

# The Resource Efficient Forwarding in the Content Centric Network

Yifan Yu, Daqing Gu

► **To cite this version:**

Yifan Yu, Daqing Gu. The Resource Efficient Forwarding in the Content Centric Network. Jordi Domingo-Pascual; Pietro Manzoni; Sergio Palazzo; Ana Pont; Caterina Scoglio. 10th IFIP Networking Conference (NETWORKING), May 2011, Valencia, Spain. Springer, Lecture Notes in Computer Science, LNCS-6640 (Part I), pp.66-77, 2011, NETWORKING 2011. <10.1007/978-3-642-20757-0\_6>. <hal-01583396>

**HAL Id: hal-01583396**

**<https://hal.inria.fr/hal-01583396>**

Submitted on 7 Sep 2017

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



# The Resource Efficient Forwarding in the Content Centric Network

Yifan Yu, Daqing Gu

Orange Labs, Beijing. 10/f, South Tower Raycom Info Park C, 2 Science Institute  
South Rd, Beijing, P.R.C.

{yifan.yu, daqing.gu}@orange-ftgroup.com

**Abstract.** Today's networking technologies in Internet that can only speak of connections between hosts no longer adapt to the network increasingly dominated by content distribution and retrieval. Some efforts therefore have been made to generalize the Internet architecture by taking content as a primitive- decoupling location in the network instead of focusing on endpoint addresses of hosts. However, one of the representative architectures known as *Content-Centric Networking* (CCN) consumes excessive system resource in terms of router's processing capability, the buffer space in the router and the network bandwidth when deployed as the "universal overlay" in the existing network. We present the solution composed of two methods to improve the resource efficiency based on the CCN approach. The results from the ns2-based simulation disclose that our solution can reduce the resource consumption in routers and network pipeline significantly in the overlay architecture.

**Keywords:** Content-Centric Networking, Internet architecture, Forwarding, Resource efficient, Data

## 1 Introduction

Today's Internet was already designed to be an end-to-end connectivity infrastructure based on packet switching at the beginning of 1960s. The communication solution offered by it targets at solving the telephony's problem: a point-to-point conversation between two hosts using the IP protocol [1]. Thus IP packets exchanged between the two hosts in communication contain two identifiers (addresses), one for the source and one for the destination host. By allowing lower and upper layer technologies to innovate without unnecessary constraints, the Internet achieves the explosive growth to show its power and elegance.

However, there have been dramatic changes in Internet since its creation [2].

- With the emergency of World Wide Web (WWW) [3] application, the increasing number of Information-intensive business such as travel, banks and financial services is done in the Internet. Since almost anything is available online today, the Internet is becoming the mirror of the real world where more and more people's activities are digitalized.

- An ever increasing range of content such as voice, images and video etc. can be distributed digitally due to the digital coding advances.
- The cheaper and more ubiquitous digital equipments driven by the Moore's-Law facilitate connecting everything to the Internet: not just computers but also factories, municipal infrastructure, phones, cars, appliances, even light switches. There is considerable consensus that the amount of these equipments could reach the number of billions or even trillions [4] [5].

The above changes make people tend to value the Internet for what content it contains rather than where they can do communication. Then IP protocol, which was designed to act as a key enabler supporting the conversations between communications endpoints, is overwhelmingly used for content distribution. However, the 'conversational' nature of IP makes the existing Internet architecture a poor match to its primary use today.

In order to overcome the poor match between the IP's nature and the practical requirements of people using the Internet, several efforts [6–16] have been made to pursue the direct and unified way to solve the problems. From the view point of these works, the named data is a better abstraction for today's communication problems than named hosts. Then, the principle of the current innovation aim to interconnect the hosts by using the address that names content instead of the IP addresses traditionally identifying the physical network nodes [18–20]. In other words, the currently proposed solutions attempt to solve the problem of poor match in the way of replacing *where* with *what* in the core layer that the Internet centers on, which is typically termed as content-centric networking or named data networking.

The architectural principles that guide the design of the new architecture are stemmed from the existing Internet architecture. Substantially speaking, it departs from IP in creating the new universal component, known as content-based "thin waist", in the protocol stack [2] [6].

