INTRODUCTION TO NETWORK DESIGN AND ANALYSIS CS206

Table of Contents

Topic 1: Course Introduction	9
Topic 2: What is a Network?	10
Topic 3: Traditional Computer Networks	12
Topic 4: Using the Network by Accident	14
Topic 5: Using the Network on Purpose	15
Topic 6: Well-known Network Applications	16
Topic 7: Building a Network: Starts with a Plan	18
Topic 8: Examples of Good Rules for Networking	20
Topic 9: Proprietary and Public Models	21
Topic 10: How TCP/IP Standard Grows?	23
Topic 11: Two-Well Known Networking Models	25
Topic 12: What is a Local Area Network?	26
Topic 13: LAN Cables and Connectors	28
Topic 14: Ethernet Hubs	32
Topic 15: Ethernet Frames and Collisions	33
Topic 16: How to avoid Collisions over Ethernet?	34
Topic 17: Importance of Ethernet Addresses	36
Topic 18: Frame Check Sequence	37
Topic 19: Two Ethernet Standards	38
Topic 20: Working of a LAN Switch	39
Topic 21: Switch: Collision Avoidance	41
Topic 22: Full Duplex and Full Switching	43
Topic 23: Learning Ethernet addresses	45
Topic 24: Ethernet Network Speeds	47
Topic 25: Multiple Physical LANs	49

Topic 26: Introduction to Virtual LANs	51
Topic 27: Packing VLAN's frames in a Trunk	52
Topic 28: Email: A Network Application	54
Topic 29: TCP/IP Email Standards: SMTP, POP3	56
Topic 30: File Transfer Protocol (FTP)	58
Topic 31: World Wide Web and HTTP Protocol	61
Topic 32: Main Features of TCP	63
Topic 33: Working of TCP	65
Topic 34: TCP Flow and Congestion Control	66
Topic 35: User Datagram Protocol (UDP)	68
Topic 36: TCP Ports Numbers	69
Topic 37: TCP Segmentation	71
Topic 38: Basics of Internet Protocol (IP)	72
Topic 39: Working of IP and IP addresses	74
Topic 40: How to run an IP Network?	75
Topic 41: Classes of IP Networks	77
Topic 42: IP Subnetting	78
Topic 43: Network Address Translation (NAT)	80
Topic 44: Dynamic Host Configuration Protocol (DHCP)	81
Topic 45: Internet Control Message Protocol (ICMP)	83
Topic 46: End-to-End: Processing IP Packet	85
Topic 47: A Link State Routing Algorithm	
Topic 48: Distance Vector Routing Algorithm	
Topic 49: Working of Default Gateway Router	90
Topic 50: Address Resolution Protocol	91
Topic 51: Router's Routing Logic and Table	94
Topic 52: Routing with Subnets	96
Topic 53: Router Hardware Architecture	97

Topic 54: Routing to Nearby Places	
Topic 55: Dynamically Learning Routing Tables	101
Topic 56: How to Pick the Best Route	103
Topic 57: Interior & Exterior Routing Protocols	104
Topic 58: Introduction to Domain Name System	105
Topic 59: Domain Name System (DNS) Servers	106
Topic 60: Working of Domain Name System	
Topic 61: DNS Resource Records	109
Topic 62: Introduction to Wide Area Network	111
Topic 63: Different Aspects of WAN Link	112
Topic 64: A Cross-over Cable versus Leased Circuit	113
Topic 65: Routers and WANS	114
Topic 66: Frame Relay	116
Topic 67: Frame Relay Switching	
Topic 68: Virtual Circuits	
Topic 69: Point-to-Point WANs & Frame Relay	120
Topic 70: Routing over PVCs	121
Topic 71: The Internet: A Large IP Network	122
Topic 72: Using a Phone Line for Data	124
Topic 73: Digital Subscriber Line (DSL)	125
Topic 74: Sending Data without a Phone Line	126
Topic 75: Introduction to AAA Security Model	128
Topic 76: Password Authentication Protocol	130
Topic 77: Virtual Private Network (VPN)	132
Topic 78: Enterprise Network and the Internet	133
Topic 79: Firewalls, Demilitarized Zone, IDS	134
Topic 80: Introduction to Wireshark	138
Topic 81: Five Common Network Problems	142

