APPS052-Codensity T408^{TM} and Livepeer Interworking Configuration and Usage Application Note

V4.0

2022-10-10

Table of Contents

1 Table of Abbreviations

Abbreviation:	Full Form:
AI	Artificial Intelligence
API	Application Programming Interface
AUD AV	Access Unit Delimiters Audio Video
AV1	Alliance Open Media Video 1 Codec
CABAC	Context Adaptive Binary Arithmetic Coding
CBR	Constant Bit Rate
CPU	Central Processing Unit
CRF	Constant Rate Factor
DSP	Digital Signal Processing
EOF	End Of File
FIXQB	Fixed Quantization Parameter
FF	Form Factor
FPS	Frames Per Second
GOP	Group of pictures
HDR	High Dynamic Range
HHHL	Half Height, Half Length
HHFL	Half Height, Full Length
HLG	Hybrid Log Gamma
HRD	Hypothetical Reference Decoder
IE	Inference Engine
Mbps	Mega Bit per second
MBps	Mega Byte per second
NLP	Natural Language Processing
NPU	Neural Processing Unit
NVMe	Non-Volatile Memory express
PCIe	Peripheral Component Interconnect express
PPS	Picture Parameter Set
QP	Quoted Printable
RGB	Red Green Blue
RGBA	Red Green Blue Alpha
RDO	Optimization
ROI	Region of interest
SDK	Software Development Kit

SEI	Supplemental Enhancement Information
SRIOV	Single Root I/O Virtualization
SPS	Sequence Parameter Set
VBR	Variable Bit Rat
VCL	Video Coding Layer
VPS	Video Parameter Set
VUI	Video Usability Information

2 References

Technical Notes:
-QuickStartGuideT408_T432_FW2_6.pdf

Specifications:

3 Background

This document describes the configuration and usage for Livepeer application interworking with T408 Video Transcoder.

3.1 Intended Audience

This document is intended to help engineers/technicians/developers operating NETINT video transcoders.

3.2 Compatibility

Software Compatibility

This guide is intended to be used with NETINT Codensity T408 Video Transcoder software Release 2.5.9 or newer.

Hardware Compatibility

Release 2.5.9 or newer supports NETINT Codensity T408 Video Transcoder hardware.

Operation system

All OS supporting T408 Video Transcoder.

4 Setting up Livepeer Environment on Linux Host

4.1 Scope

This section details the minimum environment needed for a successful Livepeer and Livepeer_bench compile.

4.2 Installing Golang

Refer Section 4.5 for link to install go language support.

The steps include downloading a go tar.gz file for Linux from link in Section 4.5 and untar'ing to /usr/local folder. The go tar.gz file version may be newer than the example below.

```
sudo su
cd /usr/local
tar -zxvf go1.17.2.linux-amd64.tar.gz
```

4.3 Some Linux environment

Run 'visudo' to add /usr/local/go/bin path to secure_path.

```
Defaults        mail_badpass
Defaults        secure_path="/usr/local/sbin:/usr/local/bin:/usr/local/go/bin:/usr/sbin:/usr/bin:/sbin:/bin:/snap/bin"

# Host alias specification
```

Edit ~/.profile to add /usr/local/go/bin to $PATH.

```
# set PATH so it includes user's private bin if it exists
if [ -d "$HOME/.local/bin" ] ; then
    PATH="$HOME/.local/bin:/usr/local/go/bin:$PATH"
fi
```

Reboot the host.

Running 'which go' should give output as below. Livepeer code compile is based on 'go'.

```
$ which go
/usr/local/go/bin/go
```

4.4 Installing Livepeer

Refer Section 4.5 for link on steps to install Livepeer from source.

The steps include installing few linux packages and cloning Livepeer repos.

```
sudo apt-get update && sudo apt-get -y install build-essential
pkg-config autoconf git curl
```

```
sudo apt-get -y install clang-8 clang-tools-8
```

User may need to install some gnutls packages to get livepeer compile going.

```
sudo apt install libgnutls28-dev
sudo apt install gnutls-bin
sudo apt install python-gnutls
```

Clone the public git repos lpms and go-livepeer to get the livepeer code.

```
git clone https://github.com/livepeer/lpms.git
git clone https://github.com/livepeer/go-livepeer.git
```

User will see a ~/lpms and ~/go-livepeer folder.

4.5 Useful Links

1] Livepeer 'Build from source' – https://docs.livepeer.org/installation/install-livepeer/installing-for-development (link may change and can be accessed from docs.livepeer.org)
2] go lang - https://go.dev/doc/install

5 T408 FW and SW with Livepeer support

5.1 Scope

This section details the T408 release that has changes specific to Livepeer.