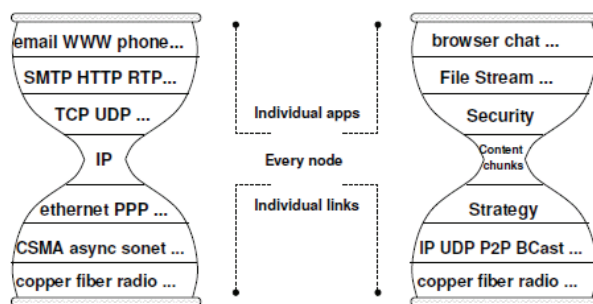


Fig. 1. Today's Internet Architecture Vs. Content Centric Architecture

As illustrated in Fig. 1, the current Internet architecture is built as the hourglass architecture where network layer (IP) acts as a universal component, namely "thin waist", to implement the minimal functionality necessary for global interconnectivity. The content centric architecture keeps the same hourglass-shaped architecture except that the "thin waist" is oriented to the content chunks instead of the physical hosts with the identifier of IP addresses.

Unlike today's Internet architecture that regards the security as an afterthought, the content centric one provides a basic security embedded in the thin waist by signing all named data. In such architecture, security is provided to content itself, rather than the connections over which it travels, thereby avoiding many of the host-based vulnerabilities that plague IP networking. The end-to-end principle that guarantees the robust applications in the face of network failures is retained and expanded by the content centric architecture. Particularly, the multiple simultaneous connections (e.g., Ethernet, 3G, LTE, Zigbee and 802.11) can be utilized to achieve the fine-grained, dynamic optimization for the data under changing conditions with the consideration of traffic self-regulation.

Routing and forwarding plane separation is still stuck to in the content centric architecture. But the new routing system should be carried out to allow the addressing oriented to the content name.

The rest of the paper is organized as following. The motivation of our work in the paper is presented in section 2. In Section 3, our solution is described in detail. We evaluate the performance of our proposal with the extensive ns2-based simulation in section 4. Finally, in Section 5, we give the conclusion of the paper.

## 2 Related Work and Motivation

The works concerning the content centric architecture can be classified as two categories: hierarchical name-based and flat name-based. The former identifies the content with the hierarchical names much like Uniform/Universal Resource Locator (URL) [21] used in the current web application [6, 7], while the latter would like to define the name of the content using the flat label [8–16] that is similar to the identifiers used in the existing distributed hash table (DHT)-based P2P system [22] [23]. Since our work in the paper focuses on the hierarchical name-based solution, we just describe the existing approaches in this category.

### 2.1 Content-Centric Networking

The classical approach realizing the content centric networking with the hierarchical name is presented in [6]. It is called as Content-Centric Networking (CCN). The communication in CCN is driven by the data consumer. As shown in Fig. 2, two CCN packet types, Interest and Data, are defined to enable the communication oriented to the specific content chunk.

To receive data, a receiving end, namely the consumer, sends out an Interest packet carrying a name that identifies the desired data. For example, a consumer

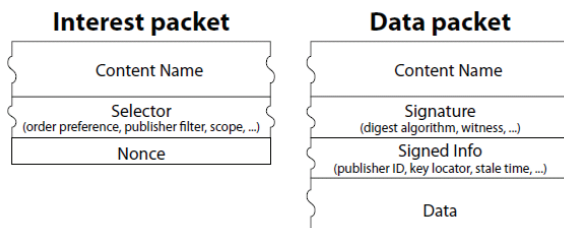


Fig. 2. CCN packet types

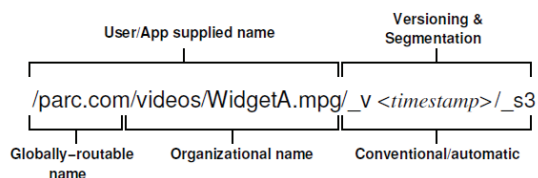


Fig. 3. Example of Data Name

may send an Interest packet requesting the data named after `/parc/videos/WidgetA.mpg`. Fig. 3 gives the example that how the name is defined.

