19

Information-Centric Networking: The Case for an Energy-Efficient Future Internet Architecture

Mayutan Arumaithurai[1], Kadangode K. Ramakrishnan[2] and Toru Hasegawa[3]
[1]*Institute of Computer Science, Computer Networks Group, University of Goettingen, Goettingen, Germany*
[2]*Riverside Computer Science and Engineering, University of California, Riverside, USA.*
[3]*Information Networking, Osaka University, Osaka, Japan*

19.1 Introduction

The Internet was traditionally designed in a host-centric manner with its primary focus being the establishment of end-to-end connectivity between them. It was designed on the assumption that the network elements are always available (i.e., switched on and connected) and that end-to-end connectivity such as that provided by TCP/IP is sufficient to facilitate data transfer between two nodes. If an established end-to-end connectivity between two nodes is broken, the infrastructure primarily focuses on re-establishing the broken connection. In case the nodes are mobile, Mobile-IP [1] and related protocols further ensure that the communication (via TCP connection) is maintained even as the nodes move.

In reality, such a heavy focus on connection establishment is not necessary for all usage scenarios, especially in the case of data delivery and could result in inefficient use of network resources such as bandwidth and power. Moreover, the presence of intermediate nodes that facilitate end-to-end connectivity such as proxies, Home-Agents, and Network Address Translation (NAT) devices further complicate energy-saving approaches such as shutting down routers, network interfaces, or even switching off certain routing paths based on the network

---

Green Communications: Principles, Concepts and Practice, First Edition.
Edited by Konstantinos Samdanis, Peter Rost, Andreas Maeder, Michela Meo and Christos Verikoukis.

load. Instead, the focus should be on the data regardless of where it is obtained from. This would allow the requester to obtain the data from a cache/node that is fewer hops away from the original source of the data if it becomes available during an ongoing transfer. Moreover, in case the source of the content becomes unavailable (because it moved or switched off to save power) during an ongoing transfer, the requester could continue to download the remaining content from other sources. Predominantly battery powered hosts such as laptops, smartphones, and tablets that connect intermittently to the Internet and are not powerful enough to support too many parallel requests cannot be considered as reliable sources for content, especially a large piece of data. The influx of such devices further strengthens the need to focus on content instead of establishing and maintaining connection.

In short, users are primarily interested in obtaining content and do not care much about where they obtain the content from. But, the Internet as it is currently designed focuses on establishing end-to-end connectivity. In this chapter, we argue that energy optimization techniques applied on the current Internet infrastructure will not result in orders of magnitude increase in energy efficiency and that we need to consider the possibility of deploying future Internet architectures, namely Information-Centric Networking (ICN) [2–5]. We will first look at popular enhancements to the current design of the Internet that primarily focuses on the actual content retrieval, discuss the energy saving potential they pose and the reasons behind why they are limited in their functionality as a pure information-centric network and the lessons learnt. We will then shift our focus to the ICN paradigm that focuses primarily on content and helps to deliver content in a more intelligent manner. Finally, we discuss the potential it has in terms of energy efficiency, the challenges it imposes and present some use-scenarios.

19.2 Popular Content-Centric Enhancements

Though the Internet has been designed with a focus on end-to-end connectivity, many popular solutions exist that try to overcome this and turn the focus on content.

19.2.1 Peer-to-Peer

Peer-to-peer is a prime example of a content-centric approach where users interested in a particular content attempt to obtain it from other peers. Popular peer-to-peer services such as BitTorrent make use of a Tracker Server to store the mapping between available content and which of the peers have it. A peer interested in a particular content contacts a Tracker Server and obtains a list of peers that are serving that particular content. The advantage of peer-to-peer solutions is that the peers that are downloading a particular content can also choose to serve that content, thereby increasing the number of sources for a particular content. Peer-to-peer solutions thus provide users a wide range of options from where one could obtain the content. Peer-to-peer services also facilitate the possibility for a requester of content to obtain it from multiple sources simultaneously.

19.2.1.1 What is the Energy Saving Potential?

Peer-to-peer facilitates multiple nodes to participate in the redistribution of content and therefore make the content simultaneously available in multiple nodes. This increases the likelihood that a requester of content finds one or more optimal sources to download the data. The optimal source could be close enough to the requester compared to the original source, thereby providing the possibility to reduce the number of hops the data has to traverse. Moreover, the original source for the data need not be available 24 hours on-line because other peers that are available contribute their uplink capacity to support redistribution. Moreover, unlike end-to-end connectivity based data retrieval, peer-to-peer systems are resilient to churn, and, therefore, switching off hosts or network routers based on energy saving plans does not affect the data delivery significantly. Peer-to-peer-based data delivery is also resilient to changes in paths.

19.2.1.2 Why They are not Completely Effective as a Content-Centric Alternative?

The disadvantage of a peer-to-peer-based solution is that it is not topology aware and, therefore, proximity in terms of hops in the peer-to-peer topology does not in reality mean that they are close to each other in the routing topology. For instance, three peers that appear close to each other on the peer-to-peer topology could in fact be world apart with one of them being in the United States, another in Europe, and another in Japan. Therefore, in terms of the actual distance the content has to traverse, it might have to traverse a larger number of hops, thereby increasing energy consumption. To overcome the problem of topology unawareness, solutions such as Application Layer Transport Optimization (ALTO) [6] have been proposed. The ALTO servers are envisioned to have information about the network topology and other factors and, therefore, support the clients in the peer selection process. The ALTO-based solution looks promising, but cannot operate at small timescales because the updates it receives are usually averaged over larger timescales. Furthermore, the effectiveness of the ALTO solution depends on the level and accuracy of the information it obtains from the various network operators.

