Performance Measurement of Content Distribution in Named Data Networking

Haowei Yuan, hyuan@wustl.edu (A paper written under the guidance of Prof. Raj Jain)

Download

Abstract:

Named Data Networking (NDN) [Zhang10] is a new network architecture that attempts to replace today's host-centric IP networks with a content-centric approach. In NDN, each packet has a unique name and routers make forwarding decisions by looking up packet names. Naming the data rather than the location enables NDN routers to cache packets and thus optimize network bandwidth. Content distribution applications consume a significant amount of bandwidth of today's Internet. NDN intrinsically supports content distribution with its caching functionality. In this paper, we measure the performance of NDN-based content distribution systems by running a prototype implementation of NDN, namely CCNx [CCNx], in Open Network Laboratory (ONL) [ONL]. Our preliminary performance analysis shows that content packet size has the most significant impact on the system throughput. In addition, we identified an inefficient implementation decision of CCNx. The system throughput was almost doubled in our modified CCNx program.

Keywords: Performance Measurement, Named Data Networking (NDN), Content Distribution, Content-Centric Network (CCNx)

Table of Contents

1. Introduction

The problem with today's Internet is that its usage doesn't match well with its communication model. The Internet was created about half a century ago, aimed at enabling communication between hosts. This communication focuses on the hosts rather than the transfered content. In IP networks, each host is assigned an IP address to identify its location. To retrieve some content from a server, the client needs to have the server's IP address and then visit that machine to get the content. To send content from the server to the client, the server uses the end-host's IP as the destination IP address of the outgoing packets. Essentially, the entire process is similar to a conversation, thus this communication model is sometimes referred as the conversational model. The conversational model and Internet work unexpectedly well and we have seen the fast growth of the Internet. A significant portion of today's Internet traffic fall into the category of content distribution[Saroiu02]. One important characteristic of content distribution is that multiple clients ask for the same content. If the traditional Internet conversational model is strictly followed, the same copy of content needs to be transferred multiple times across the network from the server, which is obviously not efficient. Several papers [Anand08][Anand09a] have discussed the redundancy they found in network traffic. There have been many new Internet architectures proposed to address this issue [Stoica04][Koponen07][Anand09b] [Jokela09]. NDN takes the content-centric [Jacobson09] approach, focusing on what the content is rather than where is the content is.

The two main characteristics of NDN are 1) packets have unique names, and they are routed based on packet name lookup results. There are two types of packets in NDN networks, interest packets and data packets. If a user want some content, the user will send out interest packets to express the requests. Data packets are used to reply interest packets with proper content. 2) The NDN routers are capable of caching some amount of forwarded data packets. This way, when there is an incoming interest packet to the router and the router has the requested content, it can send the content back to the clients immediately rather than forwarding the request all the way to the server. This feature can potentially save lots of bandwidth within the network.

NDN supports efficient content distribution intrinsically, but the performance of NDN-based content distribution system has not been studied. To the best of our knowledge, this is the first paper that conducts performance measurement on content distribution in NDN networks. As the first step of our performance study, we have measured the NDN forwarding engine throughput, which is a key metric in the NDN-based content distribution system. We found that content packet size is the factor that has the most significant impact on system throughput. In addition, we profiled the CCNx program and discovered that one inefficient implementation introduced a lot of unnecessary packet decoding. We modified the implementation and the system throughput of was almost doubled.

The reminder of the paper is organized as follows. Section 2 presents the background of content distribution in NDN. We design the experiments formally in Section 3. In section 4, the performance results are discussed. The performance bootstrapping methods and CCNx profiling results are presented in Section 5. Section 6 summaries the related work and Section 7 concludes the paper and discusses our future work.

2. Background

This section presents the background of content distribution systems, Named Data Networking and the Open Network Laboratory testbed.

2.1 Content Distribution

In general, content distribution is about delivering the same piece of content to multiple clients. There are many approaches for delivering content efficiently in the network.

The simplest approach is the client-server model. Where the server stores the content and the client can only get content from the server directly. To reduce the workload of the server, caching proxies [Spare01] are deployed, for instance, proxies are typically deployed on the edge of the network.

