CHAPPELL UNIVERSITY

Internet Society Interplanetary Chapter

IPNSIG

LAB 1 WASTC SPECIAL EVENT

TCP/IP IN SPACE

Configure and test ION-DTN on a Raspberry Pi

LAURA CHAPPELL
laura@chappellu.com

SCOTT SPICER
scott@chappellu.com

Contents

Presenters: Laura Chappell and Scott Spicer

Laura Chappell and Scott Spicer are leading voices at the intersection of advanced networking, education, and the future of space communications. As collaborators with the **InterPlanetary Networking Special Interest Group (IPNSIG)**, they champion the global expansion of Delay/Disruption-Tolerant Networking (DTN) and its transformative potential for both terrestrial and interplanetary connectivity.

Laura Chappell, founder of **Chappell University**, is recognized worldwide as a premier network analyst and educator. Known for her engaging and insightful teaching style, she has spent decades training IT professionals, engineers, and educators to understand, optimize, and secure networks across every scale. Laura's recent work with IPNSIG extends her lifelong mission of "connecting people through understanding networks" into the next frontier: space. She focuses on translating DTN's technical depth into practical, human-centered knowledge that inspires collaboration across scientific, academic, and engineering communities.

Scott Spicer is an emerging technology strategist and space networking advocate whose work bridges education on communication systems, distributed networks, and real-world applications of DTN. He contributes to the development and educational implementation of DTN lab environments, focusing on accessible demonstrations of interplanetary networking concepts. His work emphasizes the practical setup and operation of ION-DTN nodes, contact graph configuration, and the translation of complex networking principles into hands-on learning experiences.

Together, Scott and Laura combine technical insight with an educator's clarity and a visionary perspective on the future of communication-one where humanity stays connected, even across the vast distances of the solar system.

What is ION-DTN?

NASA's Interplanetary Overlay Network (ION) is an implementation of the Delay Tolerant Network (DTN) architecture usable in embedded environments (including spacecraft flight computers) and ground systems.

Lab 1 Overview

This lab begins with an introduction to the Interplanetary Overlay Network (ION) and how it works at a high level.

We will then image a microSD card for use in a Raspberry Pi-5, build and install ION DTN software, and run connectivity tests on a single Raspberry Pi-5 with two instances of ION-DTN running (communicating over *localhost*).

Lab 2 Overview

In Lab 2, we will show you how to set up a second Raspberry Pi 5 with ION to communicate between ION nodes over a network. We will examine the issues that may occur when communicating between ION nodes on separate networks, as well.

Prerequisites for Lab 1

One Raspberry Pi-5 with at least 4 GB RAM, keyboard, mouse, monitor, power supply, internet access via wireless or LAN, and a minimum 64 GB microSD card.[1]

This ION-DTN lab will be done via command line in terminal windows, but Raspberry Pi operating system installation and initial configuration will be demonstrated on a desktop GUI using the **Raspberry Pi OS Imager** tool.

Raspberry Pi-5s sold alone are just the board, with no power supply, case, or microSD card. Kits from various resellers usually come with microSD cards, and some even have the Raspberry Pi OS already installed.

[1] This lab can be configured with a "headless" Raspberry Pi (no monitor, keyboard, or mouse directly attached) if you desire. Configuring the lab using a headless Raspberry Pi is beyond the scope of this lab.

Lab 1 List of Steps

We will go through the following nine steps in this lab.

