CCNxServ Project –

Extending Services with File type Check

Yu-Wen Chen
Columbia University
New York, NY 10025
USA
yc2450@columbia.edu

Abstract

This project extends the CCNxServ to allow services check the file type is supported or not before processing. It enhances the error-handling capability for CCNxServ and avoids users to spend time on completing the service with the wrong output file.

Introduction

Content centric networks is the architecture of computer networks, which allow users request for specific content more than other specific references and physical locations. While many implementations of CCN protocols, like CCNx [1], focus on centering the network on content and handling requests, many people also believe that service scalability and mobility are also the important part of networking. To address this, the paper "CCNServ: Dynamic Service Scalability in Information-Centric Networks" [2] and the Dynamic Services with Content Centric Networking project [3] enable dynamic services on top of the CCNx.

About the CCNxServ project [4]

The CCNxServ project enables services on top of CCNx protocol. It supports executing services in a chain and provides two service examples, linenumber and nextservice, working on txt files.

Extended Scope

While surveying on this project, I found that services did not check the file type before processing. This situation will let user unnecessary waiting for the services process finish with wrong output file. To avoid this situation, adding the file type check function not only enhances the ability of error-handling for CCNservice but could also avoids users to spend time on completing the service with the wrong output file.

Approach

I use an array to store the supported file type for each service. This allows the service support multi types.

The form of the output file from the CCNxServ project is "CONTENT_NAME%2B%SERVICE NAME", and the service will separate the file type from the CONTENTNAME_NAME, like the red circles as followed.

- test.txt
- test.txt%2Blinenumber
- test.txt%2Bnextservice%2Blinenumber

After extracting the file type, the service will write the error message in the output file if the type is not supported. This approach will not let the system exit so users could continually make other commands conveniently.

The procedure in the service side:

Step1. Store supported file types in an array.

Step2. Get the filePath from the command and split the filePath to extract the file type.

Step3. Determine the file type is supported by the service or not

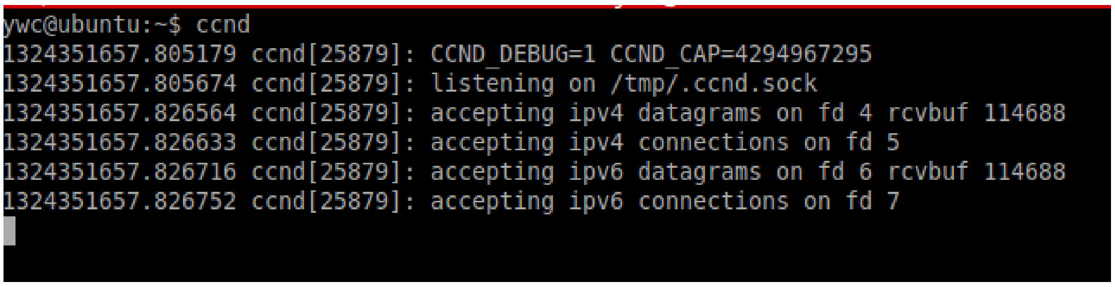
1. Supported:
   Process the service
2. Not Supported:

Write an error messages “Error: File type is not supported” in the output file.

