Mac Mini Network Configuration

Mac Mini Network Configuration: Breaking Up Thunderbolt Bridge for Fine-Grained Control

When working with Mac Minis, particularly when using Thunderbolt for high-speed networking, you might run into limitations with macOS's default behavior of grouping all Thunderbolt ports under one Thunderbolt Bridge. By default, this setup combines all the Thunderbolt ports into a single network interface, which doesn't allow for individual point-to-point control of each Thunderbolt connection. In this post, we'll go through the steps to break up this Thunderbolt Bridge into three separate network services, each tied to a specific Thunderbolt port, enabling better control and flexibility for networking between nodes.

The goal is to bypass Thunderbolt PCI device-tree routing and instead create individual network services for each Thunderbolt port. This provides full control of point-to-point routing while maintaining the networking stack involved for communication. Here's how to set up a Mac Mini for neighbor-to-neighbor communication over Thunderbolt ports.

Step 1: Initial Network Configuration

Before any configuration, the network services should look something like this after a fresh macOS installation.

```
~ % networksetup –listallnetworkservices
An asterisk (*) denotes that a network service is disabled.
Ethernet
Thunderbolt Bridge
Wi–Fi
```

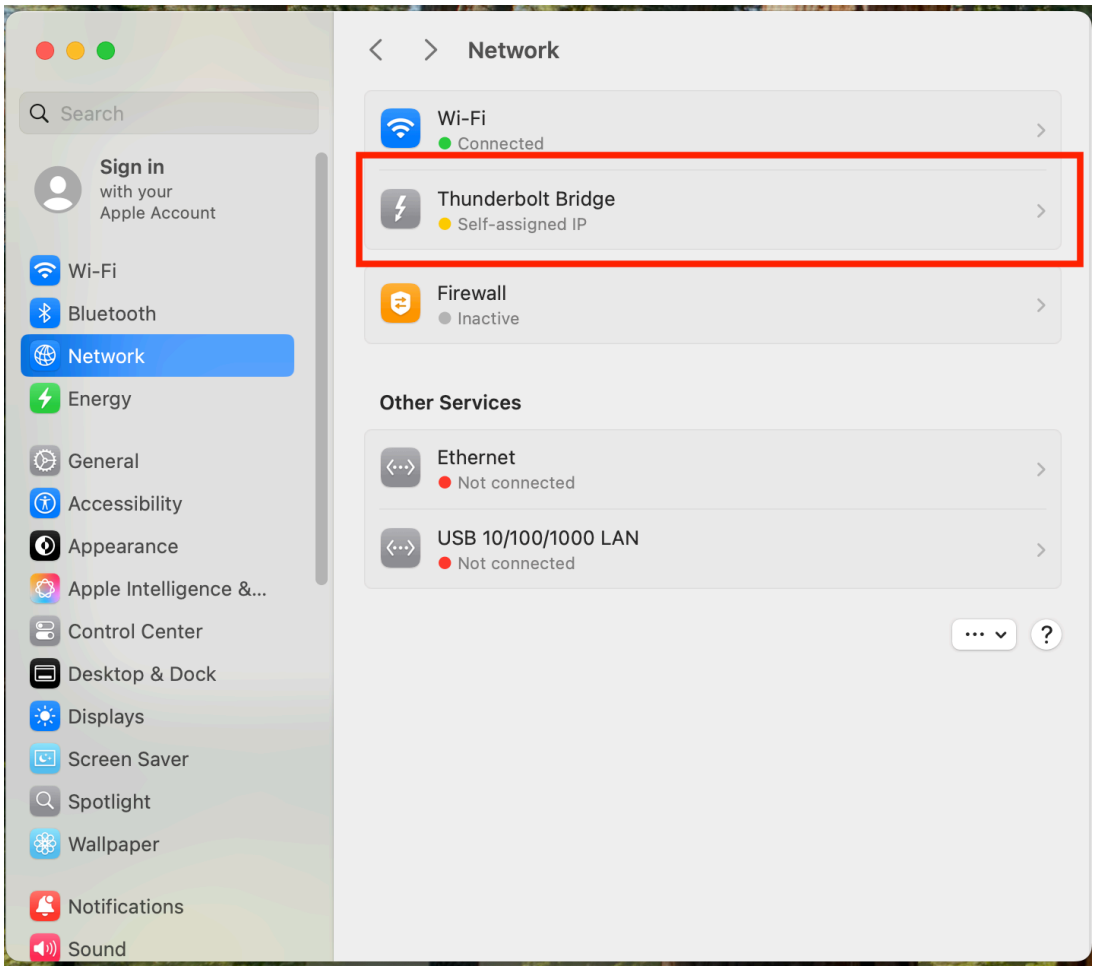
We need to get rid of the Thunderbolt Bridge service, as it will bridge all 3 thunderbolt ports into one network service, which we don't want. Instead, we are going to create three new networking services, one for each thunderbolt port named "thunderbolt1" "thunderbolt2" and "thunderbolt4". I am following the mac-mini device naming conventions from the settings app.