19.2.2 Content Delivery Network (CDN)

Content Delivery Networks or Content Distribution Networks (CDNs) are a distributed network of large storehouses for content and support the redistribution of content. The goal of a CDN is to serve content to end users with high availability and high performance. CDNs help users to obtain their content faster and reduce the load on the original source of content as well as on the network. CDNs are in fact a group of servers present in data-centers that cache and serve content such as downloadable files (movies, software, documents), web-objects (images,scripts, text), location-specific advertisements, and other static content. They are also used by content providers to serve live-streaming and video on demand. CDNs were usually deployed in backbone networks, but recently, network operators have been deploying smaller scale CDNs closer to the edge to optimize traffic in their network as well as to provide content providers an alternative CDN service.

19.2.2.1 What is the Energy-Saving Potential?

As CDNs are essentially data-centers serving content, energy optimization techniques used in data-centers are applicable. Cheaper/renewable energy sources, efficient cooling mechanisms, and efficient load distribution could be used for energy efficiency, which is difficult and infeasible for smaller/individual content providers. Moreover, due to the concentration of content, servers can serve multiple contents instead of having to be available 24/7 just serving a single piece of content regardless of the load. The advantage is that end-nodes that serve only one type of content can be switched off because their content is being served by a CDN server. Furthermore, based on load, multiple clusters could be switched off. Content could also be served from a closer CDN, thereby reducing latency and the number of hops.

19.2.2.2 Why They are not Completely Effective as a Content-Centric Alternative?

CDNs are application layer solutions, and, therefore, the client will have to establish connection to the content provider (e.g. HTTP), in order to receive a list of content and the corresponding CDN cache server where the content can be obtained from. Therefore, content providers might need to have their servers available for initial connection establishment and depend on CDNs to increase efficiency. Moreover, CDN-based solutions can be suboptimal because the CDN source has to be decided prior to the actual data transfer. A web server might place data in a CDN and request users to go there, but a CDN server close to the user might not be used because the web server did not store content there. Dynamically deciding which content to cache in which CDN server is not straightforward. In the case of mobile nodes, this could result in larger inefficiency because a Uniform Resource Identifier (URI) that has been resolved earlier to a particular CDN might not be the optimal one once a node moves and because no URI resolution is involved after movement, the nonoptimal CDN is being used, which could be a larger number of hops away.

19.2.3 Domain Name Systems (DNS)

DNS is another example of a content-centricity approach. The DNS stores mapping between a URL and the IP address where the content can be obtained. For instance, a user searching for “Google.com” can be redirected to any of the Google servers based on how the DNS is configured. The configuration could be such that the load is balanced or the request is redirected to the server closest to where the request was made. When an end user moves, he can renew his DNS request to receive a server close to him.

19.2.3.1 What is the Energy-Saving Potential?

The separation between end users to request for information and a mapping service that maps information to location/node has the possibility to save energy in terms of the number of hops the data has to traverse.

19.2.3.2 Why They are not Completely Effective as a Content-Centric Alternative?

Nevertheless, the problem with DNS is that, it is performed at the beginning and is rarely updated during the session. Moreover, DNS updates are not possible in shorter time frames and make sense only for big content providers.

19.3 ICN: Motivation

ICN, also known as Content-Centric Networking (CCN) [2, 3, 5], is a new paradigm for the future Internet where the network provides users with named content, instead of communication channels between hosts. It places the content at the forefront of the design of the architecture. Unlike current IP networks that support end-to-end connectivity, ICN networks route traffic based on content names. Therefore, instead of focusing on end-to-end connection and maintaining such connections even when data is available on a router/node along the path, ICN allows us to focus on obtaining content. In the case of an IP-based end-to-end connection scenario where multiple recipients are interested in the same content, the content is delivered only from the source on a per flow basis even if the flows traverse the same intermediate nodes. Intermediate routers will therefore end up processing the same request and data multiple times in the case of IP. In the case of mobile nodes, as a mobile node moves, it can continue to obtain the data from the same source or some other efficient source with the help of ICN. It does not have to use a complex protocol like Mobile IP that involves a lot of signaling just to maintain end-to-end connectivity. To summarize, ICN is content-centric; facilitates easy caching and obtaining data from closer sources; and supports obtaining parts of the data from multiple sources at the same time.

ICN with its focus on content changes routing completely. The routing is done such that the request is sent to the closest source. Such an optimization at the routing level facilitates the introduction of drastic changes in energy efficiency. At the routing layer, the decisions made are topology-aware unlike peer-to-peer or CDN-based solutions. Here, in-path caches that serve the content can further improve latency, reduce unnecessary usage of network resources, and thereby result in energy efficiency. Dedicated resources such as CDNs and routers with a larger cache could compliment ICN and help further improve energy efficiency.

As explained in Section 19.5, ICN has the potential to introduce significant changes in energy consumption and energy saving mechanisms. The energy saving mechanisms described in other chapters of this book for the current Internet might also be applicable in an ICN-based environment. For instance, shutting down unused network cards or devices could be used in ICN too. In fact, ICN might make it easier to deploy many of these energy saving mechanisms. As ICN is a new and evolving technology, there is a lot of scope to include energy efficiency in the design phase as being done in the GreenICN joint project between the European Union and Japan [7].

19.4 ICN: Background and Related Work

Research on ICN is at an early stage, with many key issues still open, including naming, routing, resource control, security, privacy, and a migration path from the current Internet. Next, we list some of the interesting ongoing works related to ICN.

