Understanding Content Dissemination within Information Centric Networks

Intermediate Status Report

António Rodrigues (up200400437@fe.up.pt)

January 12, 2016

1 Project Status

1.1 Preliminary Work

The initial phase of the project consisted in handling technical aspects and setting up the components for the basic testbed proposed earlier in [1]. The following subsections briefly describe the work conducted during this phase.

1.1.1 Hardware

Table 1 summarizes the hardware selected for running the previously suggested (see [1]) test setups. Types and quantities were slightly altered according to the testbed arrangements on the proposal.

Test Element	**Type**	**Model**	**Qty.**
Content Source(s)	Desktop PC (w/Ubuntu 12.04)	-	1
Inner Node(s)	Wi-Fi Routers (openWRT [2])	Linksys WRT160NL [3]	3
End Node(s)	Laptop (w/Ubuntu 12.04)	Dell Latitude E6430 Toshiba M70 Macbook PRO	3

Table 1: Table of hardware for implementing the proposed test setup (refer to [1] and Sections 1.2, 1.3, 2 for details on the test setup).

1.1.2 CCNx Setup

Setting up CCNx at End Nodes (ENs): This task consisted in compiling the latest release of CCNx (0.8.1) and installing it at the EN level. Initial tests with simple CCNx applications were conducted as an introduction to the platform. As the objective of the work is to monitor the performance of CCNx according to network parameters such as throughput, latency, traffic load, etc. a dedicated version of Wireshark was re-compiled to include a CCN packet dissector [4].

Setting up CCNx at Inner Nodes (INs): Testing CCNx on 'real-world' constrained devices was a clear intention of this work, which differs from other studies (of different nature), performed in the near-past by Vahlenkamp et al. [5, 6]. As mentioned in Section 1.1.1, the chosen platform for the INs was the Linksys WRT160NL, running a Linux-based operating system openWRT (version 12.09). Although no CCNx packages were directly available to openWRT, a custom openWRT build with a CCNx package [7] (version 0.7.2) designed for a different distribution — CeroWRT [8] — was successfully accomplished.

1.2 Test 1: CCNx Throughput Analysis

Test 1 is now completed and did not significantly deviate from the description given in [1]. The main objective was to evaluate CCNx throughput and goodput under different conditions, on a 100 Mbps link, while transferring files of different sizes.

CCNx was tested using a file sending application — `ccnsendchunks` — and a receiving application — `ccncatchunks2` — which implement CCNx own flow control mechanism. For this purpose, CCNx was configured to run over UDP. The sending and receiving ends consisted in two laptops, labeled PC1 and PC2, connected with an Ethernet cable, with the following (different) characteristics (to be precised in the final document). Due to these differences, two rounds of tests have been conducted, one with PC1 as transmitter and PC2 as receiver, and vice-versa. The content used for exchange in the tests consisted in randomly generated data files of 500 kB, 5 MB and 50 MB. The integrity of the transmitted files has been verified via an MD5 checksum, for all transfers. A special CCNx parameter was varied during the tests — the chunk size C — which determines the amount of payload in a CCN Data packet. Values of 1024 B, 4096 B and 8192 B were tested.

The results of this test have been already processed and are displayed in the form shown in Figure 3 (with $F = 5$ and 50 MB and $C = 8192$ B. These are compared with a baseline consisting in a TCP exchange of the same files between PC1 and PC2.

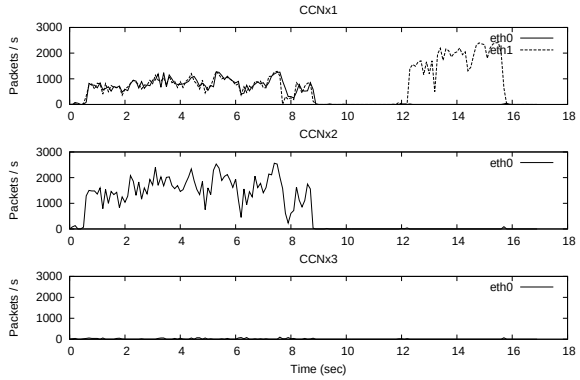

Figure 1: Test 1 results for $F = 5$ MB, $C = 8192$ B.