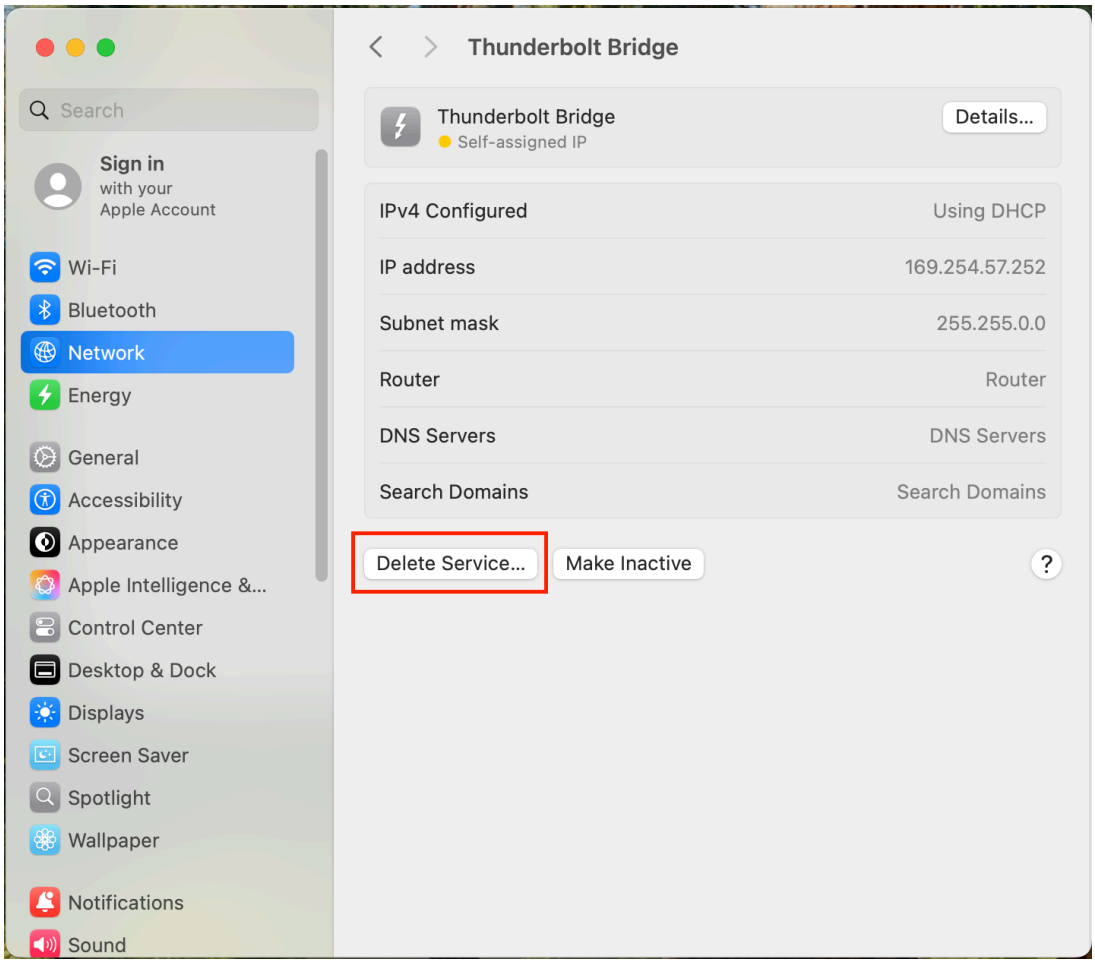
Step 2: Disable Thunderbolt Bridge

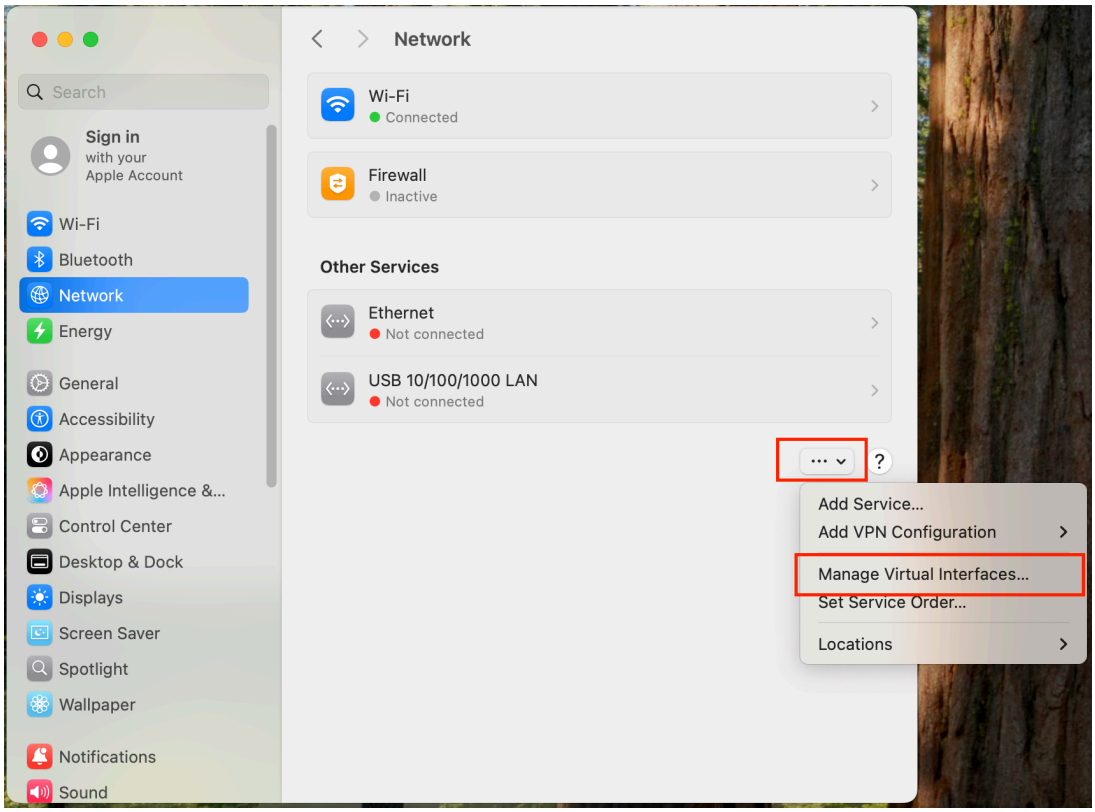
The first thing we need to do is disable the **Thunderbolt Bridge** service. This can be done via the following command:

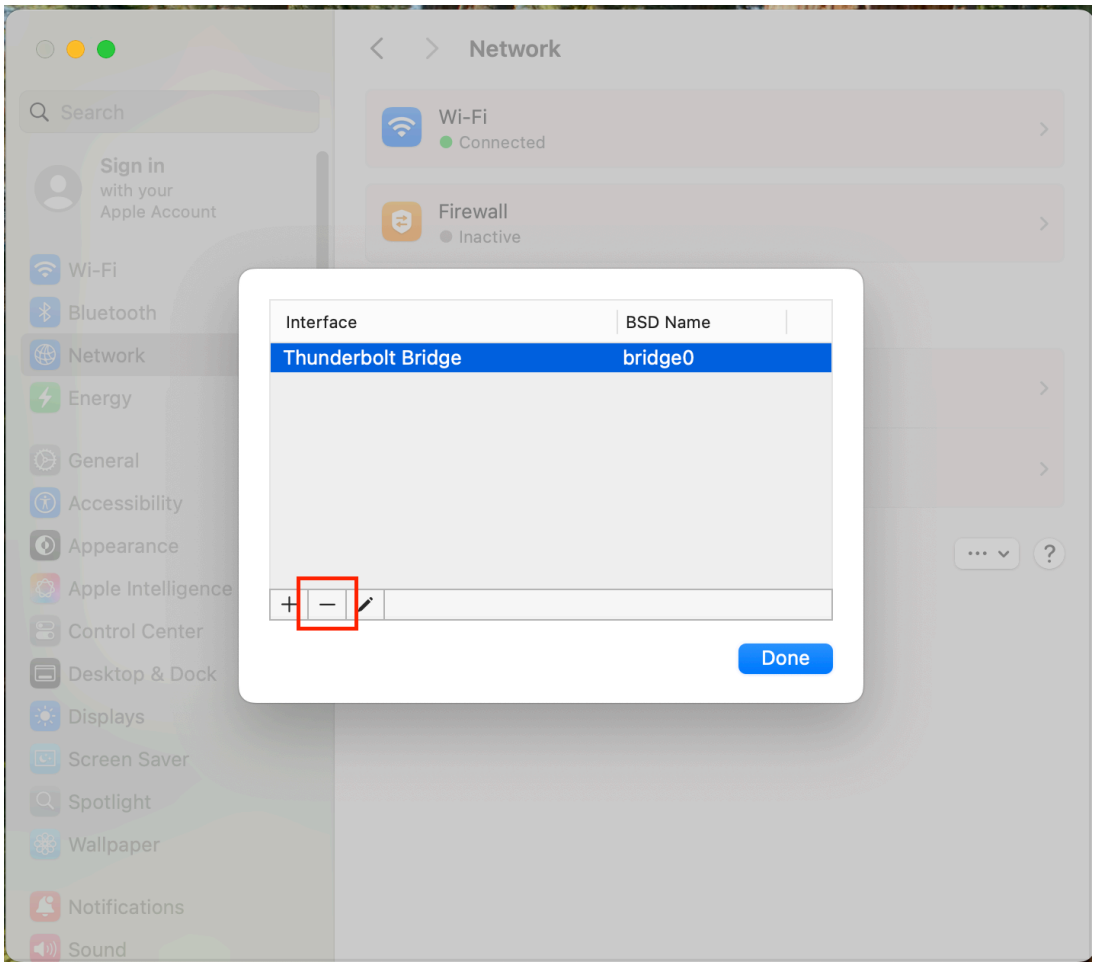
```
~ % networksetup –setnetworkserviceenabled Thunderbolt\ Bridge
off
```