Step 1: Image the Pi OS on the MicroSD Card

Step 2: Update and Upgrade Packages

Step 3: Download ION-DTN

Step 4: Set an Environment Variable

Step 5: Install Build Dependencies

Step 6: Build ION-DTN

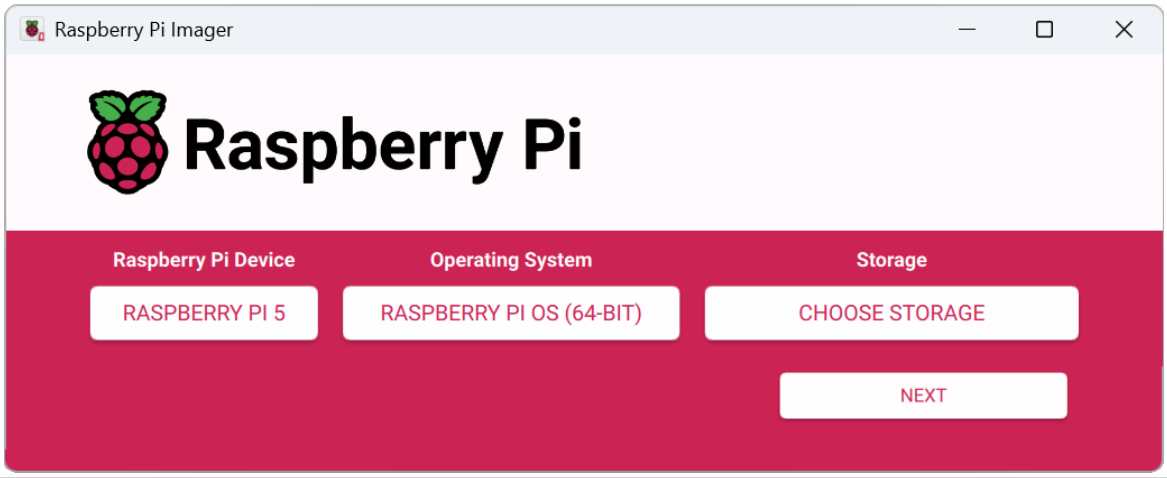
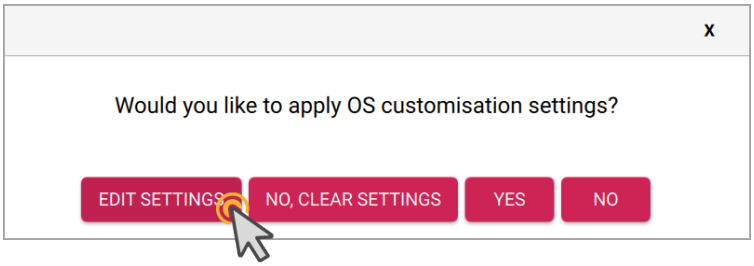
Step 7: Download the Configuration Files

Step 8: Set Up Terminal Windows

Step 9: Run a Connectivity Test (bping)

Step 1: Image the Pi OS on the MicroSD Card

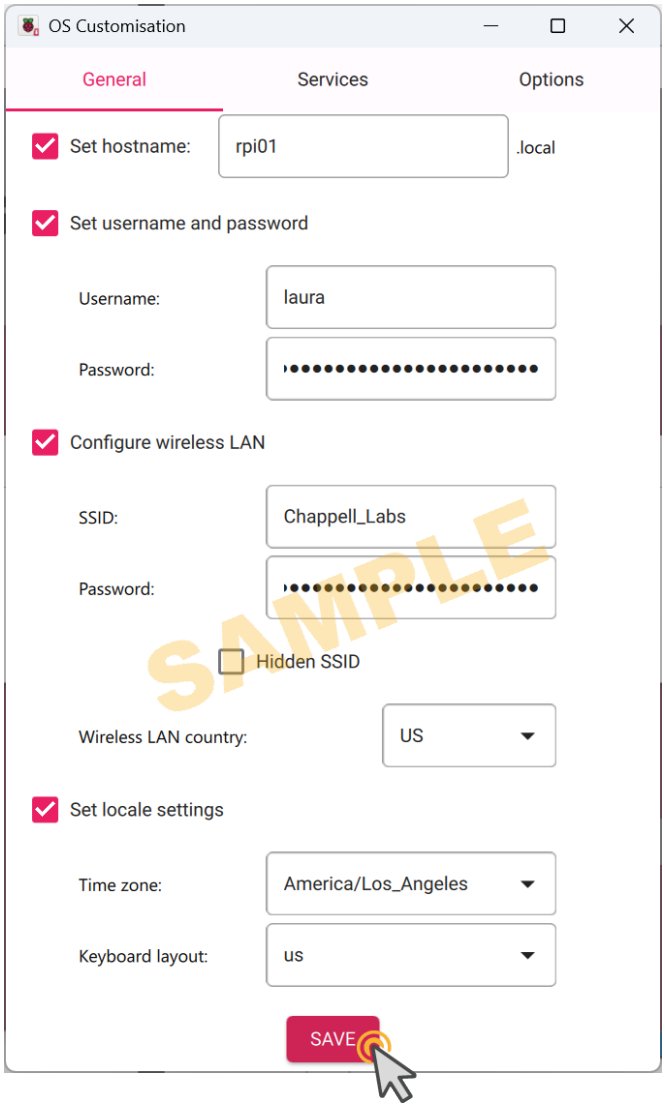
Raspberry Pi offers several different operating systems. In this lab, we will use the most recent Raspberry Pi OS, Debian Version 13 (“Trixie”) released October 1, 2025.

