D1.3 Named Data Networking

Technology Assessment

December 2011
Version 1.0
Ronald van der Pol

SARA

1 Introduction

Named Data Networking (NDN) [1] is a new way of networking and data distribution and it is a clean slate alternative for TCP/IP. In NDN data has a unique name and it is independent from a location. Instead data is cached and replicated within the network and addressed by its unique name. The basic idea behind NDN is to put data (or content) central instead of having hosts and servers as the central entities that hold, fetch and deliver content. Currently, content is downloaded by setting up a connection to a server that holds that content. In NDN, users request content by including the name of that content in a query packet that is sent on the network. The content is delivered from the nearest place in the network that has the content with that name.

This concept is used in various projects under various names, e.g. Content Centric Networking (CCN) and Information Centric Networking (ICN). CCNx [3] is an open source implementation of CCN developed at PARC (Palo Alto Research Center, a Xerox company). Van Jacobson of PARC has been promoting the CCN project for many years. In 2006 he gave a Google TechTalk [4] in which he describes what he thinks are the major problems in today's networking and his ideas about how to solve them with CCN.

In August 2011 the NDN proposal received a three year grant [2] from the National Science Foundation (NSF) in the USA. Project partners include a.o. Van Jacobson (PARC), Lixia Zhang (UCLA) and KC Claffy (UCSD).

Section 2 gives an overview of the basic CCNx concepts and section 3 gives the current status of NDN/CCN/ICN. Finally, the conclusions are in section 4.

2 Overview of CCNx

CCNx has two packet types. An *interest* packet is sent by a consumer to request content from the network. A producer sends a response back to the consumer in the form of a *data* packet. This is shown in figure 1. Both *interest* and *data* packets contain the unique *Content Name* of a piece of content. The *Data* packet also contains the piece of content itself and a signature that is used for authentication. The signature is calculated over the content name, the data and the signed info (e.g. publisher ID, key location). This signature can be used to verify that the name and content correspond to each other and that both are not tampered with.

The goal of CCNx is to change three paradigms of today's networking:

naming Currently, when a user wants to get some piece of content, he or she connects to a server (e.g. a website) to download the content. The user has a conversation with a server. This conversational model dates back to the days when networking was done by using telephone circuits. Many current day applications still use this connection oriented model (e.g. by using TCP), although it often does not fit the problem any more. The problem has changed. Users want to watch videos, listen to music, read newspapers and magazines and they do not think of it as connecting to a specific server and download the content. They just want the content without having to know where is sits. But underneath, the network is still setting up connections to servers. CCNx wants to get rid of this conversational model. In CCNx content can

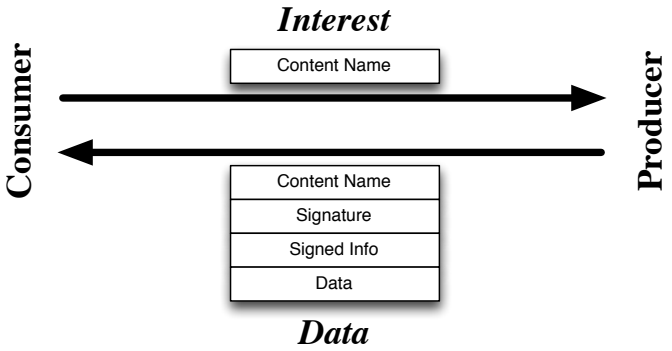


Fig. 1. *interest* and *data* packet types.

sit anywhere in the network and the content is delivered to the user from the topologically closest place. In the connection oriented model of the current internet the identifiers used are the names (addresses) of clients and servers. In CCNx the identifiers used are the names of pieces of data.

memory In a CCNx network there are explicit buffers in the network that store content. Users receive content from a buffer that is closest to them. In a TCP/IP network buffers are avoided as much as possible because they can have a very bad impact to end-to-end latency. Jim Gettys is one of the people who are studying the effect of too large buffers in the internet, which he calls *bufferbloat* [5]. Getting good TCP throughput, especially over long distances with large round-trip times is not a trivial task. This is partly due to the fact that TCP works with end-to-end retransmission. When a packet is lost, the receiving end informs the transmitting end that a retransmission is required. When the round-trip time is large, it takes time before the transmitting side is notified. In CCNx there is no end-to-end retransmission like there is with TCP, but a per link retransmission. This is especially beneficial in WIFI networks. Retransmission only needs to done in the radio part (which is possible because the CCNx base stations are likely to have the content still in their buffer), not over the complete end-to-end path.

security Most of the content is currently not secured in the sense that the content itself is encrypted, nor does it contain information about its authenticity. Instead, the connection that is used to retrieve the content is secured by encrypting it with e.g. SSL (Secure Socket Layer) [6] or TLS (Transport Layer Security) [6]. In CCNx the content itself is secured by including a signature over the name and content in the packet and optionally encrypting each piece of content.