A router maintains a data structure termed as Pending Interest Table (PIT) to remember the face where the request arrives, and then forwards the Interest packet by looking up the name in its Forwarding Information Base (FIB) where the list of faces that can serve the request is recorded. FIB is almost identical to an IP FIB except it allows for a list of outgoing interfaces rather than a single one. The name-based routing policy is used for the population of this table. Since the packets introduced in [6] are not only forwarded over hardware network interfaces but also exchanged directly with application processes within a machine, the term face rather than interface is used to denote the place where the packets come from and are forwarded.

Once the Interest reaches a node that has the requested data in its local storage termed as Content Store, a Data containing the requested name and the content of the data packet, together with a signature by the producer's key, is sent back. This Data packet traces in reverse the path created by the Interest packet back to the consumer.

When multiple Interests for the same data are received, only the first Interest is sent upstream towards the data source. When the Data packet arrives, the router finds the matching PIT entry by using longest-match look-up based on the content name and forwards the data to all the interfaces listed in the PIT entry. The router then removes the corresponding PIT entry, and optionally caches the Data in its Content Store to satisfy potential future requests.

## 2.2 Motivation

In CCN approach, the arrival of each data packet triggers the routers to look up and update its PIT and FIB with the name of received data that is expressed as a structured string similar to URL. However, such method may result in the excessive resource consumption in terms of router processing capability, router's buffer capacity and network bandwidth.

Specifically, CCN's compatibility with today's Internet centering on IP protocol makes it to readily use existing IP infrastructure services. Thus, CCN can run as a "universal overlay" over the current Internet. The routing policy of CCN router then may be regarded as IP+CCN, which means that the CCN router should execute the IP routing prior to the CCN processing of received packet. In other words, there is double load in the processor of router because the arrival of any CCN packet requires the handling both in the IP layer and CCN layer. Especially, if the router is not to cache the received CCN data, which is common because the router is unable to cache each received data due to the limited buffer space, the CCN router will be confronted with extra processing load because the data can be simply forwarded as the IP packet without any CCN operations.

On the other hand, the URL-like name of CCN data implies that the string identifying the contained content is much longer than the physical identifier such as IP address. Thus, it is expected that lots of router buffer space and network bandwidth will be spent in accommodating the name of CCN data. Suppose the CCN name is 200 bytes in length and the router is provisioned with packet buffers 2KB in size with 2 GB of total storage capacity; this means Content Store should allocate the buffer space of 200 MB to store 1 million distinct packet names. In further consideration of the maximum transmission unit (MTU) in the underlying pipeline (i.e. Ethernet), only 85% of the bandwidth is used to transport the content payload given that MTU is 1.5KB.

Our work in this paper is motivated by the above two challenges for the current CCN approach. Briefly, our contributions are summarized as following:

- We propose a method allowing the intermediate router to avoid the unnecessary processing in the CCN layer. Specifically, we design the way of configuring source network identifier (i.e. IP address) in CCN Interest to allow for the direct forwarding of the data not to be cached in the intermediate CCN router without any CCN operations.
- We design the method to reduce the size of content name in the CCN data packet so that more network bandwidth and buffer space in the router can be used by the content payload. The principle of our method is to replace the URL-like content name in CCN data with an integer that can be recognized between the upstream node and downstream node. The control messages enabling such process are presented as well.

### 3 Resource Efficient Packet Forwarding

The solution proposed in the paper focuses on reducing the system resource consumption in terms of router's processing capability, buffer space in router and the network bandwidth. In details, our proposal includes two innovative methods that involve the enhanced processing of CCN interest data and the header reduction in CCN data, respectively. The former alleviates the processing load in the router, while the latter reduces the buffer space and network bandwidth consumed by the CCN data.

Note that the CCN architecture is assumed to be built as the overlay of the IP-based network in the following sections of our paper. But it does not mean that our solution can merely run in the IP network. In essence, our proposed solution can work in other underlying networks (such as LTE or WiFi etc.).

#### 3.1 Enhanced Processing of CCN Interest and CCN Data

Unlike the CCN approach that creates the PIT entry upon receiving each new CCN Interest, our solution has the router deal with the new CCN Interest based on its decision whether to cache the data requested in the Interest. The router then further determines how to modify the destination IP address and source IP address of the CCN Interest that needs to be forwarded.

Upon the reception of Interest, the router should firstly check whether it can serve the Interest by looking up the Content Store. If the router caches the content requested by the Interest, it will send out the matched data piece whose destination address is set as the source IP address of Interest.