Topic 82: Next 5 Common Network Problems	143
Topic 83: Next 6 Common Network Problems	144
Topic 84: Next 4 Common Network Problems	145
Topic 85: Next 5 Common Network Problems	145
Topic 86: Next 8 Common Network Problems	146
Topic 87: A Four-Part Analysis Methodology	148
Topic 88: Using a Troubleshooting Checklist	148
Topic 89: Wireshark Lab 1	150
Topic 90: Wireshark Lab 2	151
Topic 91: Wireshark Lab 3	153
Topic 92: Wireshark Lab 4	154
Topic 93: Wireshark Lab 5	155
Topic 94: Wireshark Lab 6	157
Topic 95: Wireshark Lab 7	157
Topic 96: Wireshark Lab 8	159
Topic 97: Wireshark Lab 9	160
Topic 98: Wireshark Lab 10	161
Topic 99: Wireshark Lab 11	163
Topic 100: Wireshark Lab 12	165
Topic 101: Wireshark Lab 13	166
Topic 102: Wireshark Lab 14	168
Topic 103: Capture Options for a Switched Network	170
Topic 104: Wireshark Lab 15	171
Topic 105: Wireshark Lab 16	172
Topic 106: Wireshark Lab 17	173
Topic 107: Verify the Target Host Traffic	173
Topic 108: Wireshark Lab 18	174
Topic 109: Wireshark Lab 19	

Topic 110: Wireshark Lab 20	175
Topic 111: Wireshark Lab 21	176
Topic 112: Do not Focus on Acceptable Delays	177
Topic 113: Watch for the Delays that DO Matter	178
Topic 114: Wireshark Lab 22	179
Topic 115: Wireshark Lab 23	180
Topic 116: Wireshark Lab 24	181
Topic 117: Wireshark Lab 25	
Topic 118: Wireshark Lab 26	
Topic 119: Wireshark Lab 27	185
Topic 120: Wireshark Lab 28	186
Topic 121: Wireshark Lab 29	
Topic 122: Wireshark Lab 30	
Topic 123: Wireshark Lab 31	191
Topic 124: RTT: Packets 2 and 3 of TCP Handshake	192
Topic 125: Wireshark Lab 32	193
Topic 126: Wireshark Lab 33	194
Topic 127: Wireshark Lab 34	196
Topic 128: Wireshark Lab 35	198
Topic 129: Wireshark Lab 36	199
Topic 130: Wireshark Lab 37	200
Topic 131: Wireshark Lab 38	201
Topic 132: Wireshark Lab 39	202
Topic 133: Wireshark Lab 40	203
Topic 134: Wireshark Lab 41	205
Topic 135: Wireshark Lab 42	207
Topic 136: Wireshark Lab 43	208
Topic 137: Wireshark Lab 44	

Topic 138: Wireshark's Expert Infos System	210
Topic 139: Wireshark's Packet Loss Detection	211
Topic 140: Packet Loss Recovery Methods	212
Topic 141: Wireshark Lab 45	213
Topic 142: Wireshark Lab 46	214
Topic 143: Wireshark Lab 47	215
Topic 144: Wireshark Lab 48	217
Topic 145: Wireshark Lab 49	218
Topic 146: Duplicate ACKs and their Causes	219
Topic 147: Wireshark Lab 50	220
Topic 148: Wireshark Lab 51	221
Topic 149: Wireshark Lab 52	223
Topic 150: Out-of-Order Packets and their Causes	224
Topic 151: Wireshark Lab 53	225
Topic 152: Wireshark Lab 54	
Topic 153: Causes of Fast Retransmissions	228
Topic 154: Wireshark Lab 55	228
Topic 155: Wireshark Lab 56	229
Topic 156: Causes of Retransmissions	230
Topic 157: Wireshark Lab 57	231
Topic 158: Wireshark Lab 58	232
Topic 159: Causes of ACKed Unseen Segments	234
Topic 160: Wireshark Lab 59	235
Topic 161: Wireshark Lab 60	236
Topic 162: Causes of Keep Alives	237
Topic 163: Wireshark Lab 61	237
Topic 164: Wireshark Lab 62	238
Topic 165: Wireshark Lab 63	