CCNx forwarders are the equivalent of internet routers. Figure 2 shows the main components of such a CCNx forwarder. The Content Store is a large memory buffer that contains named content. Traditional internet routers only have a couple of MB of buffer memory. A CCNx forwarder will have a much larger buffer. The reasoning behind this is that memory and (SSD) disks are cheap enough to justify this choice. CCNx forwarders could have slots in which blades with large memories can be inserted. When an *interest* packet arrives, the name in the *interest* packet is looked up in the Content Store. When it is present in the Content Store a *data* packet with that content is sent to the interface where the *interest* packet arrived on. When the content is not present in the Content Store, the name in the *interest* packet is stored in the Pending Interest Table (PIT) together with the interface on which the *interest* packet arrived. Over time, a name can be associated with multiple interfaces when multiple *interest* packets are received.

The FIB (Forwarding Information Database) is used by the CCNx routing system. It contains name prefixes and a list of interfaces. It tells the CCNx forwarder where producers of content with that prefix are located and which interface should be used to forward the *interest* packets to. The packet is forwarded to the interface that has the longest name match on the content name. This

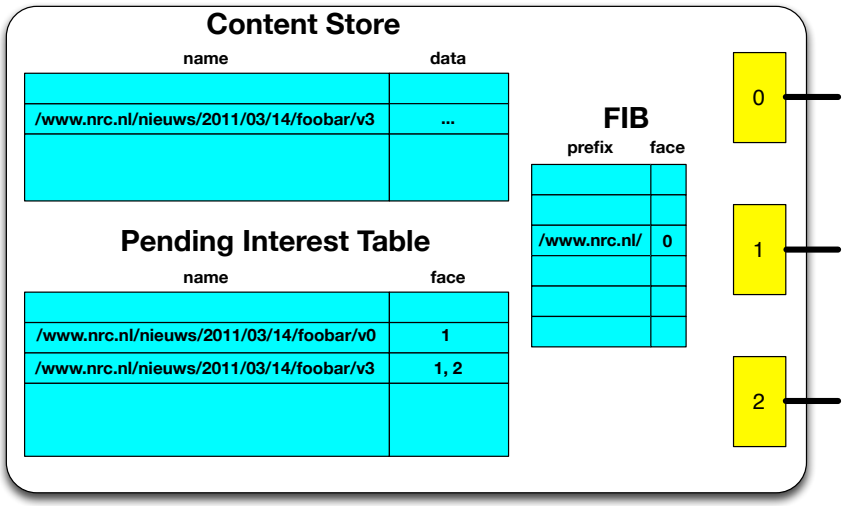
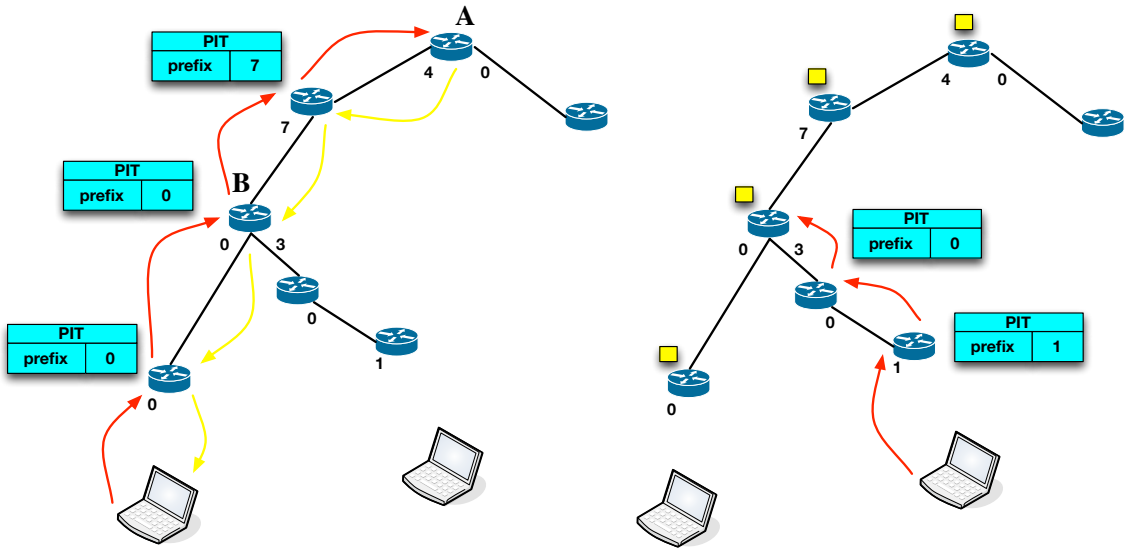


Fig. 2. Components of a CCNx forwarder.

process is intentionally very similar to the way current internet routers work in order to re-use much of the technology and algorithms. Figure 3a shows how an *interest* packet is forwarded to



(a) 1^{st} request

(b) 2^{nd} request

Fig. 3. CCnx routing

a producer. The user on the left sends an *interest* packet to the forwarder it is connected to. The *interest* packet is forwarded towards the producer (red arrows) until it reaches a forwarder that has the content in its Content Store (forwarder A in this case). Each forwarder in between stores the interest in its PIT. The *data* packet is sent on interface 4 by forwarder A, because the *interest* packet was received on that interface. Each forwarder that receives the *data* packet forwards it further according to its PIT. Each forwarder also stores the content in its Content Store. So, when

the *data* packet finally reaches the user, all the forwarders in the path have the content in their Content Store.