5.2 T408 FW and FFmpeg Software

Check with Netint Support on firmware to load on T408 cards to support Livepeer.

Currently, our libxcoder for livepeer branch is “livepeer_dev_4.3.1”. The corresponding FW is v2.6.1 .

1. Download v2.6.1 FW and install
2. Download libxcoder and FFmpeg-4.2.1 in branch “livepeer_dev_4.3.1”

Follow the steps in QuickStartGuideT408_T432_FW2_6.pdf or newer to install the host environment and the libxcoder and FFmpeg software.

6 Compiling Livepeer application

6.1 Scope

This section will help a user compile the Livepeer and Livepeer_bench applications.

Note – Livepeer uses the libxcoder and FFmpeg compiled files in ~/compiled folder when compiling livepeer_bench and livepeer bench applications. So, steps in section 6.2 are critical to correctly compiling livpeer applications in section 6.3.

6.2 Compile libxcoder and FFmpeg-n4.2.1

Steps to compile libxcoder,

```
cd ~/FFmpegXcoder
cd libxcoder
make clean
rm -rf build
sh build.sh
```

Create the following folder:

```
cd ~
mkdir compiled
```

Populate ~/compiled folder by copying the following libxcoder files.

```
cd ~/FFmpegXcoder/libxcoder
cd source
cp ni_defs.h ~/compiled/include/
cp ni_device_api.h ~/compiled/include/
cp ni_rsrc_api.h ~/compiled/include/
cp ni_util.h ~/compiled/include/
cd ../build
cp /usr/local/lib/libxcoder.a ~/compiled/lib
cp /usr/local/lib/libxcoder.so.261R2012 ~/compiled/lib
```

NOTE – the libxcoder.so.261R2012 library file name may be different in your snapshot
NOTE – libxcoder.a, libxcoder.so.261R2012 are critical files. In case your FFmpeg build fails to compile, please check with Netint Support to confirm these files.

Run the following command,

```
export PKG_CONFIG_PATH=$HOME/compiled/lib/pkgconfig
```

Compile FFmpeg,

```
cd ~/FFmpegXcoder
cd FFmpeg-n4.2.1
make clean
sh build_ffmpeg.sh --dry
```

Shown below is the output of the command:

```
./configure --pkg-config-flags=--static --enable-gpl --enable-
nonfree --extra-ldflags=-lm --extra-ldflags=-ldl --enable-
libxcoder --enable-ni --enable-pthreads --extra-libs=-lpthread --
enable-encoders --enable-decoders --enable-avfilter --enable-
muxers --enable-demuxers --enable-parsers --enable-x86asm --
disable-debug --disable-ffplay --disable-ffprobe --disable-
libx264 --disable-libx265 --disable-cuda-nvcc --disable-cuda --
disable-cuvid --disable-nvdec --disable-nvenc --disable-libvmaf -
-enable-static --disable-shared --extra-cflags=-
UNIENC_MULTI_THREAD
```

Copy the command and append the following flags:

```
--extra-cflags=-I/home/nvme/compiled/include  --extra-ldflags=-
L/home/nvme/compiled/lib --prefix=/home/nvme/compiled
```

The final command should look like this:

```
./configure --pkg-config-flags=--static --enable-gpl --enable-
nonfree --extra-ldflags=-lm --extra-ldflags=-ldl --enable-
libxcoder --enable-ni --enable-pthreads --extra-libs=-lpthread --
enable-encoders --enable-decoders --enable-avfilter --enable-
muxers --enable-demuxers --enable-parsers --enable-x86asm --
disable-debug --disable-ffplay --disable-ffprobe --disable-
libx264 --disable-libx265 --disable-cuda-nvcc --disable-cuda --
disable-cuvid --disable-nvdec --disable-nvenc --disable-libvmaf -
-enable-static --disable-shared --extra-cflags=-
```

```
UNIENC_MULTI_THREAD --extra-cflags=-I/home/nvme/compiled/include
--extra-ldflags=-L/home/nvme/compiled/lib  --
prefix=/home/nvme/compiled
```

After running the configure command the following commands to build and install ffmpeg:

```
make -j
sudo make install
```

(this step will copy the required libraries and binaries to the ~/compiled folder)

The libxcoder and FFmpeg compile and install steps are now complete.

