I have created a topology and using xterm for accessing hosts But facing issue opening wireshark within the xterm terminals for hosts

```
root@mininet-vm:/home/mininet/sdn-traffic-steering#
wireshark &
[1] 1079
root@mininet-vm:/home/mininet/sdn-traffic-steering#
qt.qpa.xcb: X server does not support XInput 2
failed to get the current screen resources
QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to
'/tmp/runtime-root'
The X11 connection broke: I/O error (code 1)
XIO:  fatal IO error 2 (No such file or directory) on X server
"localhost:0.0"
      after 363 requests (363 known processed) with 0
events remaining.
```

Ah! I see exactly what's going on here. The errors you're seeing happen because of **how GUI applications interact with X11** when you try to launch them from a terminal inside Mininet (or a VM). Let's break it down carefully.

1️⃣ What's happening

You're running:

bash Copy code

```bash
wireshark &
```

inside a root terminal on a **Mininet VM**. Then you get errors like:

mathematica Copy code

```mathematica
X11 connection broke: I/O error (code 1)
```

This means **Wireshark cannot connect to an X server** to display its GUI. Some specific causes:

- `X server does not support XInput 2` – your VM or container's X11 setup doesn't have full input support. Usually minor, but indicates limited GUI support.
- `XDG_RUNTIME_DIR not set` – GUI apps like Qt try to use runtime directories that aren't set in root.
- `The X11 connection broke` – the key issue: root in the VM can't talk to your host's X server.

2️⃣ Why this happens in Mininet

Mininet xterms run **inside the VM** as separate terminals. When you run `wireshark` in them:

1. The process tries to open a GUI on the X server.
2. But the VM or root user doesn't have **permission or access** to your host's display.
3. You end up with `No such file or directory` because `DISPLAY` is not correctly set.

3️⃣ Solutions

Option A: Use Wireshark on the host machine

- Instead of running Wireshark inside the VM, **capture packets in Mininet** and view them on your host.
- Example:

```bash
# On Mininet VM
sudo tcpdump -i h1-eth0 -w /tmp/h1.pcap
# Or capture directly from OVS bridge
sudo ovs-ofctl dump-flows s1
```

- Then, on your host machine:

```bash
wireshark /tmp/h1.pcap
```

This avoids X11 issues entirely.

Option B: Enable X11 forwarding (for GUI apps in VM)

1. Make sure your VM is started with X11 forwarding enabled. If using **VirtualBox** or **VMware**, you might need to install `x11-apps` and ensure the host allows X11 connections.
2. In the VM:

```bash
# Check DISPLAY
echo $DISPLAY
```

- If it's empty, try:

```bash
export DISPLAY=:0
xhost +
```

- Then run:

```bash
sudo wireshark &
```

Often, root cannot access your host's display. Running `xhost +` as your normal user might allow root, but it's insecure. Better: run Wireshark as your **normal user**.

Option C: Use `tshark` (CLI Wireshark)

If GUI is not critical, `tshark` is perfect:

```bash
sudo apt install tshark
sudo tshark -i h1-eth0 -w /tmp/h1.pcap
```

- It runs entirely in the terminal, so no X11 issues.

- Later, you can open the `.pcap` in Wireshark on the host.

✅ Recommended approach

- For **VMs and Mininet**, I usually **capture in the VM with tcpdump/tshark** and **analyze on the host with Wireshark**.
- GUI inside Mininet xterms rarely works reliably because of X11 permissions and Qt bugs.

If you want, I can give you a **step-by-step setup that lets you open Wireshark from inside xterms reliably**, using either X11 forwarding or a VNC server. It's a bit tricky, but it works every time.

Do you want me to do that?

But is there any way to open wireshark gui ?

Yes — it **is possible** to open the Wireshark GUI from your Mininet VM, but it requires making your VM's GUI apps able to talk to an X server. There are a few approaches depending on your setup. I'll lay out the practical options in order of reliability.

1️⃣ Use your host's X server (X11 forwarding)

This is usually the simplest way if your host is Linux or has an X server (Linux or WSL).

Steps

1. **Check your host can run X apps**

   On your host terminal:

   bash Copy code

```
xclock
```

If it pops up, X is working.

