School of Computer Engineering
Kalinga Institute of Industrial Technology (KIIT)
Deemed to be University
Bhubaneswar-751024

Course Objectives:-

CO1: understand how network protocols will work.

CO2: design network protocols for communication.

CO3: implement network protocols.

CO4: troubleshoot implementation of Network protocols.

CO5: understand the working of network routing.

Wireshark

Aim –

1. To know that there exist tools that can sniff packets from your laptop. That is why it is important to use HTTPS and VPN for privacy.
2. To know that millions of packets can get captured and you can use filters to find out packets you are really looking for.
3. To know that networks work by wrapping packets over other packets. This is essentially how the TCP/IP stack works.

Steps –

1. Install Wireshark. Please read the entirety of https://www.varonis.com/blog/how-to-use-wireshark for details and a brief tutorial on Wireshark. Do no proceed forward without first reading the blog properly. Otherwise, you will get confused if you do not have prior working experience with Wireshark. It is a simple tool, however.
2. Start capturing packets on your Wi-Fi interface or Ethernet (if you are using wired LAN).
3. Visit a few websites from your browser.
4. Stop the capture.
5. Notice that thousands of packets are captured. List out various application-level protocol used in those packets. Make sure you notice at least one HTTP and DNS packet.
6. Use display filters to only see the packets you are interested in. For example, a display filter of “dns” would only show you DNS packets.
7. Discard the captured packets and start fresh.
8. This time use capture filters to capture only specific packets. For example, a capture filter of “port 53” will only capture DNS traffic. This is different from display filters. Here you are filtering at the time of capture itself.
9. Start capture.
10. Visit some different websites (other than the ones you visited in the first run).
11. Stop capture and notice that only DNS packets have been captured.

12. Click on any packet and notice how, in the middle area, encapsulation and decapsulation work. You should be able to see the raw frame, the Ethernet part from it, the IP part from the Ethernet part, the UDP part from the IP part, and the DNS part from the UDP part.

Submission –

1. Screenshot of your Wireshark application.
2. Display filter output (screenshot the entire window). Use “dns” as filter.
3. Capture filter output (screenshot the entire window). Use “port 53” as filter.
4. Encapsulation details screenshots. Show frame, Ethernet, IP, UDP, and DNS parts.

Faculty Verification –

1. Students should be able to demonstrate that they can operate Wireshark without much help.
2. Students should be able to understand encapsulation and decapsulation properly and should be able to demonstrate how it works in Wireshark.
3. Students should be able to understand the difference between capture and display filters. They should also identify basic filters.

HTTP

Aim –

1. To understand that HTTP is an application-level protocol that can be used for getting, storing, modifying, and deleting resources. Those resources are usually web pages, images, etc.
2. To know HTTP request and response header formats.

Steps –

1. Start capture in Wireshark and the open a website. Stop capture in Wireshark.
2. This should load the website and load any images it shows.
3. Inspect the HTTP request and response. The GET method should be focused on.
4. Subsequent requests for images should be all separate GET requests and this should be observed.
5. Now start capture again and this time open a website that has a form you can submit data from. This should generate a POST request. Analyze the HTTP request and response in this case.
6. In both the GET and POST cases, various header constituents of both requests and responses should be analyzed using Wireshark.

Submission –

1. Screenshots of GET request and response packets.
2. Screenshots of POST request and response packets.
3. Notes on significance of common request and response headers.

Faculty Verification –

1. Students are now proficient with basic usage of Wireshark.
2. They can demonstrate, in Wireshark, how to locate HTTP messages and header information in requests and responses.

DNS

Aim-

To understand the purpose of different kinds of DNS records like A, AAAA, CNAME, MX, NS, and SOA.

Steps-

1. Install nslookup (works on both WSL-based Ubuntu and Ubuntu VM on Virtualbox) by issuing apt install dnsutils
2. Try the following commands:
   nslookup google.com
   The above will give you the A and AAAA addresses of one of Google servers.

   nslookup ip-address-of-google-from-above
   The above will do a reverse DNS lookup for the IP of Google you obtained from the first command.

   nslookup -type=any google.com
   This will show all available records for google.com

   nslookup -type=soa google.com
   This will show the SOA record for google.com. You can do similar things for a, aaaa, ns, mx, and cname records separately.