6.3 Compiling Livepeer and Livepeer_bench

Livepeer and Livepeer_bench require the following three folders for a successful compile –

`~/compiled` // this folder has the compiled binaries and libraries for T408 SW
`~/lpms` // this folder has the source from https://github.com/livepeer/lpms
`~/go-livepeer` // this folder has the Livepeer specific source and libraries

Compile livepeer_bench,

```
cd ~/go-livepeer
make livepeer_bench
```

Compile livepeer,

```
cd ~/go-livepeer
make livepeer BUILD_TAGS=mainnet
```

Note: For CentOS 7, the following flag needs to be added to the Makefile in the /go-livepeer folder before compiling the binaries:

```
cgo_cflags := -std=gnu99
```

```
SHELL=/bin/bash
GO_BUILD_DIR?="./"

all: net/lp_rpc.pb.go net/redeemer.pb.go net/redeemer_mock.pb.go core/test_segment.go livepeer livepeer_cli livepeer_router li

net/lp_rpc.pb.go: net/lp_rpc.proto
	protoc -I=. --go_out=plugins=grpc:. $^

net/redeemer.pb.go: net/redeemer.proto
	protoc -I=. --go_out=plugins=grpc:. $^

net/redeemer_mock.pb.go: net/redeemer.pb.go
	mockgen -source net/redeemer.pb.go -destination net/redeemer_mock.pb.go -package net $^

core/test_segment.go:
	core/test_segment.sh core/test_segment.go

version=$(shell cat VERSION)

ldflags := -X github.com/livepeer/go-livepeer/core.LivepeerVersion=$(shell ./print_version.sh)
cgo_cflags := -std=gnu99
cgo_ldflags :=

uname_s := $(shell uname -s)
ifeq ($(uname_s),Darwin)
		cgo_ldflags += -framework CoreFoundation -framework Security
```

7 Running Livepeer command

7.1 Scope

This section details the usage example for livepeer_bench application. The livepeer application usage is referred to the Livepeer docs website.

7.2 Launch Livepeer_bench

Command to launch livepeer_bench –

```
sudo ./livepeer_bench -in bbb/source.m3u8 -transcodingOptions
cmd/livepeer_bench/transcodingOptions.json -concurrentSessions 1
-netint 0 -log 2
```

where,

-log is the logging level values 1 for less to 6 for more logging

-netint tells the application to use Netint video transcoder. The value '0' is the device instance number.
NOTE - Currently, the T408 SW will auto detect the number of T408 devices on host and distribute the decoder and encoder load across all devices to get best hardware utilization. So, a value of -netint 0 will still distribute load across all T408 instances.

-concurrentSessions 1 or higher. Livepeer concurrentSession is 1 decode + 4 ladder encodes

-transcodingOptions it takes an input json file that defines the encoding parameters for the 4 ladder encodes

-in is the input clip, usually, a 30 segment big buck bunny clip set from
https://storage.googleapis.com/lp_testharness_assets/bbb_1080p_30fps_1min_2sec_hls.tar.gz

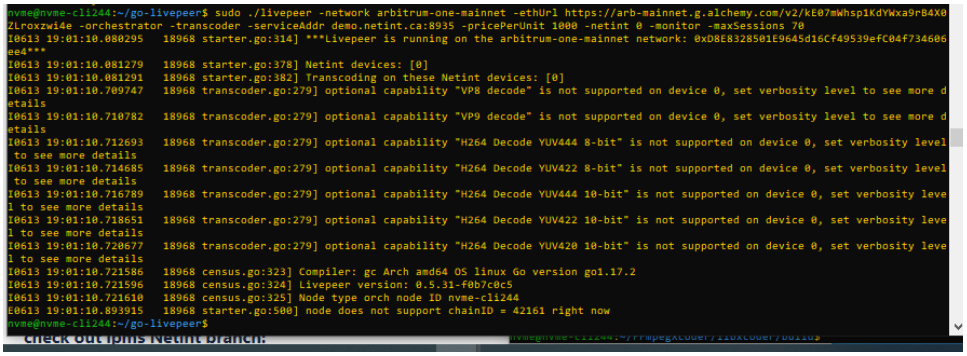
7.3 Launch Livepeer

The Livepeer end-to-end usage documentation can be had from https://docs.livepeer.org/ .

Open a terminal to run a Livepeer node and then on the other terminal to monitor it using tmux.

