



NP1

## Network Programmability

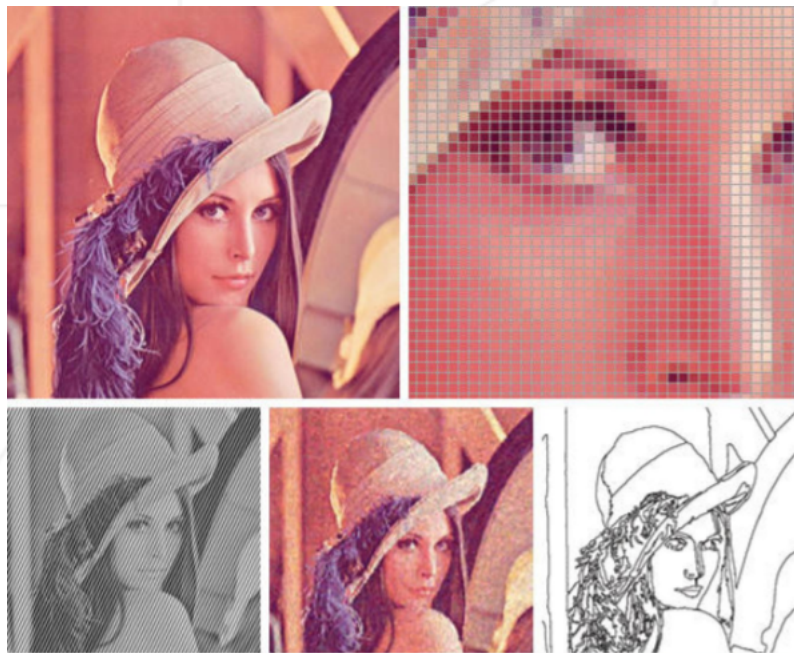
Cisco [balexey@cisco.com](mailto:balexey@cisco.com)  
UNIT Team [born2code@unit.ua](mailto:born2code@unit.ua)

*Summary: This project aims to give the opportunity to create a software product with real hardware. This subject was made in collaboration with [Cisco](#).*



# Chapter I

## Forewords



Lena

Meet Lena Soderberg! Lena is a Swedish model. She was born on March 31, 1951.

There are some details about Lena:

**Ambitions:** I'd like to become an actress and top model.

**Turn-ons:** My husband.

**Turnoffs:** Men who wear shorts with white socks and black shoes.

**In my spare time:** I read, listen to music, play with the dog, and love my husband.

**Don't tell anyone:** My in-laws remind me sometimes of Archie and Edith Bunker.

**Favorite TV:** Night Gallery, All in the Family.

**Favorite Performers:** The Bee Gees, Ingrid Bergman

**My ideal man:** A 24-year-old advertising salesman who's nutty and chubby.

**I see myself:** : In a big house in Europe, at a big yard and with a lot of children.

At the age of 21, Lena became a well-known fashion model. The men's magazine named Lena as Miss November 1972. You can find many photos of Lena on the Internet. But the most famous one, you can see below the text. This photo is very often used to demonstrate various digital image compression algorithms

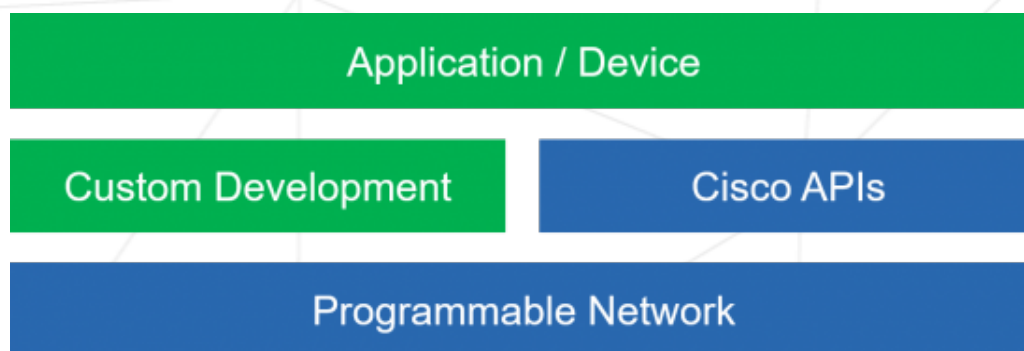
Engineers, researchers, and students who are familiar with image processing or compression most likely use Lena's picture in their experiments or project assignments because her picture is one of the most widely used standard test images. Today, Lena's image is recognized as one of the most important events in the history of digital imaging. Now you know more about Lena, the First Lady of the Internet.

See also: Lorem ipsum, China Girl, Cornell box, Calgary corpus.