When the user on the right requests the same content, the *interest* packet is forwarded until forwarder B. This is shown in figure 3b. Forwarder B has the content in its Content Store, so it can reply with a *data* packet. The *interest* does not need to travel all the way to forwarder A. This example shows that data is sent only once on the path between forwarder A and B.

3 Current Status

NDN and CCNx have attracted several respected network and computer science researchers, including Van Jacobson (PARC), Lixia Zhang (UCLA), KC Claffy (UCSD), Jim Gettys (Alcatel-Lucent Bell Labs), and Sape Mullender (Alcatel-Lucent Bell Labs). But it is also an area were many similar projects exist for many years now and there are still a lot of open research questions. The IEEE Communications Magazine will publish a special issue on ICN in 2012. There have been many meetings and conferences on the subject this year. Some of them are:

- ACM SIGCOMM Workshop on Information-Centric Networking (August 19, 2011, Toronto, Canada [8]
- At IETF 81 in Quebec, Canada, there was an IRTF BOF [9] about ICN. The conclusion of the BOF was that it is too early to start an IRTF working group. The ideas and concepts are not mature yet and it probably needs several implementation cycles to understand the design trade-offs. At IETF 82 in Tapei, Taiwan, there was another ICNRG (which is not an IRTF working group yet) side meeting which was well attended. Another meeting will be held at IETF 83 in Paris in March 2012.

The partners in NDN are PARC, University of California Los Angeles, University of Arizona, University of California Irvine, University of California San Diego, Colorado State University, University of Illinois, University of Memphis, Washington University, Northeastern University, University of Colorado, and the University of Maryland.

Finally, there is also a European FP7 project, called the SAIL project [10]. Partners are Ericsson AB, Alcatel-Lucent, Nokia Siemens Networks, NEC Europe, France Telecom, Telefonica I&D, Telecom Italia, Portugal Telecom Inovacao, Swedish Institute of Computer Science, Technical University of Lisbon, University of Paderborn, Aalto University, KTH, Fraunhofer-SIT, University of Bremen, Hewlett-Packard, Tecnalia, Institut Telecom, Israel Institute of Technology, DOCOMO Communications Laboratories Europe, INRIA, Trinity College Dublin, National ICT Australia, University of Cantabria and Lyatiss.

In the Netherlands, Fernando Kuipers of the TU Delft has a student working on NDN.

4 Conclusion

There is a lot of research going on in the area of NDN (ICN/CCN) and there is running code available. There is also a lot of interest in the internet community. Many find the concepts promising. However, NDN is still in the research phase of de innovation cycle, so a wait and see approach seems to be advisable for SURFnet. In 2012 it is probably sufficient to follow the developments at a high level.

References

1. Named Data Networking Website
   `http://www.named-data.net/`
2. NSF Press Release 10-156
   NSF Announces Future Internet Architecture Awards
   `http://www.nsf.gov/news/news_summ.jsp?cntn_id=117611&org=NSF&from=news`

3. CCNx website
   http://www.ccnx.org/
4. Jacobson, V.
   Google techTalk, 2006
   http://www.youtube.com/watch?feature=player_embedded&v=8Z685OF-PS8
5. Gettys, J.
   Bufferbloat blog
   http://gettys.wordpress.com/what-is-bufferbloat-anyway/
6. Freier, A., Karlton, P., Kocher, P.
   RFC 6101 The Secure Sockets Layer (SSL) Protocol Version 3.0
   August, 2011
   http://tools.ietf.org/html/rfc6101
7. Diers, T., Rescorla, E.
   RFC 5246
   The Transport Layer Security (TLS) Protocol Version 1.2
   August, 2008
   http://tools.ietf.org/html/rfc5246
8. ACM SIGCOMM Workshop on Information-Centric Networking (ICN-2011)
   Toronto, Canada, August 2011
   http://www.neclab.eu/icn-2011/index.html
9. Proposed Information-Centric Networking Research Group (ICNRG)
   http://trac.tools.ietf.org/group/irtf/trac/wiki/icnrg
10. Scalable and Adaptive Internet Solutions (SAIL)
    European Commissions 7th Framework Program
    http://www.sail-project.eu/