Topic 166: Causes of Reused Ports	241
Topic 167: Wireshark Lab 74	242
Topic 168: Wireshark Lab 75	243
Topic 169: Causes of Checksum Errors	244
Topic 170: Wireshark Lab 76	245
Topic 171: Wireshark Lab 77	246
Topic 172: Wireshark Lab 78	248
Topic 173: Introduction to Packet Tracer	249
Topic 174: Packet Tracer's Interface Overview	250
Topic 175: Creating a Simple topology	254
Topic 176: Introduction to Cisco and PT devices	256
Topic 177: Customizing Devices with Modules	258
Topic 178: Accessing CLI of a Device	259
Topic 179: Configuring Devices with Config Tab	261
Topic 180: Generic IP End Devices in PT	264
Topic 181: Configuring End Devices in PT	266
Topic 182: Packet Tracer's Simulation Mode	268
Topic 183: Connecting Devices and Link Status	271
Topic 184: Testing Connectivity with PDUs	272
Topic 185: Clustering a Topology	273
Topic 186: Creating Cities, Offices & Wiring Closets	275
Topic 187: Managing Cables and Distances	277
Topic 188: Static Routing with GUI	278
Topic 189: Static Routing with CLI	280
Topic 190: Configuring RIP with GUI	282
Topic 191: Configuring RIP with CLI	284
Topic 192: Load Sharing	286

Topic 1: Course Introduction

This topic provides motivation to study this course.

Motivation1: Information I can receive: Now-a-days, we can get a lot of information while sitting in front of the computer. For example, we can access our bank accounts; we can receive world news, look at the weather forecast etc. This has become possible because of computer networking. This is shown next.



Motivation 2: Daily life Applications: Email, web, instant messaging, P2P file sharing, network games, streaming videos, real-time video conferring are those applications that we use in our daily lives. We may encounter issues such as response delays, slow application at the server.

Motivation 3: Internet: The largest engineered system ever created by mankind is the Internet which consists of hundreds of millions of connected computers, communication links, and switches. There billions of users who connect via laptops, tablets, and smartphones.

In this course, we will look at how the computer networks work, how we can troubleshoot their problems using Wireshark, and how we can simulate new networks using Packet Tracer.

Course Composition: We have divided the course intro three Main Parts.

- Part 1 presents the networking fundamentals from practical point-of-view and does not delve into the theoretical nitty-gritties.
- Part 2 describes how analyzing packets with the Wireshark tool can help a student to troubleshoot his network.

• Part 3 discusses how to configure Cisco devices with practical examples and simulate networking with Packet Tracer.

Figures and Material used for Part 1(Topics 2 to 79 have been adapted from W. Odom's book with the title "*Computer Networking First-Step*", Cisco Press, 2004 unless stated otherwise.

Topic 2: What is a Network?

This topic provides the basic know how of a simple computer network.

A **computer network** is a combination of hardware, software, and cabling. This combination allows multiple computing devices to communicate with each other and thus gives computers the ability to communicate with each other. Next we show a simple network:



Figure 2.1: Simple Single Site Network

This is a typical small network used by a company at a single site. In a network diagram, we use a cloud to hide details such as hardware, software, and cabling. This is done, when details are not important to the current discussion. A server provides



Figure 2.2: A closer Look at the simple Network

We can see that User1's PC has networking software installed, and a cable that connects his PC to a socket on the wall. That socket has a cable on the (hidden) other side of the wall plate. Cables normally run under the floor, in the ceiling, or some other hidden place, with the other end being in a wiring closet. All the cables connect to a switch inside the wiring closet. A **switch** consists of specialized hardware and software that forwards the network traffic back and forth between the various network devices on the network. There are a lots of places (called switch ports) into which you can plug in one of the networking cables.

Now, we look at the different perceptions three different people have according to their work on a different aspect of a network. First of all, we study the **Server Guy**, who is responsible for the server and needs some PC hardware skills, but more importantly, he needs strong skills with software. His perception is shown next.

The server guy views the rest of the network as a network utility i.e. he treats the rest of the network just like you think of the telephone, electrical power, and water system. There is no need to think about it unless it's not working. Next is the **Cabling Guy.** He is also known as an electrician. He runs the cables from each cubicle back to the wiring closet. His job requires physical dexterity, knowledge of how to conform to the electrical building standards. He focuses on installing, testing, and troubleshooting the cabling from each wall plate to the wiring closet. He makes sure that there's a working cable running from the wiring closet to each place in the building where a computer needs to connect to the network.



