

Reflecting the Internet traffic statistics [1], four types of content files are published in the end hosts: video, audio, software, and web contents, with 68, 9, 9, and 14 percent in terms of the content volume, respectively. For each content type, 1000 content files are published and evenly distributed among the end hosts. The popularity (or request probability) of a particular content file is determined by the Zipf distribution whose parameter is set to 1.0. The arrival rate of subscriptions (or request rate) is set to 0.5 s^{-1}. For details like the subtypes of each content type and the file size of each content subtype, see [1].

Two-Tier

Through our earlier experiments, we made the following observations:
- As a tree can be formulated with network topology information (e.g., hop count between nodes), tree routing achieves higher throughput than DHT routing.[7]
- DHT routing is more scalable in terms of routing burden and more resilient to node/link failures due to multiple paths than tree routing.

Thus, we introduce a hybrid approach whose routing structure consists of two tiers: a DHT is the high tier, and a tree is the low tier.

At the low tier, CON nodes form a tree structure. As the tree covers only a part of the whole network, the routing scalability issue is not significant. At the high tier, CON nodes form a DHT. That is, only the root node of each tree will participate in the DHT. Therefore, a query for a content file published in the same tree will be serviced within the tree. If a query is for a content file outside the tree structure, the DHT structure is exploited to forward the query to the corresponding tree where the requested content is published. Figure 2 illustrates how a query is forwarded across the two trees via the DHT in two-tier.

Comparison of Routing Structures

We compare tree, two-tier, and DHT in terms of a resolution delay, which refers to how long it takes for a content request to arrive at the content publisher. Also, we measure how the routing structure resiliently routes content requests in presence of node failures.