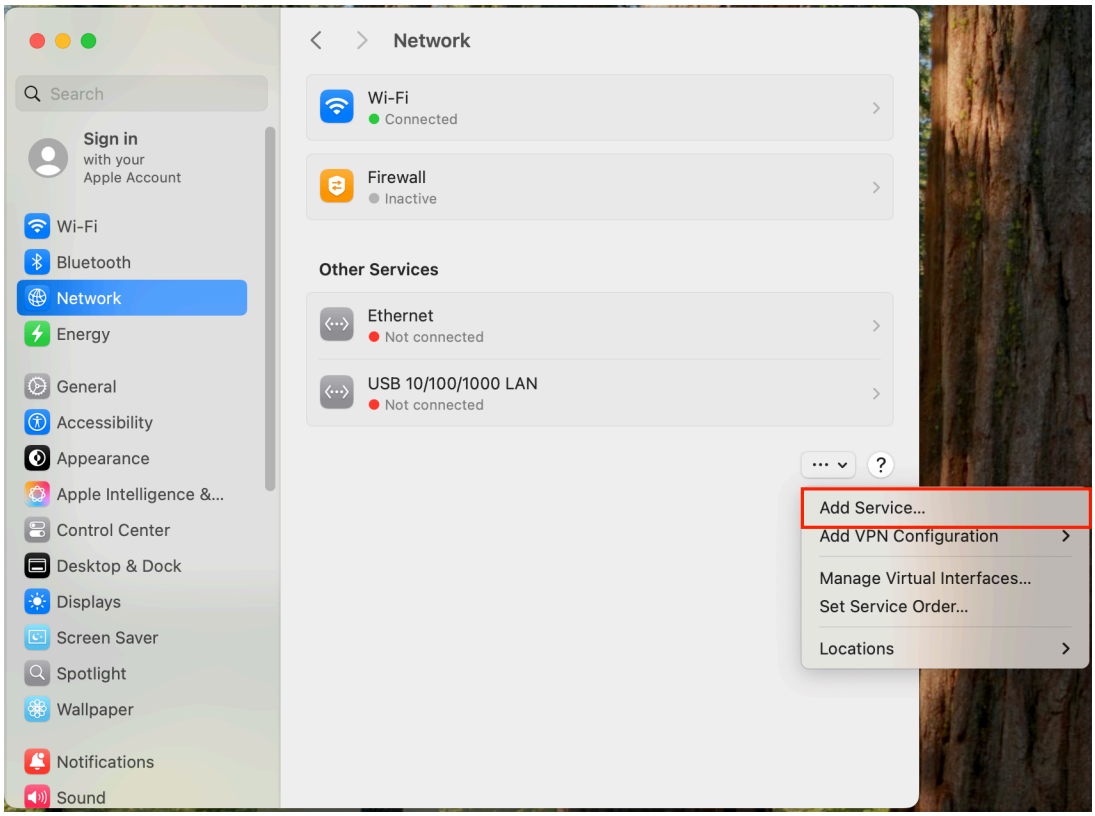
While this can be done through the Terminal, there are additional UI steps required to fully delete the Thunderbolt Bridge, particularly related to system permissions. This service, once disabled, will no longer bridge all ports into a single service.