The second popular approach is Content Distribution Network[Freedman10]. CDN essentially forms an overlay network, where the same piece of content is replicated and stored at multiple locations. When users send content requests to the CDN, the CDN routes the requests to the closest content storing place. By taking advantage of locality, the content can be delivered to the user efficiently.

The third well-known technique is the Peer-to-Peer (P2P) method. In this approach, content is divided into chunks, and clients download content chunk by chunk. The global availability of the content pieces are maintained by the trackers[James10]. Once the clients finish downloading some content chunk, it serves as a content provider for that chunk. This way, the border between the clients and the servers blurred. P2P is still very popular in the network, but there are studies show that the percentage of the P2P traffic in the network is not as significant as before [Popa10].

In our NDN content distribution evaluation, we use the simple client-server approach, as the underlying NDN network architecture provides the caching functionality by design.

2.2 Named Data Networking

In this section, we present detailed NDN design and its prototype implementation. The system overview is presented first, followed by the prototype implementation details.

2.2.1 System Overview

Based on [Jacobson09], each node in the NDN network runs a daemon program. There is no distinction between routers, clients and servers. There are two types of packets in the network, interest packets and content packets. The packet names are hierarchically organized for scalability purposes.

The key system in the NDN network is the NDN forwarding node. Logically, the NDN forwarding node consists of three component, *Content Store (CS)*, *Forwarding Information Base (FIB)* and *Pending Interest Table (PIT)*. Each NDN node has a CS to store content packets it has forwarded. The FIB stores the forwarding information for each name prefix, and the PIT is used to store the unsatisfied interest packets, so that when the replied data packet arrived at this node, by performing a lookup in the PIT, we could get the list of nodes who asked for that content before.

2.2.2 Prototype Implementation

A prototype implementation of the NDN forwarding node has been developed in PARC, namely, *Content Centric Networking Project (CCNx)*. The key component of CCNx is the *ccnd* daemon, which is the program provides packet forwarding and caching. The current ccnd program runs as an overlay of IP networks, while keeping in mind that the NDN doesn't rely on IP. In fact, if there exist an physical NDN network, an IP network can also be implemented as an overlay in NDN networks. The CCNx project also includes several demo applications that run on the CCNx network, such *ccncatchunks2* that will be used in our evaluation. For the aforementioned programs, they are implemented in both C and Java. In addition, the Java version has a stand alone repository application that can serve as a large content warehouse. Note that this repository is different from the CS in the ccnd daemon. The performance of the repository is out the scope of this paper. Since the CCNx code is still under development, the software we evaluated is the one released on March 8, 2011.