$ sudo ./livepeer -network arbitrum-one-mainnet -ethUrl [Your Ethereum URL] -orchestrator -transcoder -serviceAddr [Your server Address:Port] -pricePerUnit 1000 -netint 0 -monitor -maxSessions 70

If it asks for a password, enter the password you have setup for your account.



```
nvme@nvme-cli244:~/go-livepeer$ sudo ./livepeer -network arbitrum-one-mainnet -ethUrl https://arb-mainnet.g.alchemy.com/v2/kE07mWhsp1KdYWxa9rB4X0
ZLroxzwi4e -orchestrator -transcoder -serviceAddr demo.netint.ca:8935 -pricePerUnit 1000 -netint 0 -monitor -maxSessions 70
I0613 19:01:10.080295   18968 starter.go:314] ***Livepeer is running on the arbitrum-one-mainnet network: 0xD8E8328501E9645d16Cf49539efC04f734606
ee4***
I0613 19:01:10.081279   18968 starter.go:378] Netint devices: [0]
I0613 19:01:10.081291   18968 starter.go:382] Transcoding on these Netint devices: [0]
I0613 19:01:10.709747   18968 transcoder.go:279] optional capability "VP8 decode" is not supported on device 0, set verbosity level to see more d
etails
I0613 19:01:10.710782   18968 transcoder.go:279] optional capability "VP9 decode" is not supported on device 0, set verbosity level to see more d
etails
I0613 19:01:10.712693   18968 transcoder.go:279] optional capability "H264 Decode YUV444 8-bit" is not supported on device 0, set verbosity level
 to see more details
I0613 19:01:10.714685   18968 transcoder.go:279] optional capability "H264 Decode YUV422 8-bit" is not supported on device 0, set verbosity level
 to see more details
I0613 19:01:10.716789   18968 transcoder.go:279] optional capability "H264 Decode YUV444 10-bit" is not supported on device 0, set verbosity leve
l to see more details
I0613 19:01:10.718651   18968 transcoder.go:279] optional capability "H264 Decode YUV422 10-bit" is not supported on device 0, set verbosity leve
l to see more details
I0613 19:01:10.720677   18968 transcoder.go:279] optional capability "H264 Decode YUV420 10-bit" is not supported on device 0, set verbosity leve
l to see more details
I0613 19:01:10.721586   18968 census.go:323] Compiler: gc Arch amd64 OS linux Go version go1.17.2
I0613 19:01:10.721596   18968 census.go:324] Livepeer version: 0.5.31-f0b7c0c5
I0613 19:01:10.721610   18968 census.go:325] Node type orch node ID nvme-cli244
E0613 19:01:10.893915   18968 starter.go:500] node does not support chainID = 42161 right now
nvme@nvme-cli244:~/go-livepeer$
```