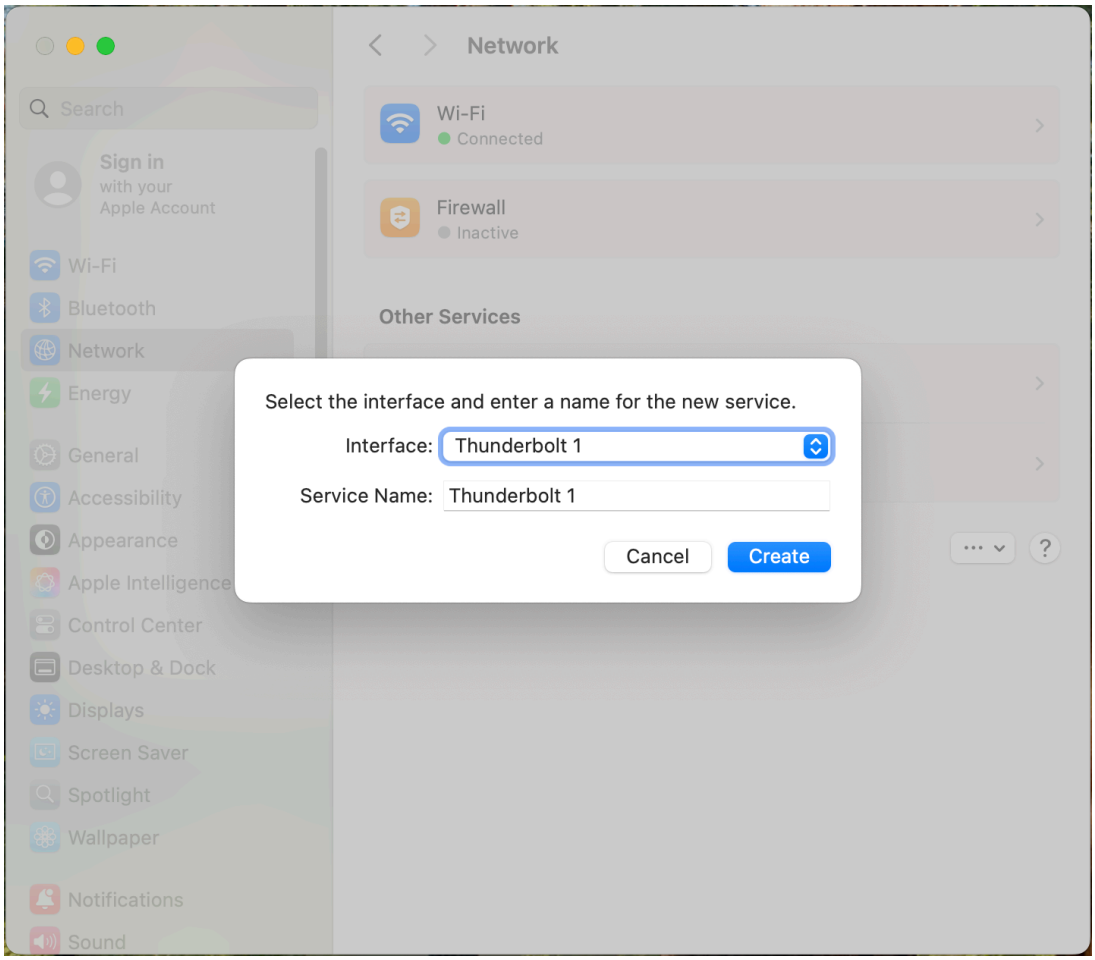



Search
Sign in
with your
Apple Account
Wi-Fi
Bluetooth
Network
Energy
General
Accessibility
Appearance
Apple Intelligence &...
Control Center
Desktop & Dock
Displays
Screen Saver
Spotlight
Wallpaper
Notifications
Sound
Thunderbolt Bridge
Thunderbolt Bridge
Self-assigned IP
Details...
IPv4 Configured
Using DHCP
IP address
169.254.57.252
Subnet mask
255.255.0.0
Router
Router
DNS Servers
DNS Servers
Search Domains
Search Domains
Delete Service...
Make Inactive
?


Search
Sign in
with your
Apple Account
Wi-Fi
Bluetooth
Network
Energy
General
Accessibility
Appearance
Apple Intelligence &...
Control Center
Desktop & Dock
Displays
Screen Saver
Spotlight
Wallpaper
Notifications
Sound
Network
Wi-Fi
Connected
Firewall
Inactive
Other Services
Ethernet
Not connected
USB 10/100/1000 LAN
Not connected
Add Service...
Add VPN Configuration
Manage Virtual Interfaces...
Set Service Order...
Locations


Network
Search
Sign in
with your
Apple Account
Wi-Fi
Bluetooth
Network
Energy
General
Accessibility
Appearance
Apple Intelligence
Control Center
Desktop & Dock
Displays
Screen Saver
Spotlight
Wallpaper
Notifications
Sound
Wi-Fi
Connected
Firewall
Inactive
Interface
BSD Name
Thunderbolt Bridge
bridge0
Done