Location: desktop or laptop computer	
☐	Visit https://www.raspberrypi.com/software/ to download and install the latest version of the **Raspberry Pi OS Imager** tool.
☐	Insert a microSD card and choose: Device: **Raspberry Pi 5** Operating System: **Raspberry Pi OS (64-bit)** Storage: **Your microSD card** 
☐	Click **NEXT**.
☐	Now you will define basic customization for your Raspberry Pi. Select **Edit Settings** when prompted to customize your OS image. 

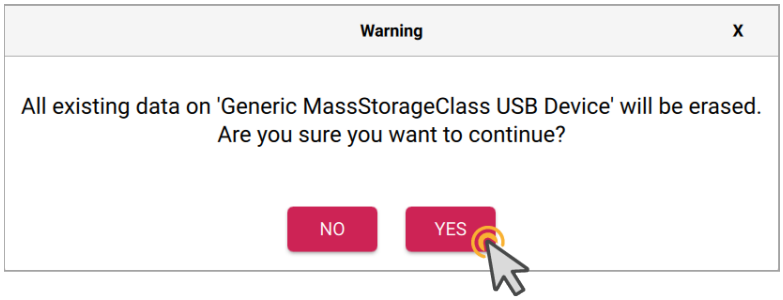
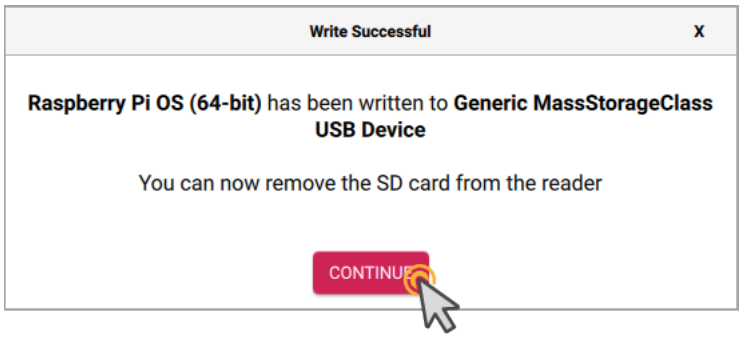
[Note: The hostname is not tied to any configuration files for this lab. It is only a recommendation. You will work with hostnames in the two-computer lab.]

☐ Configure the OS with the following settings (under the General menu):

Hostname:	`rpi01`
Username:	**Your first name (lowercase)**
Password:	**Assign a password**
WLAN:	**Customize based on your environment** (SSID, password, and country)
Locale:	**Customize based on your environment**
Keyboard:	**Customize based on your environment**



☐ After entering all your customization, click **SAVE**.

☐	When you are returned to the prompt to apply OS customization settings, click **YES**.  The Raspberry Pi Imager will download the image and install it onto the microSD card. This process can take up to 5 minutes. If you are imaging more than one microSD card, creating the next image will be faster because the imaged OS is downloaded by this tool.
☐	When the Write Successful window appears, click **CONTINUE**. 
☐	Remove the **microSD** card from the imaging computer.
☐	Plug the **microSD** card into your RPi-5.
☐	Power up your RPi-5.