Step4. Return the output file path

Process Screenshots

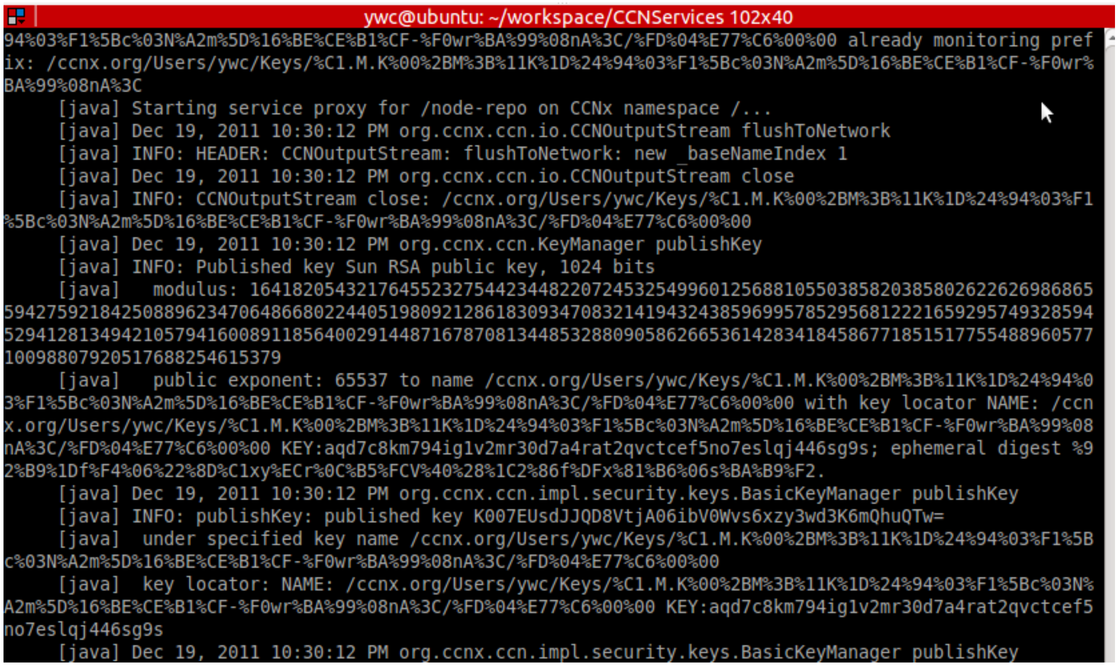
1. Start the ccn deamon with the ccnd command, “$ ccnd”.



```
ywc@ubuntu:~$ ccnd
1324351657.805179 ccnd[25879]: CCND_DEBUG=1 CCND_CAP=4294967295
1324351657.805674 ccnd[25879]: listening on /tmp/.ccnd.sock
1324351657.826564 ccnd[25879]: accepting ipv4 datagrams on fd 4 rcvbuf 114688
1324351657.826633 ccnd[25879]: accepting ipv4 connections on fd 5
1324351657.826716 ccnd[25879]: accepting ipv6 datagrams on fd 6 rcvbuf 114688
1324351657.826752 ccnd[25879]: accepting ipv6 connections on fd 7
```