19.4.1 Named Data Networking (NDN)

Named Data Networking (NDN)[1] [3], originally known as Content-Centric Networking (CCN) [2], is a popular ICN protocol where content/information is looked up and delivered

---

[1] http://named-data.net/

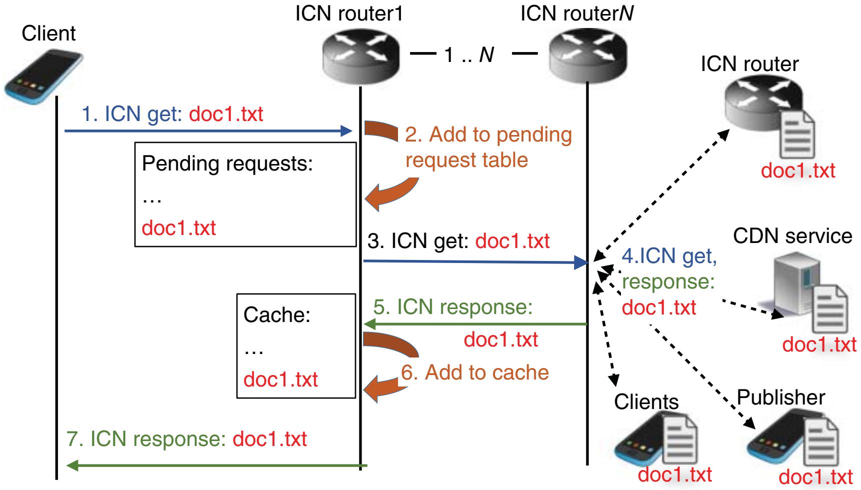


Figure 19.1 Message flow highlighting the name-based data retrieval in ICN. Requested data can be obtained from one of the multiple sources of the data. In this case, the data can be obtained from the cache of an ICN router, from a CDN service, from other clients, or directly from the original publisher (see Step-4). Step-2 shows that the `Interest` is added to the Pending Interest/Request Table (PIT), and in Step-6 we can observe that the data is stored in the cache of the router before being forwarded

according to its name without knowing the identity and location of the sender. NDN uses two packet types, `Interest` and `Data`. A consumer queries for named content by sending an `Interest` packet; a provider in turn responds with a `Data` packet. NDN requires a new forwarding engine instead of IP, which contains the Forwarding Information Base (FIB), Content Store (buffer memory which caches content), and Pending Interest Table (PIT). FIB is used to forward `Interest` packets toward potential source(s) of matching data. PIT keeps track of bread crumbs of `Interest` (i.e., to support reverse-path forwarding), which the `Data` packets follow to reach the original requester(s). If multiple `Interest` packets arrive for the same data from multiple end-nodes, they will be aggregated in the PIT and served when the data arrives. The Content Store maintains a cache of the data in order to satisfy potential future requests for that data.

Figure 19.1 shows a simple message flow in an NDN architecture. Let us assume that the requester would like to get a movie file (doc1.txt) that is published by the publisher. In an IP network, the requester would make a DNS request to identify the IP address of the publisher and issue a request that would go all the way to the publisher. On the other hand, in an ICN network, the requester would issue a request (Step-1, i.e., ICN Get) for that movie file to the ICN network. The first hop router has multiple options to deal with this request. If the movie is present in its cache, it could deliver it directly from it. Else, an ICN router could choose to forward it via any one of the paths such as those depicted in Figure 19.1 where one path leads to a CDN like content store, another to other clients that have the data, another to an ICN router cache, and another to the publisher.

19.4.2 Content-Oriented Publish/Subscribe System (COPSS)

Content-Oriented Publish/Subscribe System (COPSS) [5] was proposed as an enhancement of NDN to provide push-based pub/sub multicast capability in ICN. COPSS allows subscribers interested in a particular type of content or topic to issue a request for subscription. The COPSS network handles the delivery of the content to the interested subscribers when publishers publish content for these topics. The advantage of COPSS compared to NDN is that the routers need not maintain per packet state as being done by the PITs. Instead, they have to maintain a state per subscription in the Subscription Table (ST). A Rendezvous Point (RP) forms the root of the tree to which subscriptions are sent. Publishers forward the data toward the RP, which in turn forwards it to the subscribers via the ST. COPSS extends the naming framework with the introduction of the concept of hierarchical content descriptors to enable efficient large-scale information dissemination. COPSS also provides support for a hybrid environment [8, 9] that comprises nodes belonging to both IP and ICN nodes that can choose to function with full or partial ICN capability.

19.4.3 Projects Supported by the European Union

The Publish Subscribe Internet Routing Paradigm (PSIRP) [10] project developed an information-centric network architecture based on a publish/subscribe paradigm. PSIRP proposed to replace the current Internet protocols entirely, applying a layer-less clean-slate architecture for routing, security, mobility, and other basic network services. This was followed by the PURSUIT project [11] to address open issues such as resource control and advanced concepts for information scoping. PSIRP/PURSUIT introduced several contributions on several aspects of ICN, for example, publish/subscribe architecture (e.g., Refs. [11, 12]), fast-forwarding strategies (e.g., Ref. [13]), a new transport layer protocol (e.g., Ref. [14]), and mobility support (e.g., Ref. [15]).

The 4WARD project [16] developed an ICN architecture called Networking of Information (NetInf) [17]. It has an object model that can handle information at different abstraction levels, enabling the referencing of information independent of its encoding. The NetInf naming scheme provides name–data integrity and name persistency. The SAIL project [18] continued developing NetInf where 4WARD left off. For instance, the naming scheme was revised, the object model was simplified, and the routing and name resolution framework has become more concrete with, for example, an inter-domain interface.