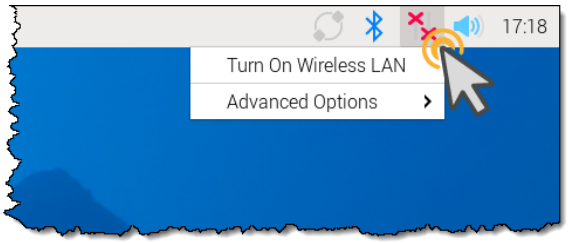
It will take a few moments for the Raspberry Pi to perform self-tests and power up.

These labs will be run by typing commands into the terminal windows directly on the Raspberry Pi.

In the future, you may be interested in using SSH (Secure Shell) to remotely access terminal windows on your Raspberry Pi. We will talk about remote access options in Lab 2.

Step 2: Update and Upgrade Packages

After you power on the Raspberry Pi, you will need to enable wireless, select a WLAN network, and update the packages on the Raspberry Pi.

Current Location: Raspberry Pi Main Screen	
☐	After booting up, you may need to **enable wireless and select a WLAN network** (even though you configured this using the Imager). 
☐	Once completed, test with the browser.
☐	Now let's manually update the packages on the Pi. Click the terminal icon to open a terminal window.
☐	`sudo apt update`
☐	`sudo apt upgrade` You will need to click "Y" when you are prompted to run the upgrade.

Your Raspberry Pi is now ready. It's time to download and build ION-DTN.

Terminal Commands to Know

All terminal commands are case-sensitive.

`[up arrow]`	Displays last commands (nice to fix typos or see last command)
`ls -l`	Lists file and directories in the current location
`cd directory`	Change to a subdirectory
`cd ..`	Move up one directory
`cd ION [tab]`	Sample using tab to auto-complete a directory name
`cd ~`	Return to home directory
`clear`	Clear screen
`wget url`	Get a file from the web via command line
`unzip file`	Uncompress a file
`nano file`	Open a file with the nano command-line editor tool
`cat file`	Display contents of a file
`export variable`	Set an environment variable
`source script`	Executes a script in the current shell
`printenv`	Prints current environment variables for shell session
`sudo command`	Run programs with administrative (root) privileges
`make`	Compiles and builds software from source code
`hostname`	Display or set the current system hostname
`ifconfig`	View and configure network interfaces (legacy command)
`ip addr`	View IP addresses
`tail -f file`	Watch a file in real time (e.g., logs)
`mkdir name`	Create a new directory
`rm file`	Delete a file
`rm -r folder`	Delete a directory and its contents
`cp source target`	Copy a file
`mv source target`	Move or rename a file
`chmod +x file`	Make a file executable
`passwd`	Change your password
`sudo reboot`	Reboot the Pi (add `now` to immediately reboot)

Step 3: Download ION-DTN

First, we will download the ION 4.1.3s distribution from Github[2].

Current Location: Raspberry Pi Main Screen

☐ Click the Terminal button on the menu to open a new terminal. The terminal window opens in your home directory.

Current Location: home directory

When you are in your home directory, the prompt is `fname@rpi01:~$` where `fname` is your first name (defined in Step 1)

☐ `wget https://github.com/nasa-jpl/ION-DTN/archive/refs/tags/\ ion-open-source-4.1.3s.zip`

(single line – if typing in the command, do not use a carriage return until you are ready to execute the command - we added the "`\`" in case you copy and paste this line in – the return will not be processed)

```
HTTP request sent, awaiting response... 302 Found
Location: https://codeload.github.com/nasa-jpl/ION-DTN/zip/refs/tags/ion-open-source-4
3s [following]
--2025-06-29 18:45:44--  https://codeload.github.com/nasa-jpl/ION-DTN/zip/refs/tags/io
pen-source-4.1.3s
Resolving codeload.github.com (codeload.github.com)... 140.82.116.10
Connecting to codeload.github.com (codeload.github.com)|140.82.116.10|:443... connecte
HTTP request sent, awaiting response... 200 OK
Length: unspecified [application/zip]
Saving to: 'ion-open-source-4.1.3s.zip.2'

ion-open-source-4.1.3s     [                    <=>  ]  70.82M  11.0MB/s    in 5.6s

2025-06-29 18:45:50 (12.7 MB/s) - 'ion-open-source-4.1.3s.zip.2' saved [74263381]

laura@rpi01:~ $
```

☐ `unzip ion-open-source-4.1.3s.zip`

[Press **Enter**. From this point on, remember to press **Enter** after each command.]