If there is no matched entry in Content Store, the router checks whether the Interest has been pending in PIT. If so, the Interest is discarded and the router adds the source IP address of the Interest to the corresponding PIT entry.

Otherwise, the router is to check whether there is a matched entry in FIB. If not so, the Interest has to be discarded. Otherwise, the router should firstly remove the face where the Interest arrives from the face list of the matched entry. After that, the following steps should be taken:

- a) If the router wants to cache the content required by Interest, a new entry for pending the received Interest is firstly created in PIT. Then, the router has to replace the source IP address of the received Interest with its own IP. The destination IP address of the received Interest is then set as the IP address of the node where the Interest is to be forwarded.
- b) If the router has no intension to cache the content, it keeps the source IP address of the received Interest unchanged but the destination IP address is set as the IP address of the node where the Interest is to be forwarded.

Finally, the Interest is forwarded over all the faces of the matched FIB entry.

The processing of CCN Interest mentioned above allows the router to only take the CCN-defined reaction to the arriving data that it is ready to cache. It is because that the router is transparent to both its upstream node and downstream

node in the sense of CCN routing supposing that the source address of the forwarded Interest is not set as its own IP address.

Then, as the requested CCN data arrives, there will be two consequences: the router deals with the received data in the CCN-defined way and the source IP address of data is set as the router's IP address given that the data was cached. Otherwise, the source IP address of the data remains unchanged and it is treated as the normal IP packets without any CCN operations in the router.

### 3.2 Header Reduction in CCN Data

The header reduction method in our solution compresses the name of the CCN data by replacing the URL-like identifier with an integer understandable to upstream node and downstream node. It is used in the CCN data delivery between the CCN routers.

In our method, as the downstream CCN router receives a CCN data with the URL-like name, it may generate an integer ranged from 0 to  $2^{16} - 1$  to represent the User/App supplied name in the Content Name (in Fig. 3). The generated integer is locally unique in the downstream node. A table, termed as **Local Code Table (LCT)**, as in Fig. 4(a), that records the mapping between the integer and User/App supplied name is maintained in the downstream router.

User/App supplied name	Code	User/App supplied name	Code-Face Pair
/Orange.com/videos/01.mp4/	33	/Orange.com/videos/01.mp4	<A, 33>, <B, 91>
/Orange.com/news/01.mp3/	75	/Orange.com/news/01.mp3	<C, 75>
.....	.....	.....	.....

(a) Local Code Table

User/App supplied name	Code-Face Pair
/Orange.com/videos/01.mp4	<A, 33>, <B, 91>
/Orange.com/news/01.mp3	<C, 75>
.....	.....

(b) Remote Code Table

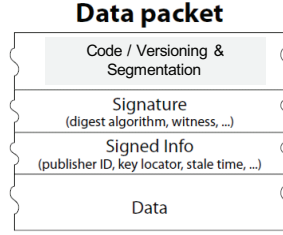
Fig. 4. Table for Header Reduction

Since the mapping entries in LCT needs to be known to the upstream router as well, the dedicated message called as **Code Notification Message (CNM)** is sent from the downstream node to the upstream node for LCT duplication that is known as **Remote Code Table (RCT)**. As shown in Fig. 4(b), considering the upstream router may have multiple downstream routers for the dissemination of same data piece, the codes oriented to various faces, namely downstream nodes, should be differentiated in RCT.

The arrival of CNM triggers the upstream router to update its RCT and replace the User/App supplied name in the follow-up data packets composing the same content with the integer announced in the message. Thus, the CCN data with the reduced header appears as Fig. 5 where the component of Versioning & Segmentation is identical to that defined in CCN.

After receiving the CCN data with the reduced header, the downstream router looks up its LCT with the code in the data header. If there is matched





**Fig. 5.** CCN Data with Reduced Header

entry, the data can be processed in the CCN way. Otherwise, the downstream router has to abandon the data and send a message, called as **Code Error Message (CEM)** where the invalid code is notified, to the upstream router, which then guides the upstream node to remove the invalid entry in its RCT.