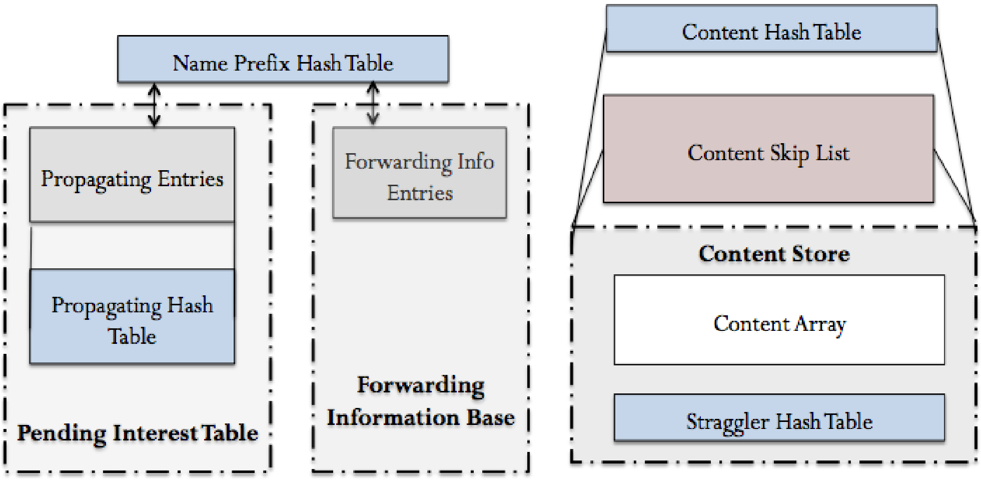

Figure 1. CCNx Forwarding Engine Data Structures

i) *Data Structures*. Figure 1 shows the data structures of the ccnd C implementation. This figure was created based on our study of CCNx source code. The logical FIB and PIT share a hash table named *Name Prefix Hash Table (NPHT)*, which indexes the *Propagating Entries(PEs)* and *Forwarding Info Entries (FIEs)*, where the detailed pending interest information and forwarding information are stored, respectively. The *Propagating Hash Table (PHT)* is keyed by the *nonce* field of interest packets and stores all the nonce fields of the interests packets presented in the PEs. PHT prevents loops in the network. For the CS, each content packet is assigned a unique *accession number* when they entering the daemon. The cached content packets are stored in an array called *Content Array (CA)* indexed by the accession number. Some of the content packets are stored in the *Straggler Hash Table (SHT)* for space efficiency. There are two data structures, *Content Hash Table (CHT)* and *Content Skip List (CSL)*, are used to index the Content Store. The CHT is a normal hash table keyed by the name of the content packet. The CSL provides efficient content-order lookup.

ii) *Processing Interest Packets*. When an interest packet arrives at the router, the PHT is queried first to ensure it is loop-free. After that, CSL is queried to find potential matching content. If there is a potential match answered by the CSL, the content will be pulled from the CS, and a comprehensive match check is performed. The reason for conducting this comprehensive match check is due to the fact that there might be restrictions embedded in the interest packet, for the detailed explanation, please refer to [CCNx]. If this is a satisfied match, the content packet will be delivered to the user who sent this interest packet. If this is not a satisfied match or there is no potential match at all, the daemon looks up the interest name on the NPHT. A longest prefix matching is performed to locate the appropriate FIE. Then this interest packet is forwarded to that host and it is also inserted into the PIT and PHT.

iii) *Processing Content Packets*. When a content packet arrives, the CHT is queried to make sure this packet is not stored in the CS. If the packet is not duplicated, it will be inserted to the CSL and CS after performing a lookup on the CSL. After that, the NPHT is queried in the longest prefix matching fashion to find all the interest packets that can be satisfied by this content packet. Once the interest packets are located, this content packet is delivered to those clients. Multi-casting is intrinsically supported in this design.

2.3 Open Network Laboratory

The *Open Network Laboratory (ONL)* is a publicly available network testbed. It consists of more than 100 hosts. In addition, there are several IXP Network Processors (NPs) that can be programmed by the ONL users. ONL provides a graphical user interface, namely *Remote Laboratory Interface (RLI)*, to the users for easy network topology constructions. For measurements, the NP-based ONL routers support traffic monitoring on each port of the router and the traffic throughput can be displayed in the RLI. We took advantage of this feature in our measurement study.

3. Evaluation Design

To evaluate the performance of distributing content in NDN, we designed our experiment formally. The definition of the system and the goals of this evaluation work are introduced first. After that, the metrics that will be used to evaluate the performance are selected. The parameters that can potentially affect the system performance are discussed, and then we select the factors that going to be studied in this paper. In the end, we describe the evaluation technique and 2^k experimental design.

3.1 System Definition

The content distribution system in NDN consists of three components, NDN Clients, NDN Servers and NDN routers. All three of these roles run the same ccnd daemon program, so that all of the hosts involved in the system have the capability of forwarding and caching packets. The clients run client-end software while servers run server-end application on top of ccnd. The router doesn't run any other application, but is dedicated to the forwarding functionality. To distribute a content, the content is initially stored in the server, the clients download the content by sending out interest packets. The server application replies to the interest packet by sending the content packet. When the content packet is forwarded along the path, it is cached in the NDN router's content store.

3.2 Goals and Metrics

The goal of this project is to evaluate the performance of distributing content in the NDN context. The metrics should be selected to reflect the quality of services provided by the studied system. The essential functionality provided by NDN-based content distribution system is distributing content to the end users. Therefore, ideally, the appropriate metrics are Content Download Time (CDoT) at each end host and Content Distribution Time (CDiT) of the entire system. CDoT is the time between when the client sends out the first interest packet and all the content packets are received. CDiT is the time between when the first interest packet is sent among the clients and all the clients receive the content. Measuring these metrics requires developing our own client and server programs, and it's currently work in progress. To simply the measurement for this course project, we use the NDN router forwarding rate as the metric to be studied. NDN forwarding rate can be measured using the NP-based router in ONL. We believe this is a good starting point for evaluating NDN-based content distribution system. The NDN router forwarding rate is defined as the maximum throughput it can support when its CPU is saturated.