[2] Although 4.1.4 is available, this lab is designed and tested with version 4.1.3s.

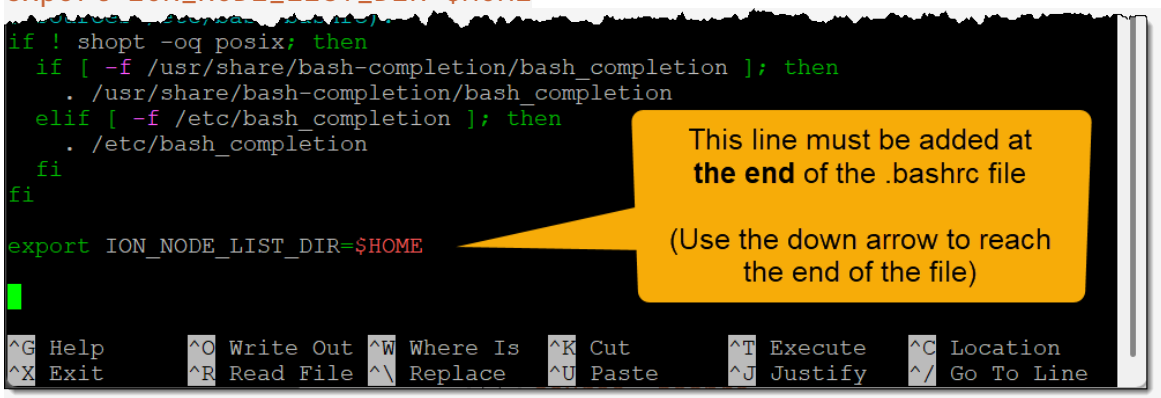
Step 4: Set an Environment Variable

Now, you will set an environment variable named ION_NODE_LIST_DIR and point it to your home directory path in your *.bashrc* file. The *.bashrc* file is a hidden configuration script that runs every time you open a new terminal session using the Bash shell.

Nano is a text editor that we will use to edit the *.bashrc* file.

QUICK TIP

Consider expanding your terminal window to full size to see the end of longer lines and files. Click the **up arrow** at the top right corner of the terminal window that you want to expand.

Current Location: home directory	
☐	`nano .bashrc` The .bashrc file opens. Some basic nano commands are listed at the bottom of the window.
☐	Scroll down (**with the down arrow key**) to the bottom of the *.bashrc* file and add this new line at the end. (Note that `HOME` is preceded by a dollar sign.) `export ION_NODE_LIST_DIR=$HOME` 
☐	Write out your changes and close *nano* (**Ctrl-o**, **return**, **Ctrl-x**)
☐	`source .bashrc` This executes the .bashrc file in the current directory.
☐	`printenv` Verify your **ION_NODE_LIST_DIR=/home/[username]** setting. If the entry does not appear or it is incorrect, run `nano .bashrc` again to review and edit the file, as needed. Run `source .bashrc` and `printenv` again after editing the file.

Step 5: Install Build Dependencies

Before we compile the ION-DTN code, we need to ensure that all the developer dependencies are installed.

Current Location: home directory	
☐	`sudo apt update && sudo apt install automake autoconf libtool \ m4 gcc make` (single line – if typing in the command, do not use a carriage return until you are ready to execute the command - we added the "\" in case you copy and paste this line in – the return will not be processed)

Step 6: Build ION-DTN

In Step 2, you unzipped the *ion-open-source-4.1.3s.zip* file. This created a new *ION-DTN-ion-open-source-4.1.3s* directory. Follow the steps below to change into that new directory and build ION-DTN.