2. Run the CCNService.

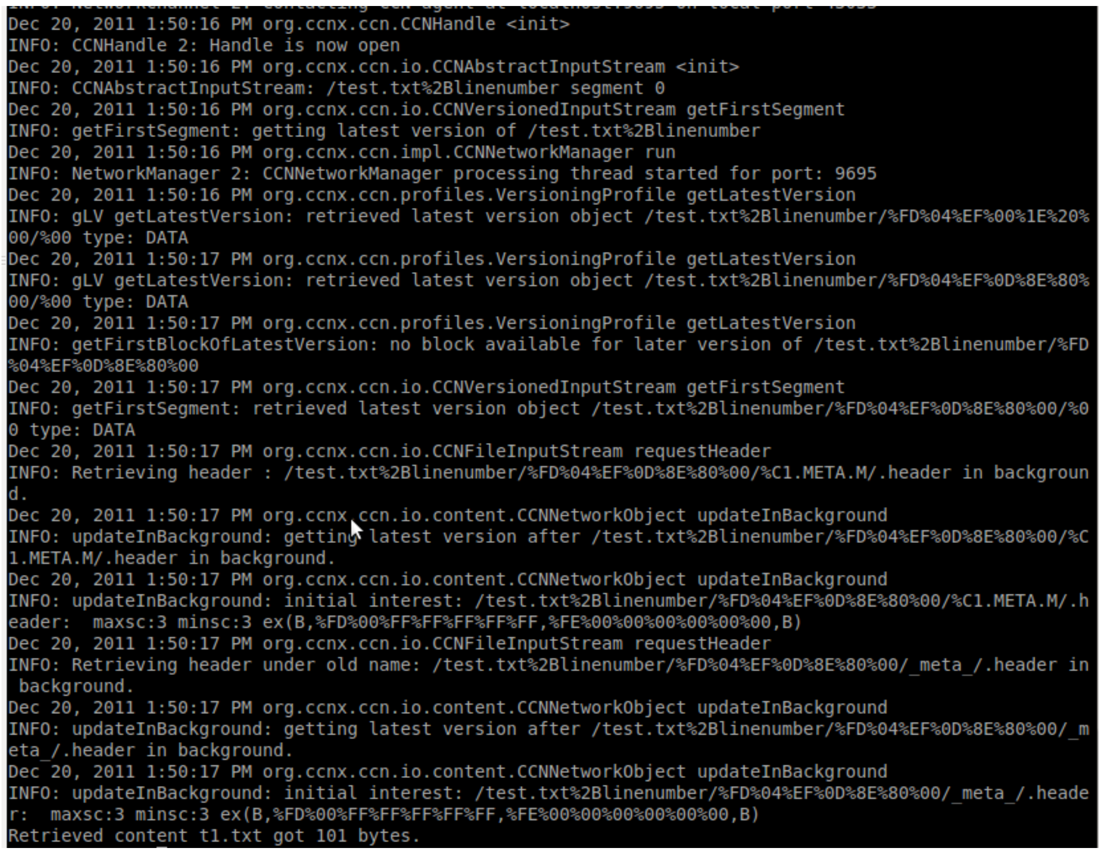
   “CCNServices $ ant run-ccnservices –DCCNX_PREFIX=ccnx:/ -DFILE_PREFIX=/node-repo”



```
ywc@ubuntu: ~/workspace/CCNServices 102x40
94%03%F1%5Bc%03N%A2m%5D%16%BE%CE%B1%CF-%F0wr%BA%99%08nA%3C/%FD%04%E77%C6%00%00 already monitoring pref
ix: /ccnx.org/Users/ywc/Keys/%C1.M.K%00%2BM%3B%11K%1D%24%94%03%F1%5Bc%03N%A2m%5D%16%BE%CE%B1%CF-%F0wr%
BA%99%08nA%3C
     [java] Starting service proxy for /node-repo on CCNx namespace /...
     [java] Dec 19, 2011 10:30:12 PM org.ccnx.ccn.io.CCNOutputStream flushToNetwork
     [java] INFO: HEADER: CCNOutputStream: flushToNetwork: new _baseNameIndex 1
     [java] Dec 19, 2011 10:30:12 PM org.ccnx.ccn.io.CCNOutputStream close
     [java] INFO: CCNOutputStream close: /ccnx.org/Users/ywc/Keys/%C1.M.K%00%2BM%3B%11K%1D%24%94%03%F1
%5Bc%03N%A2m%5D%16%BE%CE%B1%CF-%F0wr%BA%99%08nA%3C/%FD%04%E77%C6%00%00
     [java] Dec 19, 2011 10:30:12 PM org.ccnx.ccn.KeyManager publishKey
     [java] INFO: Published key Sun RSA public key, 1024 bits
     [java]   modulus: 164182054321764552327544234482207245325499601256881055038582038580262262698686
594275921842508896234706486680224405198092128618309347083214194324385969957852956812221659295749328594