Do not touch this terminal to disturb this Livepeer node, switch to the other tmux terminal to check and set this Livepeer node. If everything is setup correctly, the livepeer_cli will look like:

```
nvme@nvme-cli244:~/go-livepeer$ ./livepeer_cli
+-----------------------------------------------------------+
| Welcome to livepeer-cli, your Livepeer command line tool    |
|                                                             |
| This tool lets you interact with a local Livepeer node       |
| and participate in the Livepeer protocol without the         |
| hassle that it would normally entail.                        |
|                                                             |
+-----------------------------------------------------------+


+-----------+
|NODE STATS|
+-----------+
*-----------------------------------*------------------------------------------*
|                         Node's version |                         0.5.31-f0b7c0c5 |
*-----------------------------------*------------------------------------------*
|              Node's GO runtime version |                                go1.17.2 |
*-----------------------------------*------------------------------------------*
|                    Node's architecture |                                   amd64
|
*-----------------------------------*------------------------------------------*
|                Node's operating system |                                   linux |
*-----------------------------------*------------------------------------------*
|                              HTTP Port |
7935 |
*-----------------------------------*------------------------------------------*
```

```
|                    Controller Address | 0xD8E8328501E9645d16Cf49539efC04f734606ee4 |
| FOR REFERENCE - DO NOT SEND TOKENS |
|
*----------------------------------*----------------------------------------*
|                  LivepeerToken Address | 0x289ba1701C2F088cf0faf8B3705246331cB8A839 |
| FOR REFERENCE - DO NOT SEND TOKENS |
|
*----------------------------------*----------------------------------------*
|           LivepeerTokenFaucet Address | 0x0000000000000000000000000000000000000000
|
| FOR REFERENCE - DO NOT SEND TOKENS |
|
*----------------------------------*----------------------------------------*
|                    Orchestrator Account | 0xE7DfD92ECF05f841166673B64efdF261Af66A0Be |
|              YOUR WALLET FOR ETH & LPT |
|
*----------------------------------*----------------------------------------*
|                              LPT Balance |                                   0
LPTU |
*----------------------------------*----------------------------------------*
|                              ETH Balance |                  1922035257850508
WEI |
*----------------------------------*----------------------------------------*
|                            Max Gas Price |
n/a |
*----------------------------------*----------------------------------------*
|                            Min Gas Price |                                   0 GWei
|
*----------------------------------*----------------------------------------*
+------------------+
|ORCHESTRATOR STATS|
+------------------+
*--------------------*--------------------------*
|                Status |                Registered |
*--------------------*--------------------------*
|                Active |                      true |
*--------------------*--------------------------*
|           Service URI | https://208.98.203.76:8935 |
*--------------------*--------------------------*
|      Delegated Stake | 105.410544362901602456 LPT |
*--------------------*--------------------------*
|        Reward Cut (%) |                        10 |
*--------------------*--------------------------*
```

```
|              Fee Cut (%) |                          30 |
*----------------------*----------------------------*
|      Last Reward Round |                        2591 |
*----------------------*----------------------------*
| Base price per pixel |          1000 wei / 1 pixels |
*----------------------*----------------------------*
+---------------+
|DELEGATOR STATS|
+---------------+
*------------------*---------------------------------------------*
|              Status |                                      Bonded |
*------------------*---------------------------------------------*
|               Stake |                                       5 LPT |
*------------------*---------------------------------------------*
|      Collected Fees |                                       0 WEI |
*------------------*---------------------------------------------*
|       Pending Stake |                 5.130920073750001525 LPT |
*------------------*---------------------------------------------*
|        Pending Fees |                 0.046627624121351565 ETH |
*------------------*---------------------------------------------*
|     Delegated Stake |             105.410544362901602456 LPT |
*------------------*---------------------------------------------*
| Delegate Address | 0xE7DfD92ECF05f841166673B64efdF261Af66A0Be |
*------------------*---------------------------------------------*
| Last Claim Round |                                         2497 |
*------------------*---------------------------------------------*
|         Start Round |                                        2498 |
*------------------*---------------------------------------------*
CURRENT ROUND: 2598

What would you like to do? (default = stats)
1. Get node status
2. View protocol parameters
3. List registered orchestrators
4. Invoke "initialize round"
5. Invoke "bond"
6. Invoke "unbond"
7. Invoke "rebond"
8. Invoke "withdraw stake" (LPT)
9. Invoke "withdraw fees" (ETH)
10. Invoke "transfer" (LPT)
11. Invoke "reward"
12. Invoke multi-step "become an orchestrator"
```

13. Set orchestrator config
14. Set maximum Ethereum gas price
15. Set minimum Ethereum gas price
16. Sign a message
17. Sign typed data
18. Vote in a poll

8 Revision History

Version	Change	Create Date	Author
0.1	Document Created	January 13, 2022	Balwinder Sidhu
0.2	Updated based on trail by SL	January 14, 2022	Balwinder Sidhu
1.0	Issued version	January 14, 2022	Balwinder Sidhu
2.0	Update Section 4.2. Issued version.	January 31, 2022	Balwinder Sidhu
3.0	Extend Section 5.2. Issued version.	February 2, 2022	Balwinder Sidhu
3.1	Extended Section 6.2	June 16, 2022	Arav Bhatt
3.2	Extended Section 7.3	July 5, 2022	York He
4.0	Issued version 4.0	August 24, 2022	Arav Bhatt

9 Legal Notice

Information in this document is provided in connection with NETINT products. No license, express or implied, by estoppel or otherwise, to any intellectual property rights is granted by this document. Except as provided in NETINT's terms and conditions of sale for such products, NETINT assumes no liability whatsoever and NETINT disclaims any express or implied warranty, relating to sale and/or use of NETINT products including liability or warranties relating to fitness for a particular purpose, merchantability, or infringement of any patent, copyright or other intellectual property right.

A "Mission Critical Application" is any application in which failure of the NETINT Product could result, directly or indirectly, in personal injury or death. Should you purchase or use NETINT's products for any such mission critical application, you shall indemnify and hold NETINT and its subsidiaries, subcontractors and affiliates, and the directors, officers, and employees of each, harmless against all claims costs, damages, and expenses and reasonable attorney's fees arising out of, directly or indirectly, any claim of product liability, personal injury, or death arising in any way out of such mission critical application, whether or not NETINT or its subcontractor was negligent in the design, manufacture, or warning of the NETINT product or any of its parts.

NETINT may make changes to specifications, technical documentation, and product descriptions at any time, without notice. The information here is subject to change without notice. Do not finalize a design with this information. The products described in this document may contain design defects or errors known as errata which may cause the product to deviate from published specifications.

NETINT, Codensity, and NETINT Logo are trademarks of NETINT Technologies Inc. All other trademarks or registered trademarks are the property of their respective owners.