Current Location: home directory	
☐	`cd ION-DTN-ion-open-source-4.1.3s`
Current Location: *ION-DTN-ion-open-source-4.1.3s* directory	
☐	`./configure` (If *configure* is not present, run `autoreconf -fi` and then `./configure`) The *autoreconf* tool is used to generate or update the *configure* script and related files. You may see some warnings as the *autoreconf* process runs.
☐	`make` **This process can take approximately 10 minutes.**
☐	`sudo make install` You may see some warnings as the *make* process runs.
☐	`sudo ldconfig` (This tells your system where to find newly installed software libraries.)

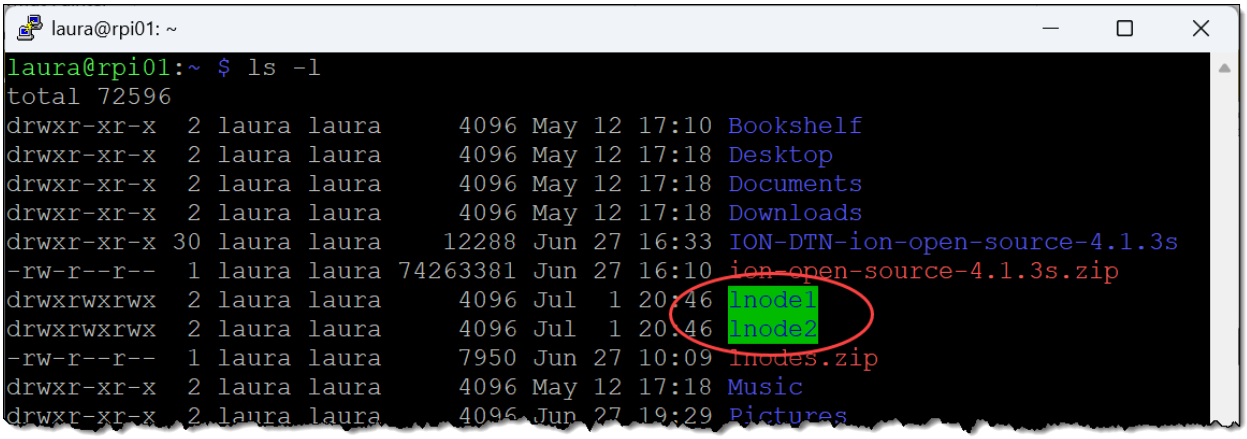
You've done it! Now let's get the lab configuration files and run some ION-DTN tests.

Step 7: Download the Configuration Files

As mentioned in the introductory presentation, running ION-DTN requires a set of configuration files and start scripts. At this point, you will download the initial configuration file set (*lnodes.zip*) for a two-node/single computer scenario.

Current Location: home directory	
☐	`cd ~` (return to your home directory)
☐	`wget -O lnodes.zip "https://bit.ly/725-lnodes"` (capital letter "O" – not a zero and *lnodes* begins with a lower-case "L")
☐	`unzip lnodes.zip` (This uncompresses two folders – **lnodes1** and **lnodes2**)
☐	`ls -l` (Verify the *lnode1* and *lnode2* directories are listed in your home directory)

This unzips two folders (**lnodes1** and **lnodes2**)



If you make a mistake and uncompress these two directories to the wrong location, consider using the File Manager tool to locate and delete the directories from the wrong location and start Step 6 again – being careful to download the zip file into your home directory.

The File Manager tool is launched from the Raspberry Pi main menu. 