Figure 2.3: The cabling Guy's perception of the network "Elephant"

The third guy is the **Network Guy.** He is responsible for the switch, as well as any other hardware and software used to create a network utility for the computers. He installs, supports, and troubleshoots the hardware and software on the switch. He knows which computer's cable plugs into the various numbered ports on the switch. The network guy's perception is shown next.



Figure 2.4: The network Guy's perception of the Network "Elephant"

Topic 3: Traditional Computer Networks

In this topic, we study different traditional computer networks.

There are two well-known examples of computer networks. They are named an **Enterprise WAN** and the **Internet.**

An Enterprise WAN is a network that should have the ability to connect all sites of a big company, located at multiple sites. You call it an enterprise network because the network is owned by and created by one company, and a company can be considered to be an enterprise.

An enterprise network is shown next.



Figure 3.1: A Larger Network: AN Enterprise WAN

This typical enterprise has three remote branch offices, each with few PCs and a printer. The headquarters site has more users (hence, more PCs), as well as several servers. Each remote site consists of a simple network.

The **Internet** is a unique computer network as it connects almost all enterprise networks and individual users. We next show a part of the Internet.





The Internet works with the help of companies called Internet service providers (ISPs), which provide service to companies and individuals to connect them to the Internet. This makes almost all computers on the planet can communicate with each other.

Topic 4: Using the Network by Accident

This topic discusses the scenario when we use the network by accident.

Some of the most common network services hide the network from the end user. We call them "accidental" network services. To understand this concept, let's assume an employee named Fred working at a remote office. He is the master of his own domain; he has a PC, a printer, and he can do all his work without much interruption from the home office. Fred's daily tasks include opening a document with a word processor, changing some of the text contained in it, and printing it. He then mails the letter to the customer and saves the changes to the letter. Fred's PC is shown next.



Fred's PC

Figure 4.1: Daily task computer

As Fred's company grows, Home office sends Wilma to Fred's office to work with him and help him get all the work done. She brings her fast high-end PC with her to the new office and installs a network. She takes over Fred's old printer and connects it to her own PC. Also copies all the customer letters from Fred's PC over to her PC. The network looks like this:



Figure 4.2: Network with Fred and Wilma's computers

As Wilma has set everything up, including the network, so she knows how it works. But Fred has no idea about the network.

To do his job on the very next day, Fred starts up his word processor, grabs a diskette, and walks over to Wilma's computer and copies a customer letter onto the diskette. He walks back to his desk, updates the letter using his old computer, and then walks back to Wilma's computer with the diskette. Then Fred copies the file

back onto Wilma's computer, replacing the old file, and brings up the word processor on Wilma's computer so that he can print a copy for mailing to the customer. This is depicted next.



Figure 4.2: Fred's sneakernet

Fred could have done the same job in a simpler and more efficient way if he had used the network. He doesn't even need to know it's there. If Fred had just looked on his C drive in the folder called Customers, he would have seen all the same files he is used to working with. These files are on Wilma's PC - a <u>file server</u> but that's hidden from Fred. Also, when he clicks on the Print icon from his word processor, he'll see a printer called *same old printer*. If he prints to that printer, it will print on his same old printer, even though it's connected to Wilma's PC, which is set up as a <u>print server</u>.



Figure 4.3: Fred's PC using the new Network

Topic 5: Using the Network on Purpose

In topic, we study the case when we use the network on purpose.

When we use the Internet, chances are high that we will be aware that we are using a network. The Internet is the global network to which almost every company and organization in the world is connected. We can make a phone call to almost anyone on the planet because all the telephone companies in the world connect to each other. Similarly, most computers can communicate with each other over the Internet because most computer networks connect to each other through Internet Service Providers (ISPs). Networking connections among enterprises, individual home users, and other ISPs are provided by ISPs. A chunk of the Internet is shown next.



Figure 5.1: Conceptual View of the Internet

ISP1's network allows individual computers, such as Fred's, to connect to it. Fred could communicate with computers inside Company A, assuming that the security policies at Company A allowed it. Company A created its enterprise network using hardware, software, and cabling paid for by Company A. ISP1 creates its network as well using its own funds. ISP1 agrees to allow Company A's traffic to pass through ISP1 and on to other ISPs so that Company A can communicate with the rest of the world. In return, Company A pays ISP1 an ongoing fee.

Topic 6: Well-known Network Applications

This topic discusses working of the well-known network applications.