529412813494210579416008911856400291448716787081344853288090586266536142834184586771851517755488960577
10098807920517688254615379
     [java]   public exponent: 65537 to name /ccnx.org/Users/ywc/Keys/%C1.M.K%00%2BM%3B%11K%1D%24%94%0
3%F1%5Bc%03N%A2m%5D%16%BE%CE%B1%CF-%F0wr%BA%99%08nA%3C/%FD%04%E77%C6%00%00 with key locator NAME: /ccn
x.org/Users/ywc/Keys/%C1.M.K%00%2BM%3B%11K%1D%24%94%03%F1%5Bc%03N%A2m%5D%16%BE%CE%B1%CF-%F0wr%BA%99%08
nA%3C/%FD%04%E77%C6%00%00 KEY:aqd7c8km794ig1v2mr30d7a4rat2qvctcef5no7eslqj446sg9s; ephemeral digest %9
2%B9%1Df%F4%06%22%8D%C1xy%ECr%0C%B5%FCV%40%28%1C2%86f%DFx%81%B6%06s%BA%B9%F2.
     [java] Dec 19, 2011 10:30:12 PM org.ccnx.ccn.impl.security.keys.BasicKeyManager publishKey
     [java] INFO: publishKey: published key K007EUsdJJQD8VtjA06ibV0Wvs6xzy3wd3K6mQhuQTw=
     [java]  under specified key name /ccnx.org/Users/ywc/Keys/%C1.M.K%00%2BM%3B%11K%1D%24%94%03%F1%5B
c%03N%A2m%5D%16%BE%CE%B1%CF-%F0wr%BA%99%08nA%3C/%FD%04%E77%C6%00%00
     [java]  key locator: NAME: /ccnx.org/Users/ywc/Keys/%C1.M.K%00%2BM%3B%11K%1D%24%94%03%F1%5Bc%03N%
A2m%5D%16%BE%CE%B1%CF-%F0wr%BA%99%08nA%3C/%FD%04%E77%C6%00%00 KEY:aqd7c8km794ig1v2mr30d7a4rat2qvctcef5
no7eslqj446sg9s
     [java] Dec 19, 2011 10:30:12 PM org.ccnx.ccn.impl.security.keys.BasicKeyManager publishKey
```