The Architecture and Applications of Green Information Centric Networking (GreenICN) project [7] is a relatively new project that was started in April 2013 and is supported by both the European Union and Japan. GreenICN plans to build an energy-efficient content-centric network from the onset instead of treating it as an after-thought. GreenICN aims to perform an application-driven design, with disaster and large-scale video delivery as the chosen key application scenarios. The aftermath of a disaster introduces challenges in terms of data delivery as well as efficient use of scarce resources such as power. Many of the functioning devices such as base stations and mobile terminals are primarily driven with the use of batteries. GreenICN aims to efficiently distribute disaster notification and critical rescue information, with its ability to exploit fragmented networks with only intermittent connectivity, while ensuring that energy is efficiently consumed. Video delivery on the other hand introduces issues of scale in terms of network traffic, efficient use of caching, and load on the various nodes. GreenICN also aims to investigate migration from IP to a pure ICN architecture.

19.4.4 Internet Research Task Force (IRTF)

The IRTF [19] is associated with the Internet Engineering Task Force (IETF)[20]. It promotes research of importance to the evolution of the Internet by creating focused, long-term Research Groups working on topics related to Internet protocols, applications, architecture, and technology. Recently, Information-Centric Networking Research Group (ICNRG)[21] was started with the main objective of coupling ongoing ICN research in the aforementioned areas with solutions that are relevant for evolving the Internet at large. The ICNRG serves as a forum for exchange and analysis of ICN research ideas. Its current goals are to produce documents that further the understanding of the current state of the art and identify the research challenges. In-network caching techniques have also been investigated in the IETF for potential standardization in the Decoupled Application Data Enroute (DECADE) Working Group [22] and might be pursued in the ICNRG Working Group too. The Light-Weight Implementation Guidance (lwig) [23] focuses on small devices and work related to energy efficiency.

19.4.5 ICN-Related Research papers

DONA [4] was one of the first, clean-slate, ICN proposals. DONA uses flat, self-identifying and unique names for information objects and binds the act of resolving requests for information to locating and retrieving information. The authors of Ref. [24] investigate the use of CCN in the case of real-time applications such as audio-conference, while the authors of Ref. [25] investigate the use of COPSS for a gaming application with stringent requirements. CONIC [26] is a network architecture designed for efficient data dissemination using storage and bandwidth resources in end-systems (i.e., available storage located in end hosts is used for caching). A similar approach, where content is cached in routers, is the Cache and Forward architecture [27]. MultiCache [28] is an information-centric overlay network architecture aiming to improve network utilization via resource sharing. In MultiCache, network operators deploy and control proxy overlay routers that enable the joint provision of multicast and caching, targeting both synchronous and asynchronous requests.

19.5 ICN: Energy Efficiency

19.5.1 Content-Centric Routing

Content-centric routing ensures that the routers are able to make an informed decision on what the user is interested in. Based on the request, the routers are able to forward it to the closest source possible. Such a service is naturally resilient to energy saving techniques such as switching off network elements and can react to changes in the routing path at quicker time scales. It provides services similar to peer-to-peer at the routing layer by facilitating the recipient to receive from one or more of the multiple sources (including in-path) caches; thereby enhancing efficiency, reducing latency, and control by relying on topology-based decisions.

Furthermore, it allows easy integration of CDN like services at the routing layer. It will reduce the overhead and complexity involved in unnecessarily re-routing the data, especially in the case of Mobile IP, to maintain existing connections. This could result in the reduction of latency and also help find optimal routes that need not go via predetermined proxy nodes. Similar to peer-to-peer, nodes that are currently downloading content could also double up as content providers because they are in any case switched on.

19.5.2 Reduction in the Number of Hops

Reduction in the number of hops implies that less nodes are used to process the request as well as data, thereby reducing the overall energy consumption in the network on a per request/data level. The ICN architecture, as mentioned earlier, has in-path caching, that is, routers along the way are capable of caching content. Therefore, popular content that has multiple requests at the same time might find a close enough cache, thereby reducing the number of hops the request and the data have to traverse. The in-path caching support could be more efficient than a CDN like solution because the data storage is available on the path from the requester to the data source and need not take a different path. The data source in this case could be either another peer or a CDN service. In case there are multiple sources, a closer source can be utilized to serve the data, thereby reducing the number of hops.

In-path delivery also ensures that in the case of mobility when a node moves, the request would find a closer source from the requester at the new location to the previous data source. Moreover, when a mobile node moves, one need not worry about maintaining the same IP, Mobile IP, and so on. What one needs to do is find a closer source for that data and continue delivering it. Even if there is only one source, signaling can be avoided and a new request from the new location is sent. Reduction in signaling messages after node movement also has energy saving potentials.