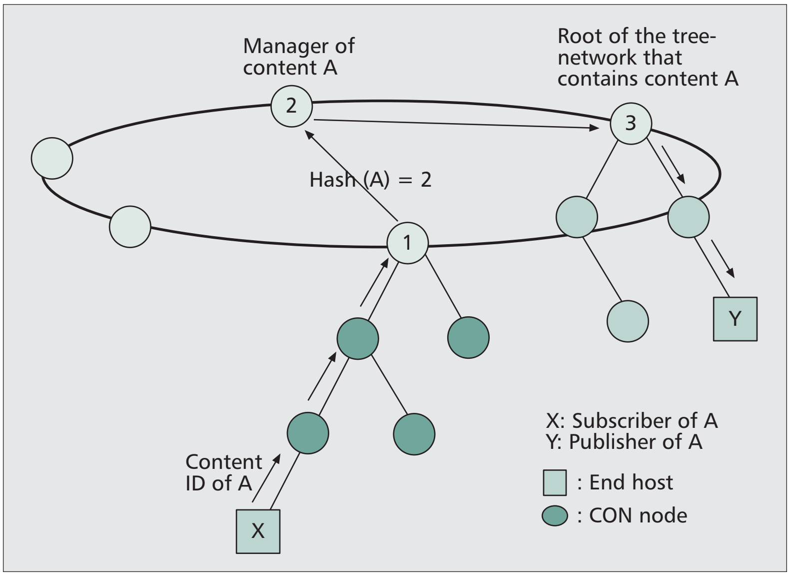

Figure 2. *Two-tier name-based routing architecture.*

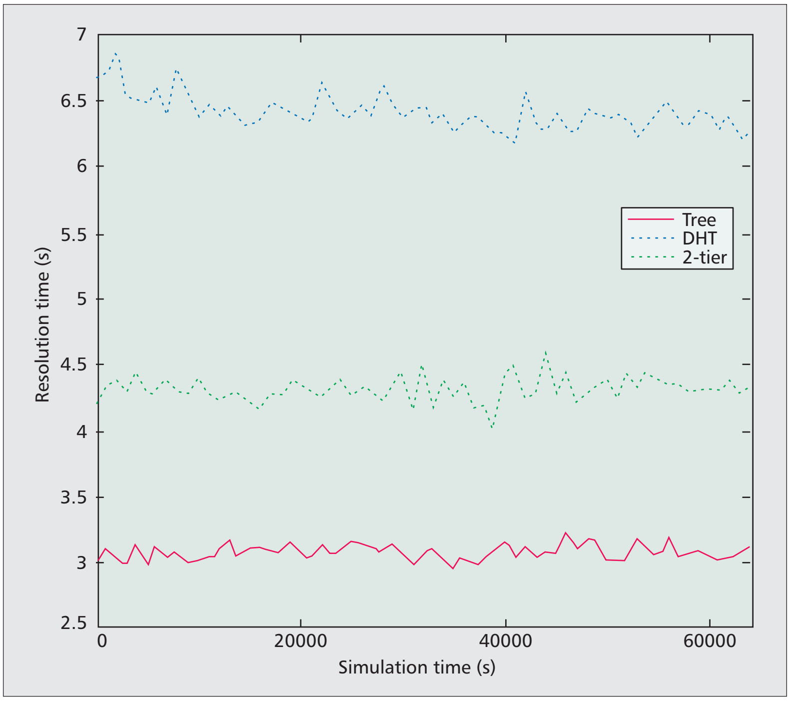

Figure 3. *Resolution delay comparison between three structured approaches.*

Figure 3 shows the resolution delays of the three routing structures. The tree structure outperforms the DHT since the DHT topology is constructed without any information on the physical topology. Thus, a content request goes back and forth among CON nodes of the DHT to reach its corresponding publisher node. The performance of two-tier falls between because it is a hybrid approach.

Figure 4 shows the successful resolution ratio as the node failure rate increases. Here R means that each of 100 CON nodes will fail for one hour once during a simulation run on average. The performance gain of the DHT over the tree is noticeable since there are multiple paths

[7] *Usually, a node's position in a DHT is determined randomly, e.g. by a hash function.*

among nodes in the DHT, and a failure of a higher-level node in the tree results in more routing failures. In the case of two-tier, there are 10 trees; each tree is formulated by the CON nodes in the same stub domain. The root nodes of the 10 trees participate in the DHT as well. From Figs. 3 and 4, the two-tier structure compromises on the trade-off between the tree (performance) and the DHT (resilience). Recall that the nodes of the two-tier will have less routing burdens than the tree.