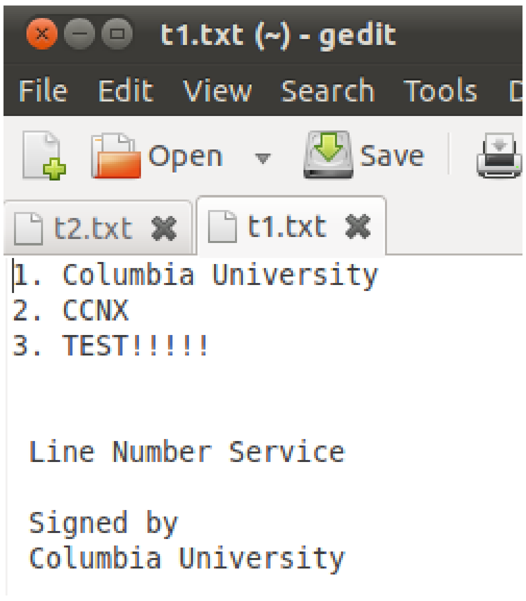
3. Test the service "linenumber" with supported file type

   $ ccngetfile ccnx://test.txt+linenumber t1.txt



```
Dec 20, 2011 1:50:16 PM org.ccnx.ccn.CCNHandle <init>
INFO: CCNHandle 2: Handle is now open
Dec 20, 2011 1:50:16 PM org.ccnx.ccn.io.CCNAbstractInputStream <init>
INFO: CCNAbstractInputStream: /test.txt%2Blinenumber segment 0
Dec 20, 2011 1:50:16 PM org.ccnx.ccn.io.CCNVersionedInputStream getFirstSegment
INFO: getFirstSegment: getting latest version of /test.txt%2Blinenumber
Dec 20, 2011 1:50:16 PM org.ccnx.ccn.impl.CCNNetworkManager run
INFO: NetworkManager 2: CCNNetworkManager processing thread started for port: 9695
Dec 20, 2011 1:50:16 PM org.ccnx.ccn.profiles.VersioningProfile getLatestVersion
INFO: gLV getLatestVersion: retrieved latest version object /test.txt%2Blinenumber/%FD%04%EF%00%1E%20%
00/%00 type: DATA
Dec 20, 2011 1:50:17 PM org.ccnx.ccn.profiles.VersioningProfile getLatestVersion
INFO: gLV getLatestVersion: retrieved latest version object /test.txt%2Blinenumber/%FD%04%EF%0D%8E%80%
00/%00 type: DATA
Dec 20, 2011 1:50:17 PM org.ccnx.ccn.profiles.VersioningProfile getLatestVersion
INFO: getFirstBlockOfLatestVersion: no block available for later version of /test.txt%2Blinenumber/%FD
%04%EF%0D%8E%80%00
Dec 20, 2011 1:50:17 PM org.ccnx.ccn.io.CCNVersionedInputStream getFirstSegment
INFO: getFirstSegment: retrieved latest version object /test.txt%2Blinenumber/%FD%04%EF%0D%8E%80%00/%0
0 type: DATA
Dec 20, 2011 1:50:17 PM org.ccnx.ccn.io.CCNFileInputStream requestHeader
INFO: Retrieving header : /test.txt%2Blinenumber/%FD%04%EF%0D%8E%80%00/%C1.META.M/.header in backgroun
d.
Dec 20, 2011 1:50:17 PM org.ccnx.ccn.io.content.CCNNetworkObject updateInBackground
INFO: updateInBackground: getting latest version after /test.txt%2Blinenumber/%FD%04%EF%0D%8E%80%00/%C
1.META.M/.header in background.
Dec 20, 2011 1:50:17 PM org.ccnx.ccn.io.content.CCNNetworkObject updateInBackground
INFO: updateInBackground: initial interest: /test.txt%2Blinenumber/%FD%04%EF%0D%8E%80%00/%C1.META.M/.h
eader:  maxsc:3 minsc:3 ex(B,%FD%00%FF%FF%FF%FF%FF,%FE%00%00%00%00%00%00,B)
Dec 20, 2011 1:50:17 PM org.ccnx.ccn.io.CCNFileInputStream requestHeader
INFO: Retrieving header under old name: /test.txt%2Blinenumber/%FD%04%EF%0D%8E%80%00/_meta_/.header in
 background.
Dec 20, 2011 1:50:17 PM org.ccnx.ccn.io.content.CCNNetworkObject updateInBackground
INFO: updateInBackground: getting latest version after /test.txt%2Blinenumber/%FD%04%EF%0D%8E%80%00/_m
eta_/.header in background.
Dec 20, 2011 1:50:17 PM org.ccnx.ccn.io.content.CCNNetworkObject updateInBackground
INFO: updateInBackground: initial interest: /test.txt%2Blinenumber/%FD%04%EF%0D%8E%80%00/_meta_/.heade
r:  maxsc:3 minsc:3 ex(B,%FD%00%FF%FF%FF%FF%FF,%FE%00%00%00%00%00%00,B)
Retrieved content t1.txt got 101 bytes.
```

The output file context:



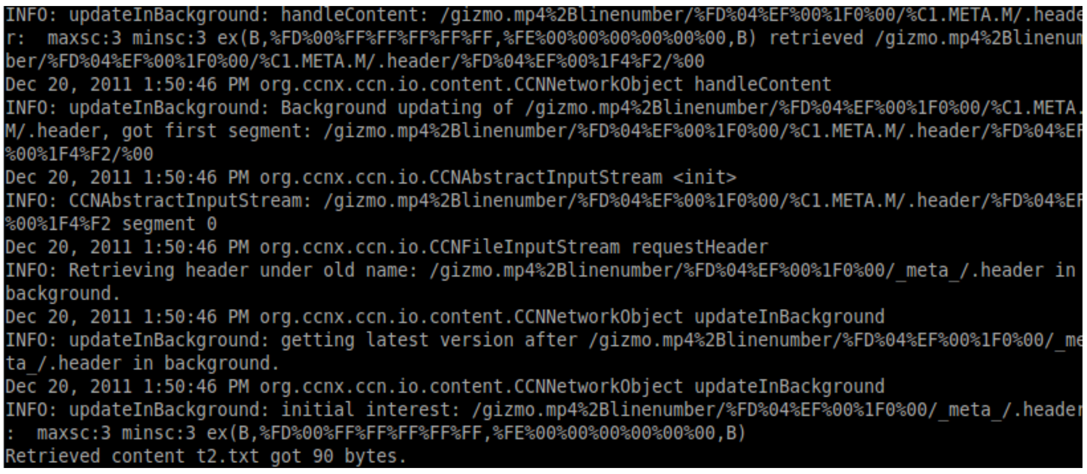
```
1. Columbia University
2. CCNX
3. TEST!!!!!


 Line Number Service

 Signed by
 Columbia University
```