3.3 Parameters and Factors

In computer network systems, there are many parameters that can affect the system performance. We present all the parameters for the NDN-based content distribution system in this section. The system parameters are introduced first, followed by the discussion of workload parameters. The selected factors are listed in the end.

Table 1. System Parameters

Parameters	Description
Operating System	Linux 2.6.24.7
Host Memory	1GB
Host CPU	2.0 GHz AMD Opteron
Ethernet Interface	1 Gbps
Transmission Protocol	TCP or UDP
Content Store Capacity	Configurable

Number of Servers	Configurable
Number of Clients	Configurable
Number of Routers	Configurable
Server Application	Configurable
Client Application	Configurable
Network Topology	Configurable
Network Reliability	Configurable

i) System Parameters: Our performance study is conducted in the ONL. The hardware parameters cannot be changed easily, while some of the software parameters, for instance, the configuration of the CCNx on each node could be modified. The system parameters are listed in Table 1.

Table 2. Workload Parameters

Parameters	Description
Content Name Length	Configurable
Content Packet Size	Configurable
Content Diversity	Configurable

ii) Workload Parameters: Since there is no real-world NDN packets trace available, we have designed and used synthetic workloads for this measurement study. Table 2 lists the workload parameters for this performance study.

Table 3. Factors to Study

Factors	Description
Content Store Capacity	10K, 50K, 100K
Content Name Length	2, 4, 8, 16, 32, 64
Content Packet Size	64B, 128B, ..., 1KB
Content Diversity	1, 2, 4, 8, 16

iii) Factors: The factors are selected from the parameters listed in Table 1 and 2. The key factors to study are listed in the Table 3. By changing the Content Store capacity, the number of pieces of content that can be stored in the NDN router is configured. Generally speaking, a larger CS can hold more content and thus provide a higher probability of satisfying the incoming requests. In the mean time, larger CS does increases the lookup time. Recall each NDN packet is assigned a unique name and packets are routed based on their names. Each NDN packet name consists of several component divided by the slash symbol, for instance, the name */wustl.edu/server/test* consists of three components, *wustl.edu*, *server* and *test*. The length of the NDN packet name in this paper is defined as the number of components in the name. Longer names increases the name lookup time and thus affects the system throughput. The content diversity is defined under the assumption that every client only asks for one content, the value of content diversity is the number of different contents being requested from the clients. For instance, if all of the clients ask for the same content, the content diversity is 1. Similarly, if all of them ask for different content, the content diversity is the same as the number of clients in the system.