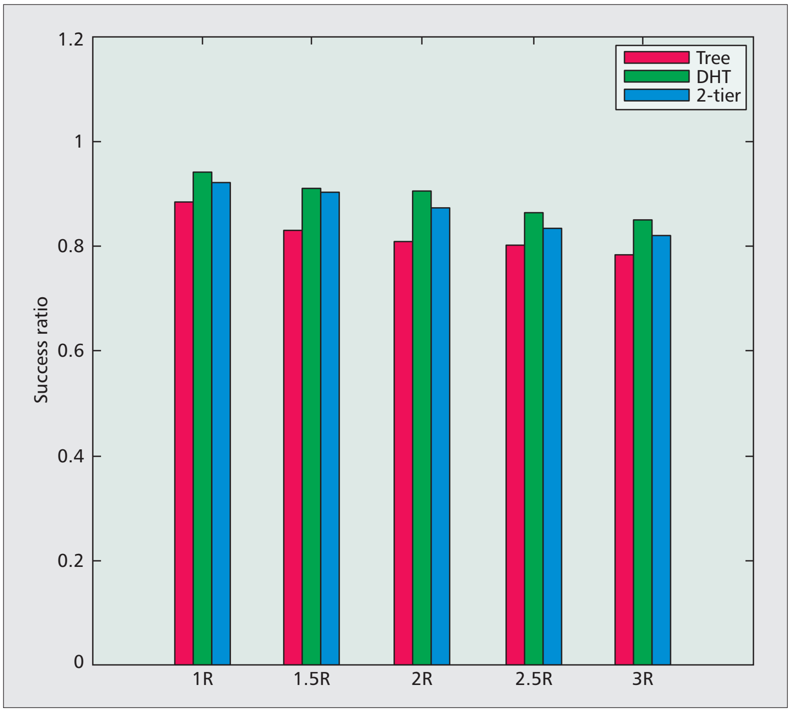

Figure 4. *Robustness comparison between three structured approaches.*

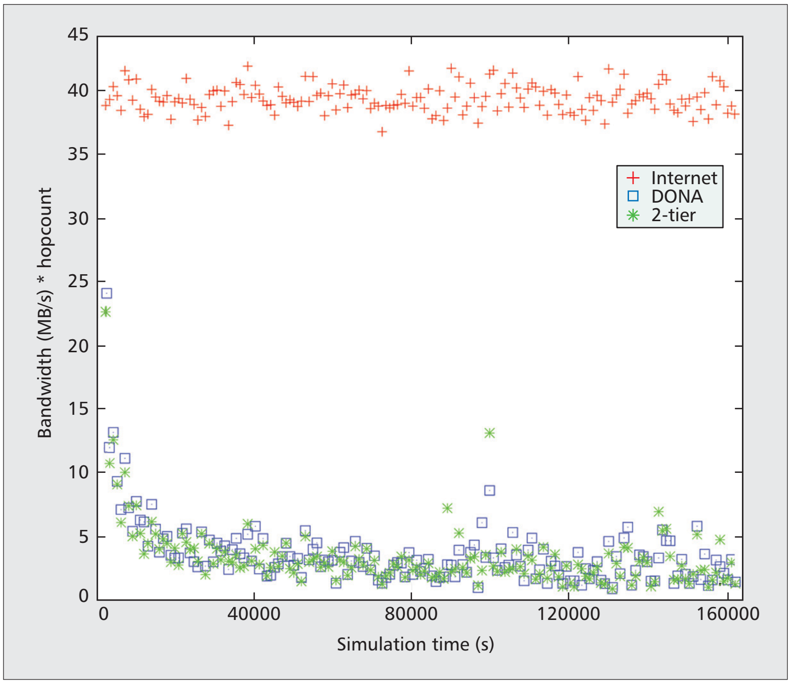

Figure 5. *The impact of publish/subscribe paradigm and in-network caching on the content delivery.*

NETWORK TRAFFIC LOAD

In the second experiment, we demonstrate how much traffic load is reduced by CON. We compare DONA (tree structure), two-tier, and the current Internet. The cache replacement policy for the CON nodes is Least

The cache size of each CON node is 5 Gbytes. Note that there is no publish/subscribe paradigm and in-network caching in the current Internet.

Figure 5 shows how much CON proposals (DONA and two-tier) can reduce the network-wide traffic load by the publish/subscribe paradigm and in-network caching. The performance metric is the product of hop count and link bandwidth (consumed to deliver contents). As the simulation time goes on, the product of hop count and bandwidth diminishes in CON proposals due to the cache effect. Each plot is the average of 1000 s. Thus, the cache effect appears almost from the beginning. Sometimes, two-tier exhibits slightly poorer performance than DONA due to DHT overlay inefficiency.

CONCLUSIONS

To fundamentally solve the mismatch between content-oriented Internet usage and host-based Internet architecture, content-oriented networking studies have been proliferated with a focus on naming and routing. In this article, we classify and compare the prior proposals in terms of naming and routing criteria. Also, we identify two important research topics in CON environments:

- How to disseminate contents from multiple sources
- How to decide which contents to cache in distributed environments

We then compare CON routing structures and demonstrate the performance gain of CON over IP networking. For future work, an interesting topic would be to compare the in-network caching part of CON proposals and CDN solutions in terms of network traffic load mitigation.

ACKNOWLEDGMENT

This publication is partially based on work performed in the framework of the Project COAST-ICT-248036, which is supported by the European Community. This work is also supported by NAP of Korea Research Council of Fundamental Science and Technology, and the IT R&D program of MKE/KEIT (10035245: Study on Architecture of Future Internet to Support Mobile Environments and Network Diversity). The ICT at Seoul National University provides research facilities. Professor Ted "Taekyoung" Kwon was on sabbatical leave at Rutgers University, where he had in-depth discussions with Professor Dipankar Raychaudhuri.