The received data can be cached in the router's buffer with the reduced header as well. Upon further forwarding it, the router should lookup its RCT with the User/App supplied name indicated by the code in the compressed data header. The User/App supplied name is easily available to the router due to the mapping recorded in its LCT. If there is matched entry, the code originally contained in the data header is set as that indicated in the matched RCT entry. Otherwise, the content name in the data header is expressed by the User/App supplied name instead of the code.

The router can optionally decide whether to delete some entry in its LCT given the finite code length (i.e. 16bits). The removal of the LCT entry makes the router to send the **Code Deletion Message (CDM)** to its neighboring CCN routers to inform the removed pair of  $\langle code, User/App\ supplied\ name \rangle$ .

Although there are several messages introduced in our method, the router can avoid the frequent message exchanges by previously setting up the mappings between the code and User/App supplied name for the popular content in its LCT and notifying them to its upstream node.

While it is feasible to reduce the CCN data header by delivering it using the UDP protocol [6] where the content name can be replaced as the UDP port number, it can only save the network bandwidth but at the expense of introducing the complicated adaption of CCN data into the UDP segment. Especially, there is a potential competence between the CCN data and other existing UDP applications for the limited port numbers resource. However, our method can make the efficient use of the system resource by relying on the single UDP port dedicated for CCN communication in the architecture of CCN over UDP.

It is known that in-packet Bloom Filters [9] can also be used to reduce the data header. But the functions of Bloom Filters should be globally defined in such solution, which sets the barrier for its increasing deployment in the large-scale network. In comparison, our method just requires the negotiation between the routers neighboring to each other rather than the global coordination.

### 3.3 Integration of Two Proposed Methods

The two proposed methods can be used either together or individually in the CCN architecture. As the enhanced CCN Interest and data processing operates together with the header reduction method, the data delivery based on the compressed header is only implemented between the CCN routers where the data is to be cached. The intermediate routers without the need of caching can still take the traditional IP-policy reaction to the received CCN packets and there are no any operations relevant to the header reduction in these routers as well.

## 4 Simulation and Discussion

To have some basic indications of the efficiency in our solution, in terms of router's processing load, buffer space consumption and network bandwidth usage, we have run some early ns-2 simulations.

The simulation scenario is illustrated in Fig. 6. The content source disseminates the content item (i.e. a video clip) composed of the 1000 data pieces each with 1KB in length to the client along the path concatenated by 5 content routers. Each router caches the CCN data passing by it with the probability  $d$ .

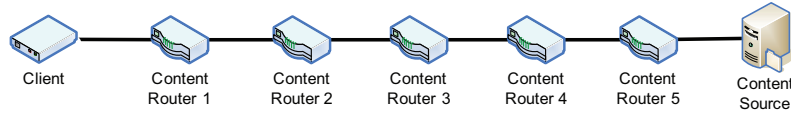


Fig. 6. Simulation Scenario

In our simulation, we firstly focus on the router's processing load at the CCN layer that is evaluated as the amount of operations on the received CCN data packets. Then, the router's buffer space and the network bandwidth consumed by the CCN data header are measured.

Fig. 7 illustrates the router's processing load occurring in the CCN approach and our proposal. It is observed that the router running the CCN approach deals with all the data pieces even it does not cache some of them. However, our solution just has the router process the data pieces cached in its buffer. Therefore, there is increasing reduction on processing load in our scheme with the falling cache probability  $d$ .

The router buffer space occupied by the content name of the cached data pieces is evaluated in Fig. 8. With the assumption that the content name is defined as the URL-like string with 50 bytes where 45 bytes constitute the User/App supplied name and 5 bytes represent the Versioning & Segmentation, our solution yields the reduction of the buffer space consumption by 86% in comparison to the CCN approach. The reduction comes from the fact that we use the 2-byte code to represent the User/App supplied name originally defined