Web Browsing

A **web browser** allows you to sit at one computer and display information that resides in a **web server** somewhere on the Internet. A **web server** consists of software that resides on the computer that is accessible to the end user via the Internet. The information can be in many forms, including simple text, graphics images, animation, video streams, and audio clips. Well-known examples of web browsers are Internet Explorer, Google Chrome, Firefox Mozilla. After a browser requests a web page from a web server, the server replies by sending the contents of the web page back to the browser. Companies, organizations, and even individuals can create their own websites on the Internet. Anyone can then ask for the content in those web pages and see the results. Next, we show the working of web browsing application.



Figure 6.1: Web Browsers and Web Pages on the Internet

A client such as Fred brings up the web browser software on his computer and points to the Universal Resource Locator (URL) of the website. For example, in this case he wants to access www.fredsco.com. The URL or a web address is a string of characters that uniquely identifies a particular web page.

E-Mail

It allows a user to create, send, and receive messages electronically. E-mail is sent between one user and another user, or in some cases, to multiple other users. The person sending the mail needs to know the e-mail address of the person who needs to receive the e-mail. An e-mail address is a text string that represents the address of a person for the purposes of sending and receiving e-mail, much like a mailing address used for postal mail. Example: abc@yahoo.com. An example is shown below.



Figure 6.2: E-Mail in the Internet Using Mail Servers

In this diagram, Barney wants to e-mail the sales department at Fred's Company by sending an e-mail to the address of <u>sales@fredsco.com</u>. Barney does not actually

send the e-mail directly to a computer at Fred's Company. Barney sends the e-mail to his e-mail server, which for an individual Internet user like Barney, typically sits inside the ISP network to which the user is connected. An **e-mail server** is a server that receives, forwards, or holds e-mail, much like the service performed by the postal service. That e-mail server forwards the e-mail to the e-mail server at Fred's Company.

Downloading and Transferring Files

The central theme of this application is to move files into and out of your computer using a network. In most cases, the end user does not have the file, and he wants to get it. Next, we show an example where **myproject.doc** is being copied between Barney and Fred.



Figure 6.3: Copying myproject.doc Between Barney and Fred

File transfer is a two-step procedure:

- 1. Fred places a copy of the file onto a file transfer server, using file transfer client software.
- 2. Barney gets a copy of the file from the file transfer server, also using file transfer client software.

Topic 7: Building a Network: Starts with a Plan

In topic, we study why rules and standards are important for networking.

Standards are rules that make life a lot easier. To understand this idea, let's have a look at a power socket on a nearby wall. You may find an electrical socket with three holes two that accept flat metal prongs, and one that accepts a round metal prong. The flat metal prongs are parallel to each other. If you go to a store to buy an electrical device such as a lamp, you will expect that its power cord would fit into the

wall socket of your house. In case, does not fit into the power socket, you will get very confused.

Let's now imagine that you bought a new lamp. You plug it in, and the light bulb instantly is broken. You put in a new bulb, and it doesn't light up. You decide that there is some problem with the lamp. So you bring it back to the store and replace it with an identical lamp. When you get it home, the same thing happens to this lamp too. What to do next? Let's call the customer service to inquire from about the problem. This is our special 'We light up your life' model of lamp. It uses the same kind of power cord you are used to using but requires less electrical voltage, saving you money. If you read the instructions for the lamp, you will see that it directs you to rewire and change the voltage coming out of the sockets you want to use for the lamp. 'If you plug this lamp into a normal wall socket, the extra voltage will fry the lamp, and it will no longer light up your life.' So contact an electrician and get your socket changed.

A standard therefore tends to define a particular thing, such as the shape of the wall socket and connector used by an electrical socket and electrical power cable. Another standard might dictate how much voltage flows through the wall socket, whether it is AC or DC, how much current, and the like. Both standards are important and must be followed to prevent exploding lamps. Similarly, networking has standards. First, the PCs in the network need some form of physical connectivity. Physical connectivity refers to the combination of cabling, networking devices, and network interface cards (NICs) in the computers, which together provide the physical capability to transmit and receive data across a network. Next, we show this idea.



Figure 7.1: Components of a Simple Network

Both PCs have an NIC installed, and each card has a receptacle into which a cable can be connected. To emphasize their existence, the NICs are shown outside the PCs. The two computers must implement the same standards for how the networking software on each computer tells the other what it wants to do.