Search
Sign in
with your
Apple Account
Wi-Fi
Bluetooth
Network
Energy
General
Accessibility
Appearance
Apple Intelligence &...
Control Center
Desktop & Dock
Displays
Screen Saver
Spotlight
Wallpaper
Notifications
Sound
Network
Wi-Fi
Connected
Firewall
Inactive
Other Services
Ethernet
Not connected
USB 10/100/1000 LAN
Not connected
Add Service...
Add VPN Configuration
Manage Virtual Interfaces...
Set Service Order...
Locations



Step 3: Create New Network Services

Next, we need to create three separate network services for each of the Thunderbolt ports. This setup will allow us to treat each port independently, which is essential for point-to-point communication between nodes.

```
~ % networksetup -createnetworkservice "Thunderbolt 1" en2
~ % networksetup -createnetworkservice "Thunderbolt 2" en3
~ % networksetup -createnetworkservice "Thunderbolt 4" en4
```

In these commands:

- **en2**, **en3**, and **en4** correspond to the Thunderbolt interfaces on your Mac Mini.
- We name the services "Thunderbolt 1", "Thunderbolt 2", and "Thunderbolt 4", following the default macOS naming conventions for clarity.

Now, we need to configure the IP addresses that are present on each node. For convenience, and to keep all IPs on the direct-connections across the network

unique, IP addresses will be given out by NodeId and PortId. NodeId is a number from 1-256, and PortId is 10/20/30. Each node can only have three ports, so these are the only possible values on the network. Every node will be uniquely identified by the third byte of the ip address. This is a useful crutch for early forms of the neighbor discovery protocols, as we can use arping to get the ip/MAC addresses of the neighboring node and immediately know the host and port id.

Step 4: Configure IP Addresses

Now that we have separate network services for each Thunderbolt port, we need to configure them with unique IP addresses. This is important for ensuring each port communicates on its own distinct network. To achieve this, we use a structured IP addressing scheme based on NodeId and PortId, where:

- **NodeId** ranges from 1-256
- **PortId** can be 10, 20, or 30 (for Thunderbolt 1, 2, and 4, respectively)

For example, if NodeId = 1, the IP addresses would be:

```
~ % networksetup –setmanual "Thunderbolt 1" 196.254.1.10
255.255.0.0
~ % networksetup –setmanual "Thunderbolt 2" 196.254.1.20
255.255.0.0
~ % networksetup –setmanual "Thunderbolt 4" 196.254.1.30
255.255.0.0
```

These IP addresses will allow each port to communicate independently and maintain unique identifiers for each device on the network. The subnet mask **255.255.0.0** allows for sufficient addressing within this direct-connect setup.

Conclusion: Scaling and Further Use

With this configuration, you now have individual control over each Thunderbolt port, enabling direct point-to-point communication between your Mac Minis. This setup provides a robust foundation for building custom networking solutions, especially useful in environments where high-speed communication over Thunderbolt is necessary (e.g., data centers, high-performance computing clusters).

For those who want to scale this setup, consider automating the configuration with a script, or setting up a system where devices can dynamically discover and configure connections using protocols like mDNS or more sophisticated neighbor discovery protocols.

As with any network setup, testing is key, so ensure all configurations are correct, and monitor the system for optimal performance. Troubleshooting steps like using **ping**, **arping**, or reviewing **syslog** can help in quickly identifying and resolving

issues.