Figure 19.2 illustrates an example scenario where a client is seen to be requesting for the data `doc1.txt` in an IP as well as an ICN scenario. In the case of IP (Figure 19.2a), we can observe that the client receives the URL of the server in which `doc1.txt` could be found from the Google search engine (Step-2) and uses this URL to obtain the IP address of that server

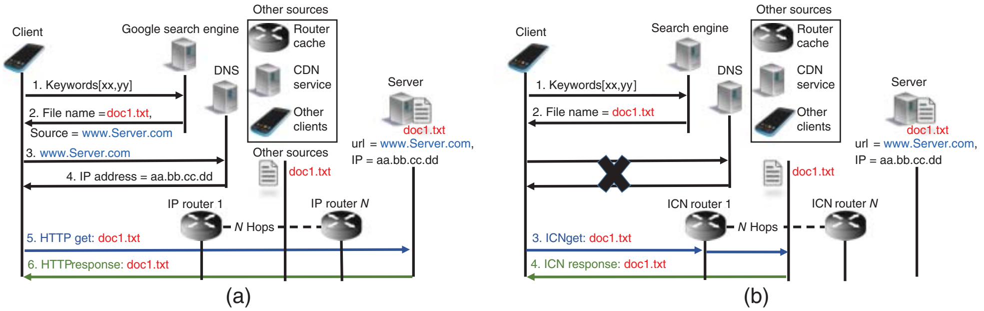


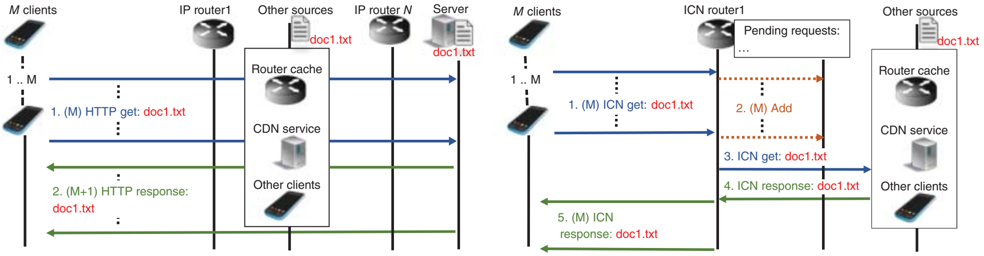
Figure 19.2 Message flow for obtaining data in a standard IP and ICN environment. (a) IP message flow: A standard HTTP Get command for data `doc1.txt` issued from a client to a server. The server receives the request even if other sources of that data exist closer to the client. (b) ICN message flow: A standard ICN Get command for data `doc1.txt` being issued to the ICN network and the first hop ICN router directing it to a closer source for that data

from the DNS server (Step-3, Step-4). It then issues an HTTP Get command for `doc1.txt` to the IP address returned by the server. This request is forwarded by the IP routers to the server mentioned in the IP header. It can be seen that even in the presence of other sources for the same data at a closer location (lesser number of hops) to the requester, the HTTP Get message is forwarded to the server. In the case of ICN (Figure 19.2b), we can observe that the search engine just returns the name of the data, that is, `doc1.txt` and does not provide a location or a URL. The client then issues an ICN Get command with the name of the data to the first hop ICN router that in turn forwards it to one of the sources that is closer (in terms of hops) to the requester, instead of forwarding it to the original source of the data. This is feasible because the ICN router does name-based forwarding and is aware of the name of the data required and where in the network is that data available. The energy efficiency achieved in this case can be represented by:

$$\eta \propto (n_1 rD + nrS + nrs) \tag{19.1}$$

where η is energy saved, n_1 is the number of hops to the DNS server, D is the cost at the DNS server to process the query, r is the cost incurred at every router to process the request or the data, n is the difference in the number of hops that the HTTP Get traverses to reach the server versus the number of hops that the ICN request travels to reach a closer source, s is the size of the request, and S is the size of the data. If we assume that the requested data file is a large video file or that the number of hops (n) is large, the gain could be considerably high.

Figure 19.3 illustrates an example scenario where multiple clients are seen to be simultaneously requesting for the same data `doc1.txt` in an IP as well as an ICN scenario. In the case of IP (Figure 19.3a), we can see that all the simultaneous requests are being forwarded by the IP routers to the server mentioned in the IP header. It can also be seen that even in the presence of other sources for the same data at a closer location (lesser number of hops) to the requester, the HTTP Get message is forwarded to the server. In the case of ICN (Figure 19.3b), we can observe that the first hop ICN router is adding all the requests to a `Pending Request Table` and forwarding just one of the requests to a source that is closer (in terms of hops) to the requesters. This is again feasible because the ICN router does name-based forwarding and is aware that the multiple requests from different clients are for the same data by looking



(a) (b)

Figure 19.3 Message flow comparing the case of multiple simultaneous HTTP Get versus ICN Get. (a) IP message flow: Multiple simultaneous HTTP requests for the same data `doc1.txt` is forwarded to the server. Therefore the server responds with M+1 copies of the same data. (b) ICN message flow: Multiple simultaneous ICN requests for the same data `doc1.txt` is added to the `Pending Request Table` and only one ICN request is forwarded to the closest source at the name of the required data. Energy efficiency represented by Eq. (19.1) can therefore be updated to:

$$\eta \propto (n_1 rD + (M-1)nrS + (M-1)nrs - (M-1)p) \tag{19.2}$$

where η is energy saved, M refers to the number of parallel requests, p is the additional cost incurred to add the M requests to the `Pending Request Table`, n_1 is the number of hops to the DNS server, D is the cost at the DNS server to process the query, r is the cost incurred at every router to process the request or the data, n is the difference in the number of hops that the HTTP Get traverses to reach the server versus the number of hops that the ICN request travels to reach a closer source, s is the size of the request, and S is the size of the data. If we assume that M, that is, the number of parallel requests is very high, the gain could be considerably high.

19.5.3 Caching

ICN, by allowing for in-path caching as well as dedicated caching, allows recipients to access the data from closer sources. This could result in the reduction of the number of hops as well as the latency as mentioned earlier. For instance, in Figure 19.1, let us consider the scenario that multiple users are interested in the same content and their path to the source of the content is the same. In the case of IP, all the routers along the path will see the different connections and process it, but in fact the number of bits (i.e., content) transferred is the same. Whereas, in the case of ICN, the first hop router will realize that in fact all the connections are requesting for the same content and will either serve them from the cache or just forward one request upward. This saves computing resources and thereby energy on routers above it.