Topic 8: Examples of Good Rules for Networking

This topic presents examples for implementing networking.

Example 1

When two humans communicate one person says something and the other person listens. The two people need to understand the same language. Similarly, when computers communicate with each other over a network, the sending application needs to send some information to an application on the receiving side. Let's assume Fred's computer is using Wilma's computer, which is configured as a file server.



Figure 7.2: Data Transmission Using Packets

Over the network, the file will be transmitted in the form of binary digits (bits). The file server reads the file from the disk. A networking standard will define the encoding rules so that data can be transmitted. For example, it can ask the sender to change the voltage to one level to mean a binary 0, and another to mean a binary 1. In our example shown above, the NIC inside Wilma's computer sends some electrical signal over the cable. The device on the other end of the cable Fred's PC NIC interprets the incoming electrical signal. This will work if both NICs agree to some standard means of transmission. For instance, imagine that Wilma's NIC sends a +5 volts to transmit a binary 0 and +10 volts to send a binary 1. If Fred's NIC expects to receive a +2 volt signal for binary 0 and +4 volts for binary 1, the network will not work because Fred will not understand what Wilma is sending him.

Example 2

Devices in a network send bits in groups, generally called packets. Assume Wilma wants to send a file to Fred. She sends the data bits in the form of three packets as shown next.



Figure 7.3: A Simple protocol for Error Recovery

We assume that Fred received all the three packets. The second packet has some bits that he is unable to understand. This may have occurred due to electrical interference (one electrical signal changes because of some other nearby electrical signal). If both NICs have agreed upon +5 volts to be a binary 0, and +10 volts to be a 1, then the receiving NIC will get confused upon receiving let's say +7.5 volts. A standard needs to be defined to address this issue. Both Wilma and Fred's computer software must agree to use the same networking protocol that provides a method to recognize and recover from errors. In the diagram below, we have shown such as error recovery mechanism:

- 1- The sender of the packet numbers the packets.
- 2- When the receiver (Fred) notices the error, he can send a message back to the sender (Wilma) asking her to send the specific packet again.
- 3- Wilma replies to Fred's request by resending packet 2.

Fred's File Old Computer		Wilma's Super- Dooper Computer
	← #1 First Part of File	
	$\leftarrow \qquad \qquad$	
	Third Part of File]=
	Please Resend #2	$] \longrightarrow$
	← #2 Second Part Again]=

Figure 7.4: Error Recovery

Topic 9: Proprietary and Public Models

In this topic, we describe Proprietary and Public Models.

Networks began to be developed as part of each computer vendor's offerings by the late 1960s, and they became popular by the late 1970s. Each computer vendor created its own networking model, which helped computers from that one vendor communicate easily. At the advent of networking, the two largest computer vendors in the world were: International Business Machines (IBM) and Digital Equipment Corporation (DEC). IBM created its own networking model called Systems Network

Architecture (SNA), and DEC created its own DECnet. Each of these vendorproprietary networking models allowed networks to be created and implemented and they worked but they were issues. Proprietary networks cannot communicate with other networks. IBM computers could not communicate with DEC computers.

Let's assume a company that owns some IBM computers and some DEC computers. How the company can form a network? The company should have two separate networks?



Figure 9.1: Non-Networking of IBM and DEC Networks in a Single company

Two solutions emerged: one short term, and one long term.

Short term solution

DEC made its computers conform to the IBM SNA model as IBM was roughly 10 times larger at the time in terms of gross revenues. So, DEC created software that converted between DECnet standards and SNA standards. DEC created a DEC-to-SNA gateway, which allowed the DEC and IBM SNA devices to talk.



Figure 9.2: DECnet Emulating SNA Using a Gateway

Long term solution

The second, better, and long-term solution was to get IBM and DEC to stop using their proprietary networking models and use a public, open networking model. Due to this, all computers can communicate easily. Public network models provide pervasively popular networks. Today, we live in a world in which practically every computer uses the same public network model, called Transmission Control Protocol/Internet Protocol, or TCP/IP. No one vendor dictates the standards and protocols for it. Individuals from many companies and organizations have participated in the standards definition process. The Internet Engineering Task Force (IETF) manages the process of creating TCP/IP standards. IETF is open to any interested individual and its URL is http://www.ietf.org.

Topic 10: How TCP/IP Standard Grows?

This topic describes how TCP/IP standard grows.