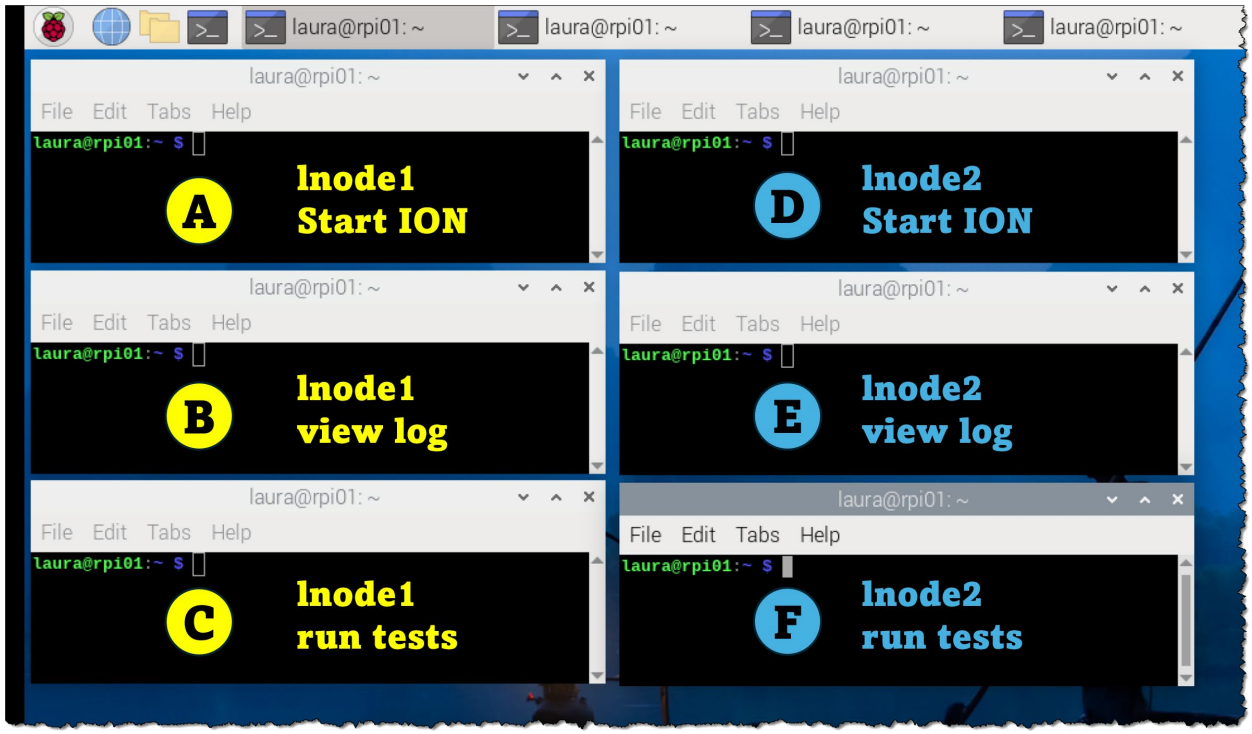
Step 8: Set Up Terminal Windows

We will set up two ION-DTN nodes (*lnode1* and *lnode2*) on one computer and send DTN communications between the hosts.

Each node will require three terminal windows.

Location: Raspberry Pi Main Screen	
☐	(Consider closing any existing terminal windows to start fresh.) Click the **terminal icon** six times to launch six new terminal windows.

We recommend organizing your terminal windows as shown in the following image.



To help identify which window you should enter commands in, we will color code lnode1 yellow and lnode2 bright blue.

QUICK TIP

As you continue your ION-DTN learning and tests, consider editing the *etc/hosts* file to include hostnames and IP addresses of the Raspberry Pi devices.

We will be editing our *etc/hosts* file in the Lab 2 next week.

Step 9: Run a Connectivity Test (bping)

Now that our terminal windows are set up, we are ready to send DTN communications between the hosts. We will begin with *bping*.

We use *bping* initially to verify connectivity between hosts (just as we do with ping, which uses ICMP, on IP networks). Observe the watch characters in the top terminal windows. Watch characters were discussed in the preceding presentation.

First, we will launch *ionstart*. in the top terminal windows for *lnode1* and *lnode2*.

Location: Terminal Window A (lnode1)		**Location: Terminal Window D (lnode2)**	
☐	`cd lnode1`	☐	`cd lnode2`
☐	`chmod +x ionstart` (makes the *ionstart* file executable)	☐	`chmod +x ionstart`
☐	`./ionstart`	☐	`./ionstart`

Next, we will set up the log file viewing terminals.