Furthermore, by allowing for large and concentrated caches on routers and services similar to CDN that are either in-path or closer to the source, one could also optimize energy by using means that are used in data-centers. For instance, servers, caching capability in routers can be brought up/down based on demand. By serving a large variety of content from such caches, the servers that are the actual source of content can be switched off. Efficient cooling and power mechanisms that are easily deployable in larger scales could be used, because these mechanisms are not straightforward for smaller content providers.

Caching policies could also be used to increase energy efficiency. For instance, policies that facilitate the switching ON/OFF caches on routers based on demand might be useful. Caching policies that are able to perform cooperative caching, that is, neighboring router caches serving complimentary data instead of the same data, could increase the likelihood of content hit in the vicinity, thereby reducing the number of hops.

It must be noted that caching introduces an additional burden of energy consumption on the routers, not only to store and save the data, but also to perform operations such as search for every request that passes it. Therefore, caching policies and algorithms must take this into account while devising the right solution.

Figure 19.4 illustrates an example scenario where multiple clients are seen to be requesting for the same data `doc1.txt` at different time periods in an IP as well as an ICN scenario. In the case of IP (Figure 19.4a), we can see that all the requests are being forwarded by the first hop IP router to the server mentioned in the IP header. It can also be seen that even in the presence of a cached version of that data in the IP router, the IP router is not able to serve that request because it is not aware of the requested data but just the IP address of the server. In the case of ICN (Figure 19.4b), we can observe that the first hop ICN router is adding the

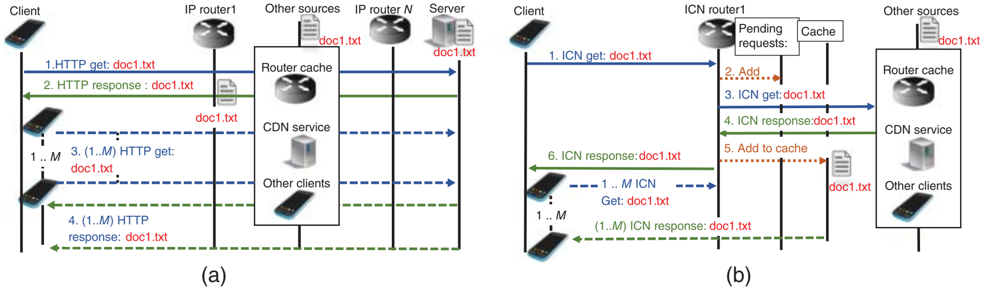


Figure 19.4 Message flow comparing a standard HTTP Get versus ICN Get. (a) IP message flow: Multiple requests for the same data `doc1.txt` is being forwarded to the server. The server therefore responds with a copy of the data for every single request received. (b) ICN message flow: Multiple requests for the same data `doc1.txt` is being served by the first hop ICN router because it has the data in its cache after the first interaction

data to its cache after the completion of the first interaction. It is, therefore, able to serve the requests arriving later from its own cache. This is again feasible because the ICN router does name-based forwarding and is aware that the multiple requests from different clients are for the data that it has in its cache. Energy efficiency represented by Eqs. (19.1) and (19.2) can, therefore, be updated to:

$$\eta \propto (n_1 rD + (M-1)nrS + (M-1)nrs - c) \tag{19.3}$$

where η is energy saved, c refers to the energy expended in the router's cache as well as to perform cache lookup. M refers to the number of parallel requests, n_1 is the number of hops to the DNS server, D is the cost at the DNS server to process the query, r is the cost incurred at every router to process the request or the data, n is the difference in the number of hops that the HTTP Get traverses to reach the server versus the number of hops the ICN request traveled to reach the first hop ICN router (i.e., 1), s is the size of the request, and S is the size of the data.

19.5.4 Seamless Support of Network Operations for Energy Efficiency

Currently, switching off nodes and/or network elements implies that a lot of signaling needs to be performed to ensure that the current state is transferred to another serving node or could result in the termination of the ongoing communication. For instance, in the case of Follow the Sun or Follow-me cloud service where data-centers that are closer to active parts of the world are switched on while data-centers away from active parts are switched off could result in the ongoing connection being terminated. Similarly, base stations, mobile gateways, service gateways, and other management servers in the case of a 3GPP network could be switched off when load is low. An ICN-based network allows for such network management operations by seamlessly adapting to the state of the network and forwarding the request to the suitable node.

Figure 19.5 illustrates an example scenario where a client is involved in a long-term transaction (e.g., a video stream or the transfer of a huge data file) in IP as well as ICN scenario. In the case of IP (Figure 19.5a), we can see that the appearance of a new data source at a closer location does not affect the ongoing communication. In the case of ICN (Figure 19.5b), we can observe that the requests are seamlessly sent to the new data source after time X. This allows

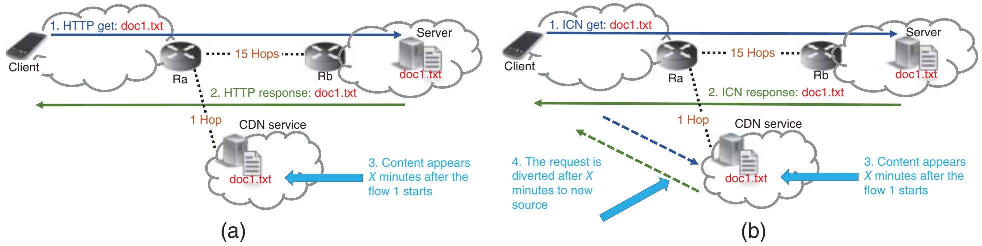


Figure 19.5 Message flow comparing a continuous HTTP Get versus ICN Get in the presence of a new data source (at Time *X*). (a) IP message flow: The requests are sent to the original server even after a new source for the data appears. (b) ICN message flow: The requests are forwarded to the CDN source after time *X* because it is closer in a seamless fashion