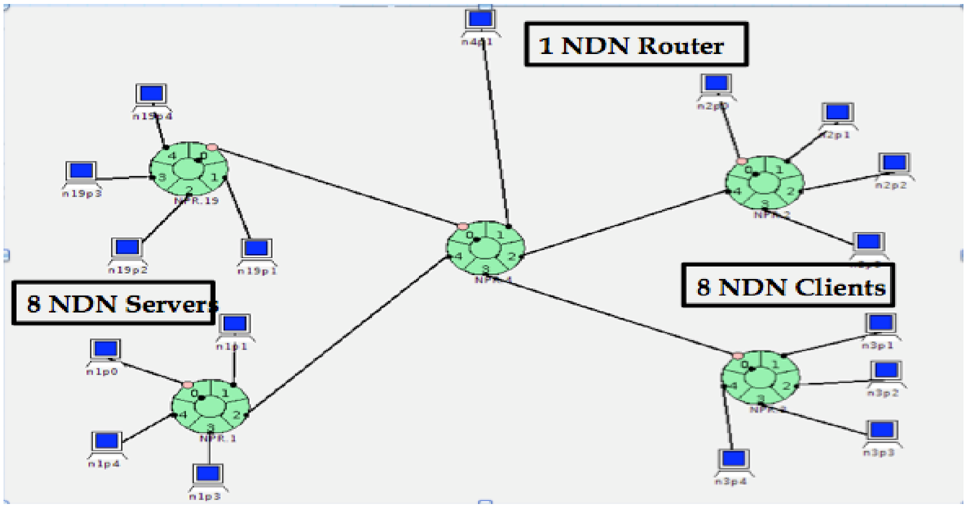

Figure 2. Network Topology

The rest of the parameters are all fixed. We use the *ccncatchunks2* program as the client download application. On the server side, our research team developed a simple server programs named *ccndelphi* that does nothing but replies any interest request with random data. The number of servers and clients are both set to be 8. This number is determined to make sure the CPU of the NDN router can be saturated such that the maximum system throughput can be measured. TCP is selected as the transmission protocol for this evaluation. Since the experiments were conducted in ONL, we assume the network is reliable and there is no interruptions. The topology of the network is shown in Figure 2, which is a screen capture of the ONL RLI interface. We can see that 8 NDN servers are connected to 8 NDN clients via an NDN router. The NP routers are used to connect the network and measure the traffic throughput.

3.4 Evaluation Technique

Since all the prototypes of the programs used in the experiment have already been developed, we can perform measurement study of the system using ONL. The network topologies is constructed using RLI, as we have seen in Figure 2. A few scripts have been developed to control the hosts that participated in the experiments. The throughput of the NDN router is measured by the NP-based routers available in ONL, the measured data will be displayed in the RLI directly.

3.5 Experimental Design

As there are 4 factors to be studied, we apply 2^k factorial design method for our experimental design, where k=4. This yields 2^4 = 16 experiments. We list the factors and each factor's level in Table 4.

Table 4. Factor Symbol and Levels

Symbol	Factor	Level -1	Level +1
A	Content Store Capacity	10K	50K
B	Content Name Length	4	16
C	Content Packet Size	128B	1KB
D	Content Diversity	1	8

4. Performance Study

In this section, we evaluate the performance of the NDN-based content distribution system. The performances results are presented and discussed briefly first, then we apply data analysis technique to identify the top factors that affect the performance.

Table 5. Measured System Throughput

#	A	B	C	D	Mbps
1	-1	-1	-1	-1	25
2	1	-1	-1	-1	27
3	-1	1	-1	-1	21
4	1	1	-1	-1	18
5	-1	-1	1	-1	97
6	1	-1	1	-1	103
7	-1	1	1	-1	46
8	1	1	1	-1	48
9	-1	-1	-1	1	26
10	1	-1	-1	1	24
11	-1	1	-1	1	15
12	1	1	-1	1	13
13	-1	-1	1	1	61
14	1	-1	1	1	59
15	-1	1	1	1	31
16	1	1	1	1	28

4.1 Performance Results

Following the experimental design, we did 16 experiments on ONL. The throughput of the incoming traffic to the NDN router and the outgoing traffic from the router are measured using the traffic monitoring tool available on the NP-based routers. The bandwidth utilization is displayed in the RLI, and the value of the throughput is directly read from the RLI. All the 16 measured system throughput is listed in Table 5. The highest throughput is 103 Mbps, which is much less than the today's line rate requirement.

Table 6. Top 5 Contributors to the Variation

Symbol	Var(%)
C	51
B	22
BC	9.5
D	9
CD	5.6

4.2 Data Analysis

To decide what factors affect the system performance most, we analyzed the data listed in Table 5. The top 5 factors in term of contribution to the variation are listed in Table 6. According to Table 6, Factor C, the content packet size, is the most important factor that affect system throughput. Factor B, the content name length is the second most important one. Factor BC, the intersection of the aforementioned two factors is the third, followed by Factor D, the content diversity. In addition, Factor A, the content store size, is not significant in term of affecting the system throughput.

5. Improving System Performance

In the previous section, the highest throughput we can get is 103 Mbps, which is still far from the current line speed, 1Gbps. In order to demonstrate NDN is capable of providing at least comparable content distribution functionality as other content distribution method, it is important to improve the system performance. In this section, methods of accelerating forwarding engine is discussed. In particular, we show that one of our simple change in the source code could almost double the forwarding node throughput.