REFERENCES

[1] H. Schulze and K. Mochalski, ipoque -internet study 2008/2009, http://portal.ipoque.com/downloads/index/study.
[2] T. Koponen *et al.*, "A Data-Oriented (and Beyond) Network Architecture," *SIGCOMM '07*, 2007, pp. 181–92.
[3] V. Jacobson *et al.*, "Networking Named Content," *CoNEXT '09,* New York, NY, 2009, pp. 1–12.

[4] K. Visala *et al.*, "An Inter-Domain Data-Oriented Routing Architecture," *ReArch '09: Proc. 2009 Wksp. Re-architecting the internet*, New York, NY, 2009, pp. 55–60.
[5] P. Eugster *et al.*, "The Many Faces of Publish/Subscribe," *ACM Computing Surveys 35*, 2003, pp. 114–31.
[6] M. Gritter and D. R. Cheriton, "An Architecture for Content Routing Support in the Internet," *3rd Usenix Symp. Internet Technologies and Sys.*, 2001, pp. 37–48.
[7] A. Carzaniga *et al.*, "A Routing Scheme for Content-Based Networking," *IEEE INFOCOM '04*, Hong Kong, China, Mar. 2004.
[8] P. Ganesan, "Canon in g major: Designing Dhts with Hierarchical Structure," *ICDCS*, 2004, pp. 263–72.
[9] J. Erman *et al.*, "Network-Aware Forward Caching," *WWW 2009*, 2009.
[10] S. BORST *et al.*, "Distributed Caching Algorithms for Content Distribution Networks," INFOCOM 2010, 2010.

BIOGRAPHIES

JAEYOUNG CHOI (jychoi@mmlab.snu.ac.kr) received his B.S. degree in computer science and engineering from Seoul National University, Korea, in 2004. He is currently working toward his Ph.D. degree at the School of Computer Science and Engineering, Seoul National University. His research interests include content-oriented networking, peer-to-peer networking/application, and global Internet infrastructure including inter/intradomain routing protocols.

JINYOUNG HAN (jyhan@mmlab.snu.ac.kr) received his B.S. degree in computer science from Korea Advanced Institute of Science and Technology (KAIST), Daejeon, in 2007. He is currently working toward his Ph.D. degree at the School of Computer Science and Engineering, Seoul National University. His research interests include peer-to-peer networks, content-centric networks, and ubiquitous computing.

EUNSANG CHO (escho@mmlab.snu.ac.kr) received his B.S. degree in computer science and engineering from Seoul National University in 2008. He is currently working toward his Ph.D. degree at the School of Computer Science and Engineering, Seoul National University. His research interests include peer-to-peer, content-centric, and delay tolerant networking.

TED "TAEKYOUNG" KWON (tkkwon@snu.ac.kr) is with the School of Computer Science and Engineering, Seoul National University. He was a visiting professor at Rutgers university in 2010. Before joining Seoul National University, he was a postdoctoral research associate at the University of California Los Angeles and City University of New York. He obtained his B.S., M.S., and Ph.D. at Seoul National Uuniversity. He was a visiting student at IBM T. J. Watson Research Center and the University of North Texas. His research interest lies in future Internet, content-oriented networking, and wireless networks.

YANGHEE CHOI (yhchoi@snu.ac.kr) received his B.S. in electronics engineering at Seoul National University, M.S. in electrical engineering at KAIST, and D.Eng. in computer science at Ecole Nationale Superieure des Telecommunications Paris, France, in 1975, 1977, and 198,4 respectively. He worked at the Electronics and Telecommunications Research Institute (Korea), Centre National D'Etude des Telecommunications (France), and IBM Thomas J. Watson Research Center (United States) before joining Seoul National University in 1991. He was president of the Korea Institute of Information Scientists and Engineers. He is dean of the Graduate School of Convergence Science and Technology, president of the Advanced Institutes of Convergence Technologies, and chair of the Future Internet Forum of Korea. He has published over 600 papers in network protocols and architectures.