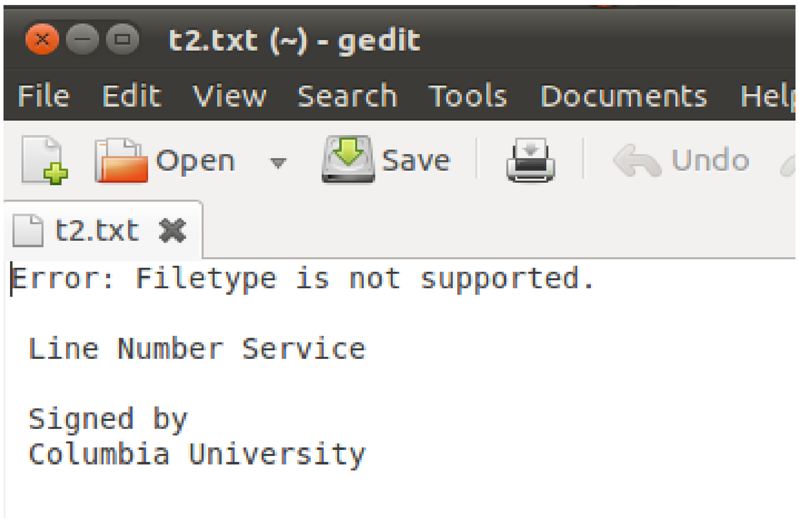
4. Test the service "linenumber" with unsupported file type

   $ ccngetfile ccnx://gizmo.mp4+linenumber t2.txt



```
INFO: updateInBackground: handleContent: /gizmo.mp4%2Blinenumber/%FD%04%EF%00%1F0%00/%C1.META.M/.heade
r:  maxsc:3 minsc:3 ex(B,%FD%00%FF%FF%FF%FF%FF,%FE%00%00%00%00%00%00,B) retrieved /gizmo.mp4%2Blinenum
ber/%FD%04%EF%00%1F0%00/%C1.META.M/.header/%FD%04%EF%00%1F4%F2/%00
Dec 20, 2011 1:50:46 PM org.ccnx.ccn.io.content.CCNNetworkObject handleContent
INFO: updateInBackground: Background updating of /gizmo.mp4%2Blinenumber/%FD%04%EF%00%1F0%00/%C1.META.
M/.header, got first segment: /gizmo.mp4%2Blinenumber/%FD%04%EF%00%1F0%00/%C1.META.M/.header/%FD%04%EF
%00%1F4%F2/%00
Dec 20, 2011 1:50:46 PM org.ccnx.ccn.io.CCNAbstractInputStream <init>
INFO: CCNAbstractInputStream: /gizmo.mp4%2Blinenumber/%FD%04%EF%00%1F0%00/%C1.META.M/.header/%FD%04%EF
%00%1F4%F2 segment 0
Dec 20, 2011 1:50:46 PM org.ccnx.ccn.io.CCNFileInputStream requestHeader
INFO: Retrieving header under old name: /gizmo.mp4%2Blinenumber/%FD%04%EF%00%1F0%00/_meta_/.header in
background.
Dec 20, 2011 1:50:46 PM org.ccnx.ccn.io.content.CCNNetworkObject updateInBackground
INFO: updateInBackground: getting latest version after /gizmo.mp4%2Blinenumber/%FD%04%EF%00%1F0%00/_me
ta_/.header in background.
Dec 20, 2011 1:50:46 PM org.ccnx.ccn.io.content.CCNNetworkObject updateInBackground
INFO: updateInBackground: initial interest: /gizmo.mp4%2Blinenumber/%FD%04%EF%00%1F0%00/_meta_/.header
:  maxsc:3 minsc:3 ex(B,%FD%00%FF%FF%FF%FF%FF,%FE%00%00%00%00%00%00,B)
Retrieved content t2.txt got 90 bytes.
```