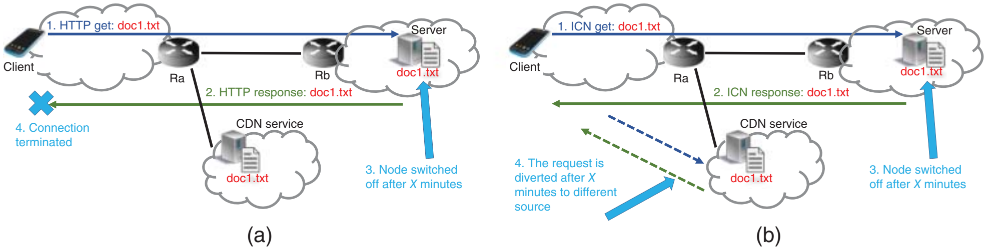


Figure 19.6 Message flow comparing a continuous HTTP Get versus ICN Get in the absence of the initial data source (at time *X*). (a) IP message flow: The connection is terminated once the source switches off after *X* minutes. (b) ICN message flow: The requests are forwarded to the CDN source after time *X* seamlessly

for greater flexibility in adapting to network conditions and making efficient use of network resources.

Figure 19.6 illustrates an example scenario where a client is involved in a long-term transaction (e.g., a video stream or the transfer of a huge data file) in IP as well as ICN scenario and the initial data source is switched off at time *X*. In the case of IP (Figure 19.6a), we can see that the disappearance of the initial data source at time *X* results in the termination of the ongoing communication. In the case of ICN (Figure 19.6b), we can observe that the requests are seamlessly sent to another data source after time *X* when the initial source is no longer available. This again illustrates the flexibility in adapting to network conditions and making efficient use of network resources.

Figure 19.7 illustrates an example scenario where a client is involved in a long-term transaction (e.g., a video stream or the transfer of a huge data file) in IP as well as ICN scenario and the client moves to a different base station/operator/location at time *X*. In the case of IP (Figure 19.7a), we can see that the connection to the initial data source is still maintained even if there is a significant increase in the number of hops. In the case of ICN (Figure 19.7b), we can observe that once the client moves at time *X*, the requests are seamlessly sent to the closer data source. This again illustrates the flexibility in adapting to network conditions and making efficient use of network resources to reduce the number of hops as well as latency for the end user.

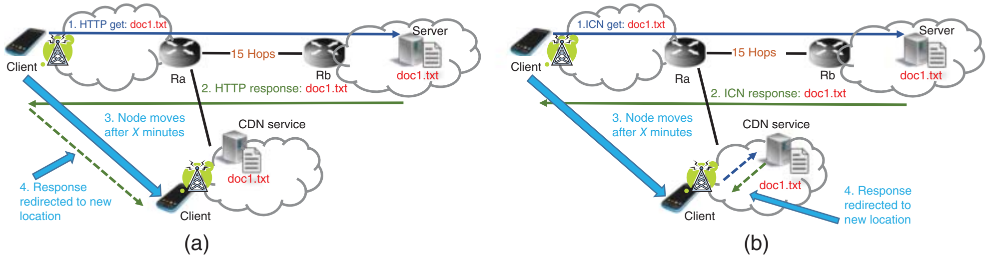


Figure 19.7 Message flow comparing a HTTP Get versus ICN Get when the node moves from one base-station/operator/location to another (at time *X*). (a) IP message flow: Though the client has moved, the connection to the initial data source is still maintained, resulting in inefficient usage. (b) ICN message flow: The requests are forwarded to the CDN source that is closer to the new location seamlessly

19.5.5 Coexistence with IP and Other Technologies

ICN is being designed not only to be incrementally deployed, but also to coexist [8, 9] with existing technologies to reap their advantages. ICN routers due to their increased complexity might consume more energy per query they handle compared to IP an router. This is due to the fact that ICN routers have to perform name-based lookup whereas IP routers do longest prefix match of an IP address. Moreover, a large amount of effort has been expended to optimize the IP routers to use the hardware very efficiently. ICN on the other hand is still in its nascent stage and would require a lot of efforts to standardize as well as optimize. During this period, as suggested in Ref. [8, 9], the ICN router could coexist with IP and use IP-based forwarding wherever possible. Note that though ICN might require more energy at a router level compared to that of IP, as we argue in this chapter, the ICN architecture on the whole might be more energy efficient.

19.6 Summary

This chapter introduces ICN, a future Internet architecture currently being developed. It details upon how the design of ICN is influenced by current workarounds that exist in the Internet infrastructure to provide content-centric features. The content-name-based routing allows for the possibility to make radical changes in how energy efficiency is achieved. The key reasons why ICN is a good candidate for energy-efficient protocol is that (i) ICN can react dynamically to the presence of data in a closer location, thereby reducing hops; (ii) ICN dynamically and seamlessly reacts to changes in path; and (iii) ICN's name-aware routing increases the flexibility of the network.

But one must also remember that the name-based routing might introduce additional work load on the routers, resulting in the need for more energy consumption. As technology advances, and the ICN protocol becomes standardized, routers will be designed to handle ICN more efficiently.