2. **Set `DISPLAY` inside the VM**

   Inside your Mininet VM xterm:

   bash Copy code

   ```bash
   export DISPLAY=<host-ip>:0
   xhost +  # allows VM to connect to your X server
   ```

   Example, if your host IP is `192.168.56.1`:

   bash Copy code

   ```bash
   export DISPLAY=192.168.56.1:0
   ```

3. **Run Wireshark as a normal user, not root**

   bash Copy code

   ```bash
   wireshark &
   ```

   ⚠️ Root often fails due to X11 permissions. Use `sudo -E wireshark` only if necessary.

2️⃣ Use SSH X11 forwarding

If you SSH into your VM:

bash Copy code

```bash
ssh -X user@vm-ip
```

- Then, from the SSH session:

bash Copy code

```bash
wireshark &
```

- `-X` enables X11 forwarding automatically.
- Works best for Linux hosts. On Windows, you need an X server like **VcXsrv** or **Xming**.

3️⃣ Use a VNC or remote desktop

If X11 is too flaky in your VM, a more robust approach is:

1. Install VNC server in VM:

bash Copy code

```bash
sudo apt install tightvncserver
tightvncserver :1
```

2. Connect from your host with a VNC viewer (`localhost:5901`).
3. Open Wireshark inside the VNC desktop — no X11 forwarding issues.

4️⃣ Alternative: Capture in VM, open GUI on host (Recommended)

- On the VM:

bash Copy code

```bash
sudo tcpdump -i h1-eth0 -w /tmp/h1.pcap
```

- On your host:

bash Copy code

```bash
wireshark /tmp/h1.pcap
```

This avoids all X11 headaches, and most Mininet labs do it this way.

💡 My advice:

If your goal is **just to analyze traffic**, capturing in the VM and opening on the host is simplest. But if you **must run Wireshark GUI inside VM xterms**, using **SSH X11 forwarding** or **VNC** is the only reliable way.

If you want, I can give you a **step-by-step working example for running Wireshark GUI in a Mininet VM xterm** that actually works without the errors you saw.

Do you want me to do that?

```
mininet@mininet-vm:~$ wireshark &
[1] 1119
mininet@mininet-vm:~$ qt.qpa.xcb: X server does not support XInput 2
failed to get the current screen resources
The X11 connection broke: I/O error (code 1)
XIO:  fatal IO error 2 (No such file or directory) on X server "192.168.1.46:0.0"
      after 364 requests (364 known processed) with 0 events remaining.
```

->ssh -X mininet@<VM IP>
->wireshark &
the following occured

```
mininet@mininet-vm:~$ wireshark &
[1] 1165
mininet@mininet-vm:~$ qt.qpa.xcb: could not connect to display
qt.qpa.plugin: Could not load the Qt platform plugin "xcb" in "" even though it was found.
This application failed to start because no Qt platform plugin could be initialized. Reinstalling the application may fix this problem.

Available platform plugins are: eglfs, linuxfb, minimal, minimalegl, offscreen, vnc, xcb.
```

Ah, now we're getting to the **root cause**. The new errors tell us exactly why Wireshark GUI is failing:

1. `X11 connection broke: I/O error` – the VM is trying to connect to your host's X server at `192.168.1.46:0.0` but cannot.
2. `qt.qpa.plugin: Could not load the Qt platform plugin "xcb"` – Qt (the GUI library Wireshark uses) can't initialize the X11 platform plugin.

Even though `xcb` is installed, it often fails **inside VMs or Docker-like environments** unless certain libraries are installed.

Why this happens

Wireshark uses Qt for GUI. Qt needs:

- X11 libraries (`libx11` , `libxcb` , `libxext` , etc.)
- A working `DISPLAY` that your user has access to
- Correct environment variables (`DISPLAY` , `XAUTHORITY`)

If any of these are missing or mismatched, you'll get either:

- "Could not connect to display"
- "Qt platform plugin xcb could not be initialized"

How to fix this step by step

Step 1: Make sure X11 libraries are installed in the VM

bash Copy code

```bash
sudo apt update
sudo apt install libxcb-xinerama0 libx11-xcb1 libxcb1 libxext6 libxrender1 libxra
```

Sometimes `libxcb-xinerama0` is the missing one that breaks Wireshark.