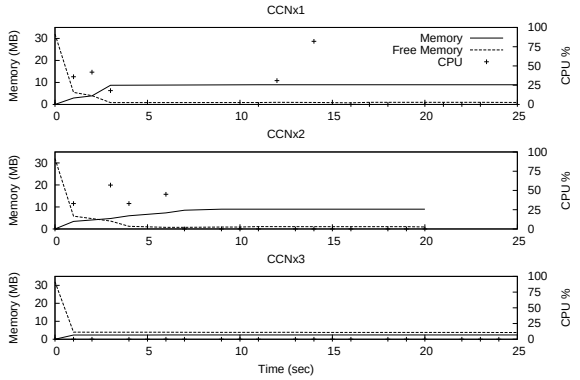

Figure 2: Test 1 results for $F = 50$ MB, $C = 8192$ B.

Figure 3: Test 1 results for $F = 5$ and 50 MB, $C = 8192$ B, between PC1 (receiver) and PC2 (sender). The top charts provide the throughput measurement during the transfer (in Mbps), and other CCNx specific parameters, as provided by the output of the the `ccncatchunks2` application.

Data for the top chart was generated by analyzing the CCNx 'flow', i.e. the exchange of UDP packets (with CCN's source/destination port number 9695) between the IP addresses of PC1 and PC2 using Wireshark captures and its IO Graph analysis. The data for the middle and bottom charts was obtained via the output of the `ccncatchunks2` command, which provides values for parameters specific to the flow control mechanism used by CCNx:

- Number of Interest packets.
- Number of Content (Data) packets.
- Holes: Event which happens when a particular Data packet, for which an Interest has been sent, has not arrived within a timeout.
- CCNx Current Window (Curwin) size: Equivalent to TCP's window size.

The main conclusions for this test — to be detailed on the final document — are (1) the basic structure of Interest and Data packets, (2) the understanding of CCNx's flow control mechanism (which, although implementation-specific, seemed as a valid 'take-home' lesson), (3) an experimental verification of the 'one Interest per Data packet' concept of CCNx and (4) a comparison to a similar task performed with TCP.

1.3 Test 2: CCNx Latency and Load Analysis

Test 2 is currently under evaluation (results not ready for display yet) over a topology similar to that shown in [1], but with readjustments to the number of INs and ENs (see Table 1). Nevertheless, the same objective remains: to verify how CCNx can reduce the latency when an end node N_i accesses (at time $t + \Delta t$) content previously fetched from a content source CS by a node N_j (at time t), connected to the same 'edge' CCN router. Besides analyzing video transmission with CCN's VLC plugin, the transfer of files such as in Test 1 will also be applied. In addition, the use of constrained devices as INs will allow an assessment of the feasibility of the CCN concept in LANs or other networks composed constrained devices.

2 Objective Adjustment

Following the descriptions in Section 1, the objective re-definition shown in Table 2 is proposed.

Test Number	**Changes**
1.1	None.
2.1	Include data files of 500 kB, 5 MB and 50 MB as in test 1.1, besides VLC tests. Change number of ENs to 2, number of INs to 3.
2.2	To be withdrawn.
2.3	Change number of ENs to 2, number of INs to 3. To be conducted conditioned to available time.

Table 2: Objective re-definition for remainder of project.

References

[1] António Damião Rodrigues. Understanding Content Dissemination within Information Centric Networks. 2013.

[2] openwrt.org. OpenWRT - Wireless Freedom. Available as https://openwrt.org/, December 2013.

[3] openwrt.org. Linksys (by Cisco) WRT160NL. Available as http://wiki.openwrt.org/toh/linksys/wrt160nl, December 2013.

[4] Project CCNx. Project CCNx : Wireshark Plugin. Available as https://github.com/ProjectCCNx/ccnx/tree/master/apps/wireshark, December 2013.

[5] Matthias Wählisch, Thomas C Schmidt, and Markus Vahlenkamp. Bulk of Interest : Performance Measurement of Content-Centric Routing. *SIGCOMM Comput. Commun. Rev.*, 42(4):99–100, 2012.

[6] Markus Vahlenkamp. CCNx Meassurement Testbed Implementation. Technical report, Hamburg University of Applied Sciences, 2012.

[7] Dave Täht. Package Repository for the CeroWrt Project. Available as https://github.com/dtaht/ceropackages-3.3/tree/master/net/ccnx, December 2013.

[8] CeroWRT Project. CeroWRT Project. Available as http://www.bufferbloat.net/projects/cerowrt/wiki, December 2013.