# Chapter II

## Introduction

The project is devoted to the solution of specific problems that arise while operating the network of any size. You will work within new direction, Network Programmability. The modern network is developing very quickly and there is a need to use new tools to manage , configured and control the network. One of such approaches is the Software-Defined Networking (SDN).



This led to a new software method of the management automation and network control, which in turn created demand for a new profession, DevOps (Development and IT Operations)

Today's fast-moving world makes DevOps essential for any flexible and lean businesses, in order to quickly respond to changing customer and market demands. DevOps helps companies to achieve continuous delivery of software-driven innovations. DevOps is all about effective communication and collaboration. Now we see the emergence of a new discipline, NetDevOps, and teams are using it to apply DevOps principles and techniques to utilizing a software programmable network to manage, deploy, and run applications.

Principles of Network as Code are as follow:

- Store Network Configurations in Source Control
- Source Control is the Single Source of Truth
- Deploy Configurations with Programmatic APIs

In this work you will use both the Southbound APIs (CLI, SNMP ...) and the Northbound APIs (Web UI, RESTful). This cooperation will allow you to learn how to interact with the network from different parties.

# Chapter III

## Goals

This project aims to make you familiar with:

- Software defined network
- Network programmability
- NetDevOps
- Southbound and Northbound APIs
- Tools or systems for automatic monitoring and creation of tasks

# Chapter IV

## General instructions

- This project will be evaluated only by humans
- This project must use the Digital Network Architecture (DNA) Center
- You must write your own scripts, or a sequence of CLI/SNMP commands to test your work efficiency.



You can also register on Cisco DevNet site [developer.cisco.com](https://developer.cisco.com) and try to find some useful information

# Chapter V

## The project

### V.1 Mandatory part

Network Health Check Daily. You need to automate daily health reports related to the network support. Generate and deliver automated network reporting. Provide different reports for different level users.



Normal network status is, when you add a device to inventory and get access to network topology for the first time.

Here's what you must do:

- Turn-key vertical-focused application, program or web service
- User-friendly interface
- CIO reports: gets basic Red, Yellow, Green health report and high level [SLA](#) stats
- IT director: gets opened/closed service requests (ticket) and [MTTR](#) stats. Assign responsible persons from the DevOps list for solving the ticket
- DevOps: can solve ticket, get WAN availability and average latency reports
- Ticket management: filters by type of problem (for example: traffic, configuration change), creation date, ticket status (for example: open, in progress, rejected, solved)
- Develop notifications module for sending alerts via Webex messaging (or other corporate messenger) and Email
- Collect and display inventory and topology from DNA-C
- Develop report letter text for different level users (CIO, IT director, DevOps)

- Integrate your work with [Webex](#) (or other corporate messenger). Create special team/space where you can automatic add all people by role, and publish all useful information/alerts there
- Send notification and report message via Email and Spark (or other corporate messenger) and use additional way if you can
- Detect and report about all changes in the configuration of drives, path trace, add/edit device interface etc.
- Write the CLI commands that will be able to display: Configuration change; Routing Protocol (OSPF); Switching Protocols (STP). Configure path trace, add/edit device interface etc. Save all your CLI commands in the file, you must run them during the defense

If everything works perfectly, certain things can be hidden. So, find a way to prove that everything works as expected. You should make everything visible, therefore, it can be shown in order to help streamlining the correction.

You can use programming languages, frameworks and libraries that you need.

- DNA-C Labs 1 & 2: These labs contain a Virtual DNA-C system, a Linux DevBox, and a sample topology of real hardware.
- DNA-C Labs 3 & 4: These labs contain a Hardware DNA-C Appliance, a Linux DevBox, and a sample topology of real hardware.



For SLA you can collect and count the following parameters: Service/device availability, mean time to recover from incidents, Customer Satisfaction (when there are some problems with the network (problems with traffic, availability of devices and etc.) customers begin to get nervous)).



If Network Sandbox Lab status is Active with Errors, don't be disheartened and try to book one more (maybe from another account)



Standard booking duration is 12 hours. If you need more, you can extend sandbox end time. DNA-C GUI must discover three and more devices to check all functionality of your work and action during defense



## V.2 Bonus

As long as the mandatory part rules and the general instructions are respected, you can always add all the bonuses you wish, they will be graded directly by your corrector. For example:

- Support APIC-EM
- Policy Administration
- Use DB for the project, and storage history
- View and Manage Users
- Compare different network state (hour to hour, day by day)
- Automatic problem-solving control
- ...

Bonuses will be taken into account only if the mandatory part is flawless

# Chapter VI

## Turn-in and peer-evaluation

Turn-in and peer-evaluation. Turn your work and author file in using your GiT repository, as usual. Only work present on your repository will be graded in defense.