To improve the forwarding rates, there are two basic approaches. 1) Design or use more efficient data structures to support packet name lookup. For example, Bloom Filter is a compact data structure for membership query. Applying bloom filter to the CCNx design might improve the performance. 2) Employ advanced hardware devices. For instance, TCAM can be used to assist longest prefix matching. In addition, implement the NDN forwarding node on network processors can potentially improve the performance.

Table 7. Top 10 Time-Consuming Functions

Function Name	**Percentage(%)**
ccn_skeleton_decode	47.34
ccn_buf_advance	7.58
ccn_compare_names	6.1
hashtb_hash	5.87
ccn_buf_match_dtag	3.55
ccn_parse_Signature	3.11
ccn_buf_match_blob	2.01
content_skiplist_findbefore	1.48
hashtb_seek	1.36
content_from_accession	1.21
The Rest	20.39
Sum	100

Since CCNx is just a proof of concept prototype, when it was developed, efficiency was unlikely to have the highest priority. Practically, it's possible to improve the performance by optimizing the current implementation. We profiled the ccnd daemon using the profiling tool gprof [Graham04]. The top 10 time-consuming functions are listed in Table 7. The functions in italic are the ones related to packet decoding. According to Table 7, more than 50% of the time were spent on packet name decoding, which is unusual in general network programs. We found what causes this problem is the content packet stored in the CS are in encoded format. When the content skip list is queried, on average log(n) packets need to be decoded on the fly where n is the number of items stored in the CSL, thus consuming a lot of time. We made a simple change to the program, such that the decoded content packet name is stored in the daemon, this way, there is no extra packet decoding when a CSL is queried. We measured the NDN router forwarding rates and found that the throughput was almost doubled.

6. Related Work

There have been performance studies on the efficiency of performing content distribution on various network architectures and applications, such as proxy caching [Wolman99], CDN [Freedman10] and P2P [James10]. Recently, there is a paper that discusses the CCNx content store hit rates [Giovanna Carofiglio11], but their study was not related to system forwarding rate. To the best of our knowledge, our work is the first performance study of distribution content in the context of NDN.

7. Conclusion and Future Work

In this project, we measured the performance of NDN-based content distribution system. Particularly in this paper, we focused on the throughput of the NDN router. Our experimental study shows that the system throughput of current CCNx implementation is far from the line rate requirement. Our data analysis shows that content packet size is the most important factor affects the performance. In addition, Several directions of improving system performance are also discussed in our paper. There is no doubt there are a lot work need to be done in order to provide a more comprehensive NDN-based content distribution system performance analysis. 1) The CDoT and CDiT metrics need to be studied once the modified client and server programs are developed. 2) The current number of name prefixes stored in the forwarding is too small comparing with any realistic applications, thus we should introduce more name prefixes in the evaluation. 3) Our current network topology is very simple, more complicated and practical topology needs to be explored.

References

The references are listed based on their importance to this report.

[Zhang10] Zhang, L. et al., "Named Data Networking (NDN) Project." Technical Report NDN-0001, NDN, 2010. http://www.named-data.net/ndn-proj.pdf

[Jacobson09] Jacobson, V., Smetters, D. K., Thornton, J. D., Plass, M. F., Briggs, N. H., and Braynard, R. L., "Networking Named Content." In Proceedings of the 5th international conference on Emerging networking experiments and technologies, CoNEXT 2009. http://conferences.sigcomm.org/co-next/2009/papers/Jacobson.pdf

[CCNx] Project CCNx. http://www.ccnx.org, 2011. http://www.ccnx.org

[Giovanna_Carofiglio11] Giovanna Carofiglio, V. G. and Perino, D., "Experimental Evaluation of Storage Management in Content-Centric Networking" In Proceedings of IEEE ICC, 2011 http://www.diegoperino.com/publications/CCN_ICC11.pdf

[Saroiu02] Saroiu, S., Gummadi, K. P., Dunn, R. J., Gribble, S. D., and Levy, H. M., "An Analysis of Internet Content Delivery Systems." In Proceedings of the 5th symposium on Operating systems design and implementation, OSDI 2002. http://www.cs.washington.edu/research/networking/websys/pubs/osdi_2002/osdi.pdf