Location: Terminal Window B (lnode1)		**Location: Terminal Window E (lnode2)**	
☐	`cd lnode1`	☐	`cd lnode2`
☐	`tail -f ion.log` (shows the latest lines in *ion.log*)	☐	`tail -f ion.log`

Now let's launch the *bping* test from *lnode1*.

Location: Terminal Window C (lnode1)	
☐	`cd lnode1`
☐	`bping -c1 -q0 ipn:1.2 ipn:2.3`

The `bping -c1 -q0 ipn:1.2 ipn:2.3` command sends one bping (`-c1`) from node 1.2 to node 2.3 (`ipn:1.2 ipn:2.3`) and will stop as soon as you hit Ctrl-C (`-q0`). Note that *ionstart* automatically starts the *bpecho* service that *bping* relies on.

In ION-DTN, the **ipn:** addressing scheme is used for identifying endpoints in a Delay-Tolerant Network (DTN) using the IPN (Interplanetary Network) URI format standardized by the CCSDS. The format of IPN URIs is **`ipn:<node number>.<service number>`**. In this *bping* test, we are sending a *bping* from **node 1, service 2** to **node 2, service 3**.

Interested in the command parameters for *bping*?
Type `man bping` to bring up the user guide.

Wrapping Up

Congratulations on completing the first Raspberry Pi ION-DTN Lab!

You've taken your first steps into Delay-Tolerant Networking by building and running ION-DTN on a Raspberry Pi and executing a connectivity tests using *bping*. By completing each step—from imaging the OS and compiling source code to running multi-terminal node simulations—you've gained practical experience with interplanetary network protocols.

This hands-on lab was designed to provide a foundational understanding of ION-DTN in a single-node simulation environment. However, your learning journey doesn't stop here. We strongly encourage you to continue exploring, experimenting, and testing with the two-computer lab taking place next week. In Lab 2, we enable true inter-device communication offering a more realistic view of how Delay-Tolerant Networking will operate across nodes in space and other high-latency environments.

Lab 2 Requirements

> Two Raspberry Pi-5s with at least 4 GB RAM (with keyboard, mouse, monitor, and power supply for each), internet access via wireless or LAN (no blocking of bit.ly links), and a minimum 64 GB microSD card (and microSD card adapter for your laptop/desktop, if required). We recommend viewing the webinar on a laptop/desktop alongside your Raspberry Pi workstation.

Consider joining the ISOC Interplanetary Chapter (which automatically adds you to the IPNSIG group). Your participation supports the mission of IPNSIG to expand interplanetary networking and ensure the Internet truly becomes universal, even beyond Earth.

Questions? Comments? Corrections? Email us at the addresses below.

In addition, please reach out to us if you are interested in bringing ION-DTN labs to your school.

Laura Chappell
Chappell University
laura@chappellu.com

Scott Spicer
Chappell University
scott@chappellu.com

Appendix

Learn More about ION

For additional information on ION, refer to
https://ion-dtn.readthedocs.io/en/ion-open-source-4.1.3s/ION-Quick-Start-Guide
This lab used *"Build ION 4.1.3 (and earlier versions) without actual cipher suite"*.

Join IPNSIG (ISOC Interplanetary Chapter)

IPNSIG (Interplanetary Networking Special Interest Group) was founded in 1998 by Vint Cerf and researchers within academia and NASA/JPL. We are now a full Chapter within the Internet Society, known as the Interplanetary Chapter.

You can join us through the ISOC member portal (https://www.internetsociety.org/).

After creating your ISOC membership and logging in, click **Join a Chapter/SIG**, enter *Interplanetary Chapter*, and click **Search**.

Donate to IPNSIG

IPNSIG works to extend terrestrial networking into solar system space, which is consistent with the Internet Society's objectives to grow the Internet to unpopulated areas and connect the unconnected domain—and to ensure that even in space, "The Internet is for Everyone".

IPNSIG, Inc. is an exempt organization as described in Section 501(c)(3) of the US Internal Revenue Code; EIN 87-4782452.

Please consider donating (https://www.ipnsig.org/donate) to help us continue our quest.