ICN and energy efficiency in ICN are at a very nascent stage. GreenICN [7], an EU project started recently, attempts to solve many of these issues and to obtain a better understanding of the energy saving potential of ICN.

References

[1] C. Perkins, “IP Mobility Support for IPv4,” IETF Request for Comments (RFC 3220), 2002.

[2] V. Jacobson, D. K. Smetters, J. D. Thornton, M. F. Plass, N. H. Briggs, and R. L. Braynard, “Networking named content,” in *CoNEXT*, Rome, Italy, 2009.

[3] L. Zhang, D. Estrin, J. Burke, V. Jacobson, and J. Thornton, “Named Data Networking (NDN) Project,” PARC, Tech, https://www.parc.com/publication/2709/named-data-networking-ndn-project.html, 2010.

[4] T. Koponen, M. Chawla, B.-G. Chun, A. Ermolinskiy, K. H. Kim, S. Shenker, and I. Stoica, “A data-oriented (and beyond) network architecture,” in *SIGCOMM*, Kyoto, Japan, 2007.

[5] J. Chen, M. Arumaithurai, L. Jiao, X. Fu, and K. K. Ramakrishnan, “COPSS: an efficient content oriented Pub/Sub system,” in *ANCS*, Brooklyn, NY, USA, 2011.

[6] J. Seedorf and E. Burger, “Application-Layer Traffic Optimization (ALTO) Problem Statement,” RFC 5693, October 2009.

[7] GreenICN, “Architecture. and Applications of Green Information Centric Networking (GreenICN),” project website: www.greenicn.org. [Accessed 10 January 2015].

[8] J. Chen, M. Arumaithurai, X. Fu, and K. K. Ramakrishnan, “Coexist: integrating content oriented publish/subscribe systems with IP,” in *ANCS*, Austin, Texas, USA, 2012.

[9] J. Chen, M. Arumaithurai, X. Fu, and K. K. Ramakrishnan, “Coexist: a hybrid approach for content oriented publish/subscribe systems,” in *ICN*, Helsinki, Finland, 2012.

[10] C. Esteve, F. Verdi, and M. Magalhaes, “Towards a new generation of information-oriented internetworking architectures,” in *ReArch*, Madrid, Spain, 2008.

[11] PURSUIT. Publish-Subscribe Internet Technology, project website: www.fp7-pursuit.eu. [Accessed 10 January 2015].

[12] D. Lagutin, K. Visala, and S. Tarkoma, Valencia FIA Book 2010 Publish/Subscribe for Internet: PSIRP Perspective, IOS Press, 2010.

[13] P. Jokela, A. Zahemszky, C. E. Rothenberg, S. Arianfar, and P. Nikander. LIPSIN: line speed publish/subscribe internetworking. In *Proceedings of ACM SIGCOMM’09*, Barcelona, Spain, August 2009.

[14] V. Koptchev and V. Dimitrov, “Traffic and congestion control in a publish/subscribe network,” in *Proceedings of CompSysTech’10*, Sofia, Bulgaria, June 2010.

[15] X. Vasilakos, V. A. Siris, G. C. Polyzos, and M. Pomonis, “Proactive selective neighbor caching for enhancing mobility support in information-centric networks,” in *Proceedings of ACM SIGCOMM ICN*, Helsinki, Finland, August 2012.

[16] 4WARD, “The FP7 4WARD Project,” http://www.4ward-project.eu/. [Accessed 10 January 2015].

[17] B. Ahlgren, M. D’Ambrosio, C. Dannewitz, M. Marchisio, I. Marsh, B. Ohlman, K. Pentikousis, R. Rembarz, O. Strandberg, and V. Vercellone, “Design considerations for a network of information,” in *ReArch*, Madrid, Spain, 2008.

[18] “SAIL project website,” http://www.sail-project.eu/.

[19] IRTF, “Internet Research Task Force (IRTF),” http://irtf.org. [Accessed 10 January 2015].

[20] IETF, “The Internet Engineering Task Force (IETF),” http://ietf.org. [Accessed 10 January 2015].

[21] IRTF, “Information Centric Networking Research Group (ICNRG),” http://irtf.org/icnrg.

[22] IETF, “Decoupled Application Data Enroute (DECADE),” https://datatracker.ietf.org/wg/decade/charter/. [Accessed 10 January 2015].

[23] IETF, Light-Weight Implementation Guidance (lwig), “http://datatracker.ietf.org/wg/lwig/charter/.” [Accessed 10 January 2015].

[24] Z. Zhu, S. Wang, X. Yang, V. Jacobson, and L. Zhang. “ACT: audio conference tool over named data networks,” in *Proceedings of ACM SIGCOMM ICN*, Toronto, Canada, August 2011.

[25] J. Chen, M. Arumaithurai, X. Fu, and K. K. Ramakrishnan, “G-COPSS: a content centric communication infrastructure for gaming,” in *ICDCS*, Macau, China, 2012.

[26] Y. Zhu, M. Chen, and A. Nakao, "CONIC: content-oriented network with indexed caching," in *INFOCOM IEEE Conference on Computer Communications Workshops, 2010*, pp. 1–6, San Diego, CA, USA, 15–19 March 2010.

[27] L. Dong, H. Liu, Y. Zhang, S. Paul, and D. Raychaudhuri, "On the cache-and-forward network architecture," in *IEEE International Conference on Communications, ICC '09*, Dresden, Germany, June 2009.

[28] K. Katsaros, G. Xylomenos, and G. C. Polyzos, MultiCache: an overlay architecture for information-centric networking, Comput. Networks., vol. 55, no. 4, pp. 936–947, 2011.