[Anand08] Anand, A., Gupta, A., Akella, A., Seshan, S., and Shenker, S., "Packet Caches on Routers: the Implications of Universal Redundant Traffic Elimination" In Proceedings of the ACM SIGCOMM, 2008. http://portal.acm.org/citation.cfm?id=1402984

[Anand09a] Anand, A., Muthukrishnan, C., Akella, A., and Ramjee, R., "Redundancy in Network Traffic: Findings and Implications." In Proceedings of the eleventh international joint conference on Measurement and modeling of computer systems, SIGMETRICS, 2009. http://http://portal.acm.org/citation.cfm?id=1555355

[Anand09b] Anand, A., Sekar, V. and Akella A., "SmartRE: An Architecture for Coordinated Network-wide Redundancy Elimination." In Proceedings of ACM SIGCOMM, 2009http://ccr.sigcomm.org/drupal/files/p87.pdf

[James10] James, S. and Crowley, P., "ISP Managed Peer-to-Peer." In IEEE P2P, 2010. http://arl.wustl.edu/~scj1/pub/james-p2p10.pdf

[Popa10] Popa, L., Ghodsi, A., and Stoica, I., "Http as the Narrow Waist of the Future Internet." In Proceedings of ACM SIGCOMM Workshop on Hotnets, 2010 http://portal.acm.org/citation.cfm?id=1868447.1868453

[Jokela09] Jokela, P., Zahemszky, A., Esteve Rothenberg, C., Arianfar, S., and Nikander, P., "LIPSIN: Line Speed Publish/Subscribe Inter-Networking." In Proceedings of ACM SIGCOMM, 2009. http://ccr.sigcomm.org/drupal/files/p195.pdf

[Koponen07] Koponen, T., Chawla, M., Chun, B.-G., Ermolinskiy, A., Kim, K. H., Shenker, S., and Stoica, I., "A Data-Oriented (and Beyond) Network Architecture." In Proc. of ACM SIGCOMM, 2007. http://ccr.sigcomm.org/online/files/fp177-koponen1.pdf

[Stoica04] Stoica, I., Adkins, D., Zhuang, S., Shenker, S., and Surana, S., "Internet Indirection Infrastructure" In IEEE/ACM Transactions on Networking, Vol. 12, Issue 2, April 2004. http://portal.acm.org/citation.cfm?id=987233

[Spare01] Spare, I., "Deploying the Squid Proxy Server on Linux" In Linux Journal, 2001. http://portal.acm.org/citation.cfm?id=364769

[Freedman10] Freedman, M. J., "Experiences with CoralCDN: A Five-Year Operational View" In Proceedings of NSDI, 2010. http://portal.acm.org/citation.cfm?id=1855718

[Wolman99] Wolman, A., Voelker, G. M., Sharma, N., Cardwell, N., Karlin, A., and Levy, H. M., "On the Scale and Performance of Cooperative Web Proxy Caching" In Proceedings of SOSP, 1999. http://portal.acm.org/citation.cfm?id=319153

[Graham04] Graham, S. L., Kessler, P. B., and McKusick, M. K., "gprof: a Call Graph Execution Profiler". ACM SIGPLAN Notices, Vol. 39, Issue 4, April 2004. http://portal.acm.org/citation.cfm?id=989401

[ONL] Open Network Lab. http://www.onl.wust.edu, 2011. http://www.onl.wust.edu

List of Acronyms

CCNx	Content-Centric Networks
CDN	Content Distribution Network
CDoT	Content Download Time
CDiT	Content Distribution Time
CHT	Content Hash Table
CA	Content Array
CS	Content Store
CSL	Content Skip List
FIE	Forwarding Information Entry
FIB	Forwarding Information Base
NDN	Named Data Networking
NP	Network Processor
NPHT	Name Prefix Hash Table
ONL	Open Network Laboratory
PE	Propagating Entry
PHT	Propagating Hash Table
PIT	Pending Interest Table
RLI	Remote Laboratory Interface
SHT	Straggler Hash Table

Last modified on April 24, 2011

This and other papers on latest advances in performance analysis are available on line at
http://www1.cse.wustl.edu/~jain/cse567-11/index.html
Back to Raj Jain's Home Page