Thunderbolt networking gotcha’s

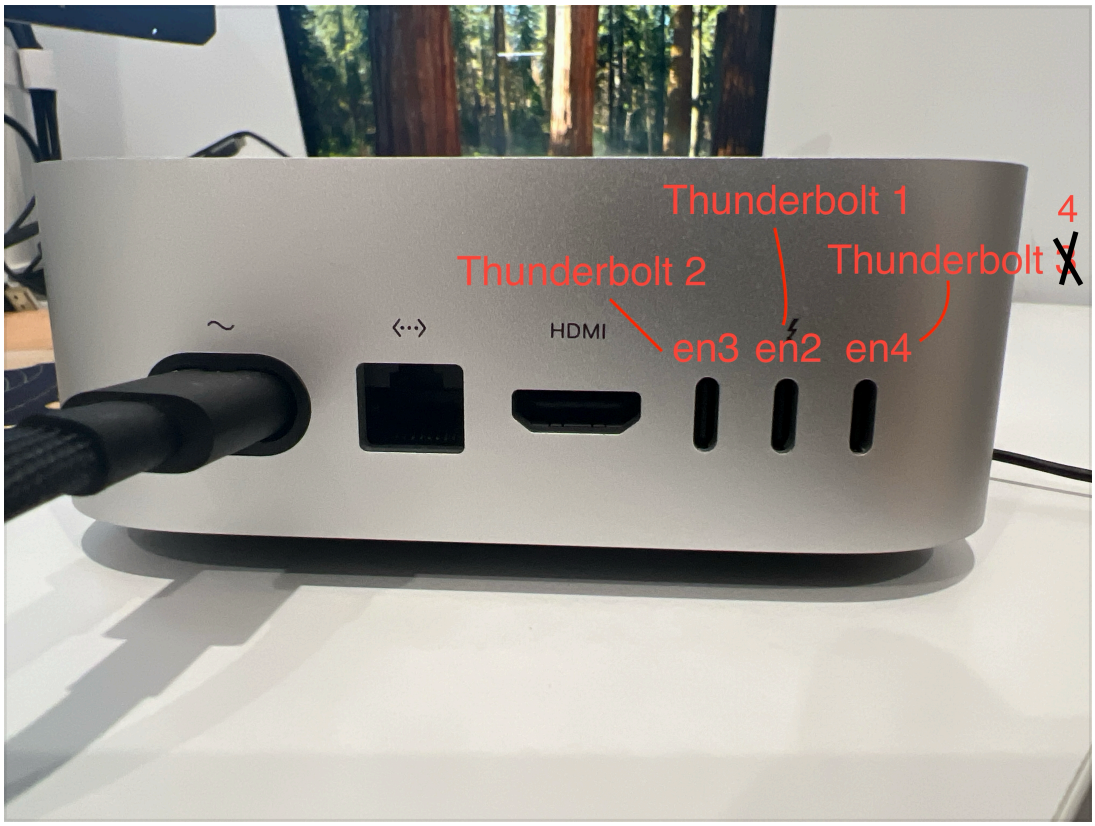
Now that the Mac mini’s thunderbolt bridge has been broken up into three network interfaces, it is up to us to bring up a link. An emulated link relies on a UDP connection to its neighbor, as building on top of UDP provides a datagram based abstraction of a link that’s easier to work with than ethernet frames. A UDP connection requires a “server” side to bind to an IP/Port to serve requests on, and a “client” side that sends datagrams to the server. The server side simply sends frames out to the IP which sent a packet to the port. Since the thunderbolt ports are on separate network services with their own IP, different threads listening on the same port but different addresses will only receive messages from their respective links. This leaves only one problem, the client side needs to know the IP of the server on the other side to establish a connection.

To get info of a network service named “Thunderbolt 2”

```
~ % networksetup -getinfo Thunderbolt\ 2
Manual Configuration
IP address: 169.254.1.20
Subnet mask: 255.255.0.0
Router: (null)
IPv6: Automatic
IPv6 IP address: none
IPv6 Router: none
Ethernet Address: 36:0a:57:3b:c0:84
```

```
Flush ARP Cache
~ % sudo arp -a -d
```

Ethernet hardware ports marked with their interface names


Thunderbolt 1
4
Thunderbolt 2
Thunderbolt 3
HDMI
en3
en2
en4