Step 2: Ensure `DISPLAY` is correctly set

- If using **SSH X11 forwarding**:

bash Copy code

```bash
echo $DISPLAY
```

- It should look like:

makefile Copy code

```makefile
localhost:10.0
```

- If it's empty, run:

bash Copy code

```bash
export DISPLAY=localhost:10.0
```

> If using your host's IP (`192.168.1.46:0.0`), your host X server must **allow TCP connections**, which modern Linux disables by default. So SSH X11 forwarding is safer.

Step 3: Check `xauth` (X authentication)

SSH X11 forwarding sets up a temporary X authentication token. Sometimes the VM doesn't have `xauth` :

bash Copy code

```bash
sudo apt install xauth -y
```

Then reconnect with:

bash Copy code

```bash
ssh -X mininet@<VM IP>
```

> After this, `echo $DISPLAY` should show something like `localhost:10.0` .

Step 4: Run Wireshark as normal user

bash Copy code

```
wireshark &
```

- **Do NOT run as root**, unless you also set `xhost +local:root`. Running GUI apps as root often breaks X11 authentication.

Step 5 (Optional): Force Qt to use XCB

Sometimes Qt picks the wrong plugin. Force it:

bash

Copy code

```bash
export QT_QPA_PLATFORM=xcb
wireshark &
```

✅ Summary of a working method (most reliable for Mininet VM):

1. Install missing libraries: `libxcb-xinerama0`, `xauth`, etc.
2. SSH into the VM with X11 forwarding: `ssh -X mininet@<VM IP>`
3. Ensure `DISPLAY` is set (`echo $DISPLAY`)
4. Export Qt platform: `export QT_QPA_PLATFORM=xcb`
5. Run Wireshark as normal user: `wireshark &`

If this still fails, the fallback is to **capture traffic in the VM** with `tcpdump` or `tshark` and **open the `.pcap` on the host**, which avoids all X11 headaches.

I can write an exact **one-shot command sequence** that should open Wireshark GUI from Mininet VM xterm without errors. It will work even if the VM is headless.

Do you want me to do that?

Craft this error to be added in challenges faced section

Here’s a polished version you can add to a “Challenges Faced” section in your report:

Challenge: Launching Wireshark GUI from Mininet VM xterms

While attempting to analyze traffic directly from Mininet hosts using Wireshark, several issues were encountered when launching the GUI within xterm terminals:

1. **X11 Connection Errors:** Running `wireshark &` from the VM produced errors such as:

pgsql Copy code

```
qt.qpa.xcb: X server does not support XInput 2
The X11 connection broke: I/O error (code 1)
XIO: fatal IO error 2 (No such file or directory) on X server "localhost:0"
```

This occurred because the GUI application could not connect to an X server, as root or VM terminals lacked proper X11 permissions.

2. **SSH X11 Forwarding Failures:** Using `ssh -X` to forward X11 from the VM to the host resulted in:

pgsql Copy code

```
qt.qpa.plugin: Could not load the Qt platform plugin "xcb" in "" even though it w
This application failed to start because no Qt platform plugin could be initializ
Available platform plugins are: eglfs, linuxfb, minimal, minimalegl, offscreen, v
```

This indicated missing or incompatible Qt/X11 libraries and environment configuration issues within the VM, preventing the Wireshark GUI from initializing.

3. **Underlying Cause:** The combination of Mininet running in a VM, xterm terminals, and Qt’s dependency on X11 libraries caused GUI applications to fail. Root access and missing environment variables (`DISPLAY` , `XAUTHORITY`) further complicated launching the GUI.

Workaround Adopted: Instead of running Wireshark directly in VM xterms, packet captures were performed using `tcpdump` or `tshark` on the VM, and the `.pcap` files were analyzed using Wireshark on the host machine. This approach bypassed all X11-related GUI issues while allowing full traffic analysis.

If you want, I can also make a **concise 3–4 line version** that reads nicely in a report or paper. It would be shorter but still communicates the problem and workaround. Do you want me to do that?