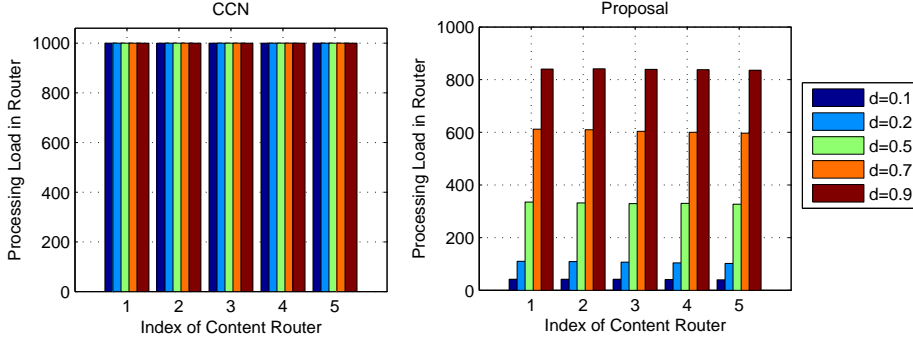


Fig. 7. Processing Load in Content Router

in CCN. The improvement is still obvious even taking into account the memory used by the LCT and RCT introduced in our solution because the space occupied by these tables is much less than those accommodating User/App supplied name component in tens or hundreds of data pieces.

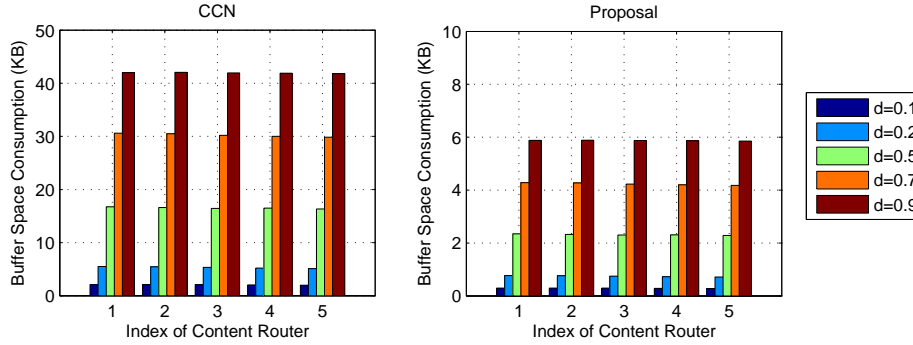


Fig. 8. Buffer Space Consumption of Content Name in Content Router

The network bandwidth consumed by the content name of data pieces is measured as the total size of the User/App supplied name components sent from the content source in Fig. 9. In order to take the insight into the impact of the content name on the network resource consumption, we illustrate the performance curves with different numbers of data pieces constituting the content item that are denoted by the X-axis in the figure where we still hold the assumption made in Fig. 8. It is noticed that our solution saves more network bandwidth as more data pieces are sent from the content source.

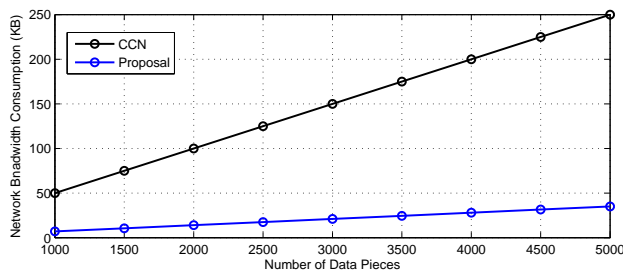


Fig. 9. Network Bandwidth consumed by Content Name

## 5 Conclusion

The CCN architecture presented in [6] is known as one of the promising candidates of building the future content centric network. However, it may require the potentially respectable resource consumption when operating as the overlay of the existing network architecture such as IP network. In this paper we have presented the solution constituting of two methods that can handle the CCN packets in the resource efficient way. The first method lessens the processing load of the router by making the router to avoid handling the data not to be cached in its buffer. Its principle is to configure the source and destination address in the underlying transmit unit (i.e. IP packet) encapsulating CCN packet based on the router's decision of caching CCN data. The second one replaces the URL-like content name in the CCN header with an integer to reduce the buffer size and network resource consumption. Two tables recording the mapping between the integer and the content name are defined to enable the proposed method can be performed locally without any global cooperation in the network.

In order to evaluate the solution, we created the extensive simulation based on ns-2. The simulation results disclose that our solution can alleviate the processing load in the router especially when the router caches less data in the buffer. The simulation also confirms that there is obvious reduction in the buffer space and network bandwidth consumption in the network with the massive content dissemination.