   The output file context:



```
Error: Filetype is not supported.

 Line Number Service

 Signed by
 Columbia University
```

Conclusion

The extended scope allows services detect the file type of the object that is passed on to the service before determining whether to process the service or not. This not only enhances the error-handling capability for CCNxServ but also avoids users to spend time on completing the service with the wrong output file.

Future Work for integrating Xuggler[5] applications on the CCService

Currently, the service examples of the CCNxServ project support txt files. It would be a potential work to add services working video files' modification in the future.

After getting familiar with the Content Centric Network and Xuggler, I tried to integrate the Xuggler applications on the CCNService during this semester. However, I found out it is a tough issue to integrate them in separated jar files/modules. Since both Xuggler and new service could work independently in the system, and the codes failed while calling the IMediaTool, which is under the Xuggle-Xuggler.jar file, I thought there are two possible reasons caused the issue. One is due to the OSGi framework could not work with the JNI successfully, and the other one is because the separated jar files could not connect to each other successfully.

For the first possible reason, between the OSGi framework and the JNI, many people had faced the same problem in the past, and all the solutions are related to Bundle Native Code. The Bundle Native Code could notify the OSGi framework of native libraries included in the bundle. Therefore, I added the attribute of Bundle Native Code in the manifest, but it showed the error message with "cannot found those .so files." In order to solve this error, I tried to include those .so files into the project directly and also set the link path to the original location of those .so files. Unfortunately, both of these methods did not solve this error.

For the second possible reason, the connection between the separated jar files, I found it comes out the solution related to the class path. Both methods of adding the attribute of the class-path in the manifest and letting the Xuggler.jar file automatically loaded when the netserv launched still did not make a real progress for integrating Xuggler applications on the CCNService.

In the previous NetServ-ActiveCDN project [6], it solved the similar issue by tightly integrating the NetServ core with Xuggler under same JAR/module. Although I changed my goal to extend the service with the file type check to ensure I could have something to present in the end of this semester under the limited time in this semester, integrating the Xuggler applications on the CCNService in separate jar file/module would still be a potential continued issue in the future.

Acknowledge

Working on this project helps me learn the concept of Content Centric Network, which is really interesting since it is true that people request more for specific content. I also know more about how video files work and how could we do the different modification for video files or streams, and the CCNService project also provide me the knowledge about how the dynamic services work on top of the CCN protocol. Although I had a lot of try and error and was frustrated during the semester, it was interesting to learn all of these new areas and enhanced my ability to think over the possible reasons and solutions with a lot of processes of try and error. Thanks for having the chance to work on this project.

References

1. Project - CCNx^{TM}. http://www.ccnx.org, Sep. 2009
2. Suman Srinivasan, Amandeep Singh, Dhruva Batni, Jae Woo Lee, Henning Schulzrinne and Volker Hilt, “CCNServ: Dynamic Service Scalability in Information-Centric Networks”
3. Project - Dynamic Services with Content Centric Networking
4. Project – CCNService, https://github.com/amanus/CCNServices
5. Xuggler, http://www.xuggle.com/xuggler/
6. Porject - NetServ-ActiveCDN