The Internet Engineering Task Force (IETF) manages the creation and approval of TCP/IP standards and protocols. Each standard or protocol of the TCP/IP model is defined in a document. It is posted on the Internet so that anyone can look at it and comment before it becomes a Requests for Comments (RFC). TCP/IP is composed of a lot of individual protocols. The name TCP/IP is a combination of the two most popular protocols inside the TCP/IP model.

Transmission Control protocol (TCP)

TCP is described in RFC 793. TCP appends a TCP header in front of the user data. A header is a bunch of overhead bits added to the user data so that a protocol can do its job. Acknowledgment number is a field in the TCP header. TCP uses it for error recovery. On the receiver, TCP uses it to inform the sender which packet he expects to receive next and can thus be used to inform about erroneous packets.



Figure 10.1: TCP Error Recovery

The Internet Protocol (IP)

IP is another TCP/IP standard protocol. It defines logical addressing and routing for the TCP/IP model and is explained in RFC 791. The best analogy for understanding IP's working relates to the postal service. Before you put a letter in the mailbox, you put an address on the front. Similarly, a computer sends a packet over the physical network by putting an address in front of the packet. In addition to TCP header, IP adds its own header. IP address is used for identifying a computer. Besides names, NTN/CNIC numbers are identifiers for different purposes. www.fredsco.com is a string for a computer which humans can remember easily. A number such as 3.3.3.3 is computer-friendly. Computers are identified by IP addresses. When packets are received by a router, the router looks at the IP address and decides where to forward the packet. A router uses a table, called a routing table. This is shown next.



There are other standards bodies who have already defined standards that TCP/IP can easily use. IETF happily references other well-known standards created by other standards organizations. For example, The Institute of Electrical and Electronic Engineers (IEEE) defines standards for LANs, which defines a type of network, or a part of a larger network, in which the devices are relatively close together. By "close", we mean in the same building or in the same small campus of buildings. Like any other network, a LAN includes computers, hardware, software, and cabling that allow communications to happen. TCP/IP standards simply say "Use IEEE LAN standards if you want to use a LAN." Another example is the International Telecommunications Union (ITU), which defines standards for WANs. A wide-area network (WAN) defines a type of network, or part of a network, in which the devices are relatively far apart. A WAN is a network, or part of a network, for which the cabling must pass outside the property of one company. The distance might only be a few miles, or it might be thousands of miles!



Topic 11: Two-Well Known Networking Models

In this topic, we study TCP/IP and OSI networking models.

Systems can be implemented more efficiently and more effectively if the activities are broken down into layers. Layering helps in product development, and it helps in keeping each protocol simple. The reduced complexity makes for better products and more stable networks.

The TCP/IP Style

TCP/IP really is a set of protocols and standards, some not even defined directly by TCP/IP, that allow you to create networks. Some people segment the TCP/IP network model into the four layers. Others consider it to be a five-layer model. From a practical perspective, it does not matter.



Figure 11.1: TCP/IP Model

Each layer represents a general function that must be accomplished.

The OSI Style

While TCP/IP was being evolved into a legitimate networking model, a competing public networking model, called Open Systems Interconnect (OSI), was being

developed. The OSI model was developed by an organization called the International Organization for Standardization (ISO). The aim was to build the end-all networking model i.e. once implemented by all computers on the planet; it would allow pervasive communications among all computers from all vendors in all countries everywhere! OSI uses a seven-layer model, instead of the four (or five) layers in the TCP/IP model.

	OSI Model	TCP/IP Model	TCP/IP Protocols
7	Application		
6	Presentation	Application	HTTP, SMTP
5	Session		rurs
4	Transport	Transport	TCP, UDP
3	Network	Internetwork	IP
2	Data Link	Network	Ethernet,
1	Physical	Interface	Frame Relay, PPP

Figure 11.2: OSI & TCP/IP Models

TCP/IP versus OSI:

OSI was developed much more slowly than TCP/IP. TCP/IP took over the marketplace before OSI could be finished. ISO plays an active role in standards development even today, working with the IETF, ITU, and other standards bodies.

Topic 12: What is a Local Area Network?

This topic describes the basics of local area network.

A LAN is a network in which the devices are relatively close together i.e. in the same building or in the same small campus of buildings. The simplest LAN consists of at least 2 computers, networking software, a cable and network interface cards (NICs). It is shown next.