## References

1. Douglas E. Comer.: Internetworking with TCP/IP Vol.1: Principles, Protocols, and Architecture (4th Edition). Prentice Hall, Englewood Cliffs (2000)
2. Named Data Networking (NDN) Project, <http://www.named-data.net>
3. Tanenbaum, A.S.: Computer Networks. Prentice-Hall, Englewood Cliffs (2002)
4. AKARI, New Generation Network Architecture AKARI Conceptual Design. Project Description v1.1 (2008)
5. Gluhak, A., Bauer, M., Montagut, F., Stirbu, V., Johansson, M., Vercher, J., Presser, M.: Towards an Architecture for a Real World Internet. Towards the Future Internet. IOS Press, Amsterdam (2009)

6. Van Jacobson, Diana K. Smetters, James D. Thornton, Michael F. Plass, Nicholas H. Briggs, and Rebecca L. Braynard.: Networking Named Content. In: the 5th international conference on Emerging networking experiments and technologies, pp.1–12. ACM, New York (2009).
7. Yuncheng Zhu, Maoke Chen, Nakao, A.: CONIC: Content-Oriented Network with Indexed Caching, In: INFOCOM IEEE Conference on Computer Communications Workshops, pp.1–6. IEEE Press, New York (2010).
8. Teemu Koponen, Mohit Chawla, Byung-Gon Chun, Andrey Ermolinskiy, Kye Hyun Kim, Scott Shenker, and Ion Stoica.: A data-oriented (and beyond) network architecture. In: the 2007 conference on Applications, technologies, architectures, and protocols for computer communications, pp.181–192. ACM, New York, (2007).
9. Petri Jokela, Andr Zahemszky, Christian Esteve Rothenberg, Somaya Arianfar, and Pekka Nikander.: LIPSIN: Line speeds publish/subscribe inter-networking. In: the ACM SIGCOMM 2009 conference on Data communication, pp.195–206. ACM, New York, (2009).
10. 4WARD Deliverable D6.2, M. D.Ambrosio, M. Marchisio, V. Vercellone et al.,”Second NetInf Architecture Description”, Jan.2010.
11. Freedman, M.: Building a service-centric network with SCAFFOLD Princeton University (2010)
12. Reaz Ahmed, Raouf Boutaba.: Distributed pattern matching: a key to flexible and efficient P2P search, IEEE Journal on Selected Areas in Communications. 25, 73–83. (2007)
13. R. Ahmed, R Boutaba: Plexus: a scalable peer-to-peer protocol enabling efficient subset search, IEEE/ACM Transactions on Networking. 17, 130–143. (2009)
14. R. Moskowitz and P. Nikander.: Host Identity Protocol Architecture. RFC 4423, IETF. (2006)
15. Dirk Trossen, Mikko Sarela, and Karen Sollins.: Arguments for an information-centric internetworking architecture. SIGCOMM Comput. Commun. Rev. 40, 26–33. (2010).
16. Matthew Caesar, Tyson Condie, Jayanthkumar Kannan, Karthik Lakshminarayanan, and Ion Stoica.: ROFL: routing on flat labels. In: the 2006 conference on Applications, technologies, architectures, and protocols for computer communications, pp.363–374. ACM, New York. (2006).
17. P. Eugster, P. Felber, R. Guerraoui, and A.-M. Kermarrec.: The many faces of publish/subscribe. ACM Computing Surveys. 35, 114–131. (2003).
18. Bryan Ford and Janardhan Iyengar.: Breaking Up the Transport Logjam. In: HotNets-VII, 7th ACM Workshop on Hot Topics in Networks. ACM, New York. (2008).
19. J. Rosenberg: UDP and TCP as the New Waist of the Internet Hourglass, internet draft 2008. <http://tools.ietf.org/id/draft-rosenberg-internet-waist-hourglass-00.txt>.
20. L. Popa, A. Ghodsi, and I. Stoica.: HTTP as the Narrow Waist of the Future Internet. UCB Technical Report. (2010).
21. Berners-Lee, T. and Masinter, L. and McCahill, M.: RFC1738: Uniform Resource Locators (URL). (1994).
22. Mischke, J.; Stiller, B.: A methodology for the design of distributed search in P2P middleware. IEEE Network.18, pp. 30–37. (2004)
23. Oram, Editor, Peer-To-Peer: Harnessing the Power of Disruptive Technologies, O’Reilly & Associates, Sebastopol (2001).