Submission-

1. Outputs of the above commands.
2. Notes on significance of the different types of DNS records.

Faculty Verification-

1. Ensure that the students understand the purpose of the nslookup command.
2. They should clearly understand the need for so many DNS records.

Lab No.	Coverage	Sample Experiments

1.	1. Review of pointer, structure, structure with in a structure, and pointer to structure concept using C programming Language.
	
	2.What is little endian and big endian. Discuss the significance of endianness in computer network.
		1. Write a C program to swap the content of 2 variables using pointer.
		
		2. Write a C program to assign values to each members of the following structure. Pass the populated structure to a function Using call-by address and print the value of each member of the structure with in that function.
		struct info{
		int roll_no;
		char name[50];
		float CGPA;
		}
		
		3. Write a C program to extract each byte from a given number and store them in separate character variables and print the content of those variables.
		
		4. Write a C Program to enter a number and store the number across the following structure and print the content of each member of the structure. Then aggregate each member of the structure to form the original number and print the same.
		struct pkt{
		char ch1;
		char ch2[2];
		char ch3;
		};
		
		5. Write a C program to check whether the Host machine is in Little Endian or Big Endian. Enter a number, print the content of each byte location and Convert the Endianness of the same i.e. Little to Big Endian and vice-versa.

2. AND 3.	1. Socket Programming Basics. CN-LAB/socket-programming-slides.pdf For video: CN-LAB/socket-programming.mp4 2. Details of Connection Oriented Socket programming APIs for TCP/IP stack using C.	1. Provide the client side and server side program for the connection oriented socket using C to the students keeping blanks in between. These blanks are mostly the API to be called with their respective parameters which is to be filled up by the students.
4.	1. Demonstrate the packet Analyzer tool i.e. wireshark.	1. Run both client and server (connection oriented) with in the same host.

	(https://www.youtube.com/watch?v=TkCSr30UojM)	2. Run the client in one host and server in another host. 3. Once the client and server runs fine then capture the packets exchanged between them and analyze them through wireshark.
5.	1. Details of Connection less Socket programming APIs for TCP/IP stack using C.	1. Write the client side and server side program with command line arguments for connection less socket using C where both of them will exchange messages with each other. If any of them will receive the "exit" message from the other end then both of them will close the connection. · Run the client and server with in the same host. · Run the client in one host and the server in another host. · Finally capture and analyze the packets exchanged between them.
6.	1. What is I/O multiplexing and when it is required? Different types of I/O multiplexing. 2. Details of I/O multiplexing using select() API. (IO-multiplexing.ppt)	1. Write the client side and server side of a connection oriented socket where the server will behave as a chart server serving multiple chart clients. When the chart server receives a "logout" message from a particular client then it terminate the respective connection with that client.
7.	1. Simple network design. 2. Router configuration.	1. Design a simple network, assign ip addresses and test the connectivity using Cisco packet tracer simulator tool. (https://www.youtube.com/watch?v=I1_zCdLm2YQ) 2. Design more than one LAN and connect them using routers and test the connectivity. Use Cisco packet tracer simulator tool. (https://www.youtube.com/watch?v=CiX30_JVyYQ)
8.	1. Wireless network	1. Design a wireless LAN and test the connectivity. (https://www.youtube.com/watch?v=Jp0hhYpNSYY) (Different mixed wire and wireless LAN problems can be given for practice.)
9.	1. DHCP, SMTP, DNS server configuration.	1. Design a LAN and attach one DHCP, SMTP and DNS server to it. Configure them and test the functionality. Use Cisco packet tracer simulator tool. DHCP- (https://www.youtube.com/watchv=Oj3nFRphDgw) SMTP- (https://www.youtube.com/watch?v=XPake9gZ0hM) DNS-

		(https://www.youtube.com/watch?v=JA8t_IEXcHc)
10.	END EXAM.	