# MAWLANA BHASHANI SCIENCE AND TECHNOLOGY UNIVERSITY

**Santosh, Tangail – 1902**


**Course Title : Telecommunication Engineering**

**Assignment : OpenFlow protocol Name**

**Assignment No. : 02**

**Submitted by,**

**Name : KAZI MUJAHID MUNTASIR**

**ID: IT-17024**

**Session: 2016-17**

**Submitted to,**

**NAZRUL ISLAM**

**Assistant Professor**

**Dept. of ICT, MBSTU.**

**Question 5.1: How the RYU GUI interface can be improved? Provide some ideas**.

Answer: Web design consists, for the most part, of interface design. There are many techniques involved in crafting beautiful and functional interfaces. Here's my collection of 10 that I think you'll find useful in your work. They're not related to any particular theme, but are rather a collection of techniques I use in my own projects. Without further ado, let's get started.

**1. Padded block links**

Links (or anchors) are inline elements by default, which means that their clickable area spans only the height and width of the text. This clickable area, or the space where you can click to go to that link's destination, can be increased for greater usability. We can do this by adding padding and, in some cases, also converting the link into a block element.

**2. Typesetting buttons**

Attention to every detail is what separates a great product from a mediocre one. Interface elements such as buttons and tabs are clicked on many times a day by your users, so it pays to typeset them properly; and by typesetting I mean positioning the label.

**3. Using contrast to manage focus**

Similarly, you can also manage the focus of your visitors' attention with contrast between elements.

**4. Using color to manage attention**

Color can also be used to effectively focus your visitors' attention on important or actionable elements. For example, during the US presidential election, pretty much all of the candidates' websites had the donation button colored red. Red is a very bright and powerful color so it attracts attention, and it stands out even more when the rest of the website is blue or another colder color. White space indicates relationships

**5. Letter spacing**

Web design is pretty limiting for typographers. But while there are only a few safe Web fonts and not a great many things you can do to style them, it's worth remembering that we do still have some level of control. "Tracking" is a term used in the field of typography to describe the adjustment of spacing between letters in words. We've got the ability to do this with CSS using the letter-spacing property.

**6. Auto-focus on input**

Many Web applications and websites feature forms. These may be search forms or input forms inviting you to submit something. If this form is the core feature of our application or website, you may want to consider automatically focusing the user's cursor on the input field when the website loads. This will speed things up because users can start typing right away without having to click on it.

**7. Custom input focus**

While the default look of form elements suffices for most functions, sometimes we want something a little prettier or a little more standardized across various browsers and systems. We can style input fields by simply targeting it with an "id," "class" or plain old "input," like so.

**8. Hover controls**

Some Web applications have extra utility controls, such as edit and delete buttons, that don't necessarily have to be shown beside every item at all times. They can be hidden to simplify the interface and focus visitors' attention on the main controls and content.

**Question 5.2: Explain at least two the advantage and disadvantage of using Mininet or Zodiac FX?**

Answer: Mininet combines many of the best features of emulators, hardware test beds, and simulators.

Compared to full system virtualization based approaches, Mininet:

- **Boots faster**: seconds instead of minutes
- **Scales larger**: hundreds of hosts and switches vs. single digits
- **Provides more bandwidth**: typically 2Gbps total bandwidth on modest hardware
- **Installs easily**: a prepackaged VM is available that runs on VMware or Virtual Box for Mac/Win/Linux with OpenFlow v1.0 tools already installed.

**Disadvantages of Mininet:**

Mininet-based networks cannot (currently) exceed the CPU or bandwidth available on a single server.

Mininet cannot (currently) run non-Linux-compatible OpenFlow switches or applications; this has not been a major issue in practice.

**Question 5.3: Explain how the open flow tables are created?**

Answer: Using the OpenFlow switch protocol, the controller can add, update, and delete flow entries in *flow tables*, both reactively (in response to packets) and proactively.
Reactive Flow Entries are created when the controller dynamically learns where devices are in the topology and must update the flow tables on those devices to build end-to-end connectivity. For example, since the switches in a pure OpenFlow environment are simply forwarders of traffic, all rational logic must first be dictated and programmed by the controller. So, if a host on switch a needs to talk to a host switch B, messages will be sent to the controller to find out how to get to this host. The controller will learn the host MAC address tables of the switches and how they connect, programming the logic into the *flow tables* of each switch. This is a reactive flow entry.

**Proactive Flow Entries** are programmed before traffic arrives. If it's already known that two devices should or should not communicate, the controller can program these *flow entries* on the OpenFlow endpoints ahead of time.

**Question 5.6: Provide your personal opinion about OpenFlow protocol? Focus on the functionalities and flexibility.**

Answer: OpenFlow is a programmable network protocol designed to manage and direct traffic among routers and switches from various vendors. It separates the programming of routers and switches from underlying hardware. It is the result of a six-year research collaboration between Stanford University and the University of California at Berkeley. The OpenFlow Network Architecture consists of three layers:

(1) One or more OpenFlow virtual and/or physical switches;

(2) One or two OpenFlow controller and,

(3) One or more OpenFlow application.

The technology consists of three parts: flow tables installed on switches, a controller and a proprietary OpenFlow protocol for the controller to talk securely with switches. Flow tables are set up on switches. Controllers talk to the switches via the OpenFlow protocol and impose policies on flows. The controller could set up paths through the network optimized for specific characteristics, such as speed, fewest number of hops or reduced latency.

Vendors offer varying degrees of user programmability on their routers and switches. This can lead to limited functionality for traffic engineering and management, or inconsistent traffic management between equipment from multiple vendors. OpenFlow is designed to provide consistency in traffic management and engineering by making this control function independent of the hardware it's intended to control.

**Conclusion:** Software Defined Networking (SDN) has emerged as the industry's response to meeting these challenges. SDN allows networks to react dynamically to changes in usage patterns and availability of network resources. Network architectures can be instantly adjusted, respond to application and user requests, and services can be introduced far more quickly, easily and at a lower cost. This makes it possible to optimize networks on the fly and quickly respond to changes in network usage without the need for manually reconfiguring existing infrastructure or purchasing new hardware. SDN separates the control of network devices from the data they transport, and the switching software from the actual network hardware.

It also provides an entity, the controller, that has a comprehensive view of the entire network and its status, and with which switches (network resources) and applications (network consumers) can communicate in real-time. The controller makes it possible for networks to interact with applications and efficiently reconfigure themselves at need, allowing them to implement multiple logical network topologies on a single common network fabric. SDN provides separation between the control plane (controller) and data plane (switch) functions of networks using a protocol that modifies forwarding tables in network switches.