Figure 12.1: Components of a simple Network

Assume Fred opens a file on his computer that resides on Wilma's disk drive. Then, prints the file on the printer connected to Wilma's computer, and finally, saves the file back on Wilma's disk drive. This sequence is shown



Figure 12.2: Basic Flow with Fred Using a file/ print server

A file is just a bunch of bits. In the previous example, the network needs to be able to transfer a bunch of bits from one computer to another. To send a binary code from one device to another, the sending device puts some electricity on the wire. An NIC can vary the voltage level of an electrical signal to different values. One value represents a binary 1 while the other means binary 0. Assume that a company employs an encoding standard that makes the NICs to define a binary 0 with a voltage of ±5 volts, and a binary 1 with ±10 volts. This is shown next.



Figure 12.3: Basics Data of Transmission across a wire

PC2 wants to send a binary value 0101. It generates some electricity on the wire according to the encoding scheme. PC1, on the other end of the wire, senses the incoming electrical signal. It interprets the electricity, using the same set of encoding rules to mean 0101, exactly as PC2 intended. **Digital Transmission** is the use of discrete, constant values, which are then instantly changed to other possible discrete values. When both sender and receiver agree to encoding scheme, and the

rate at which the bits are transmitted over the wire. The receiver (PC1) must think about the electrical signal at different points in time, on a regular interval. The sender (PC2) must use the same regular time interval to decide when it should change signal.

An Example

Assume PC2 varies the voltage to mean either 0 or 1 every .1 seconds, and PC1 samples the incoming electrical signal every .1 seconds. The speed of this connection is 10 bits per second (bps). Digital Transmission does not work correctly if the two PCs do not agree on the transmission speed. For example, if we assume that PC2 thought the speed was 10 bits per second, meaning it should encode a new bit every 1/10 of a second. If PC1 thought that it should be receiving a bit 20 times per second, it would sample the incoming electrical signal every 1/20 of a second. PC1 would think it was sending 10 bits each second, and PC2 would think it received 20 bits. In real life, LANs typically run at much higher speeds, with a slow LAN transmitting at 10 million bits per second (Mbps, also called megabits per second). Can PC1 and PC2 talk simultaneously in both directions over the same wire? The electrical signals would overlap, and neither PC1 nor PC2 could understand what was sent. This is shown next.



Figure 12.4: Concurrent Data Transmission across Two Different Wires

Topic 13: LAN Cables and Connectors

In this topic, we study LAN cables and connectors.

Cabling and connectors are used to manage the wires.

Cables

The copper wires that networking cards use are encased inside a cable which is made from plastic. A thin plastic coating painted onto each wire helps prevent it from breaking. Different colors of plastic coating for each wire helps in looking at each end of the cable and figuring out which wire is which.



Figure 13.1: Typical LAN Cable

The electrical currents on the wire can change when electrical signals that exist in the air caused by other wires or other nearby electrically powered devices – Electromagnetic interference (EMI). A computer might misinterpret a 0 as a 1 or a 1 as a 0, or might not understand what the sender really sent. To reduce electrical interference, the wires are twisted together in pairs. Such a pair is called a twisted pair. Shielding can be added to the cable to reduce EMI. The cable becomes less bendable but more expensive. Such a cable is called shielded twisted pair (STP). This is shown next.



Figure 13.2: Shielded Twisted Pair (STP) cabling

Unshielded cables are called unshielded twisted pair (UTP). Mostly, the LAN technology employs less expensive UTP cabling. STP cabling is preferred in environments where significant EMI exists.

Connectors

They line up the wires on the end of a cable. Each colored wire has a specific reserved place inside the connector. A pin is a physical position in the end of the connector in which the copper part of the wire sits.



Figure 13.3: Typical Networking connector (RJ 45)

The Telecommunications Industry Association (TIA) and Electrical Industries Alliance (EIA) define standards for which wires fit into which pins when you make a cable for use with LAN. Eight wires can fit into an RJ-45 connector. EIA/TIA standards suggest numbering schemes for the eight pin locations and the pairs of wires. Two of them are shown next.



Figure 13.4: Pinout options for (RJ 45) Connectors

The standard also specifies which color of wire goes into pin position 1, 2, and so on. The **RJ-45 Color Code** scheme is shown next.



Figure 13.5: Color code scheme of RJ45 Connector

NICs send data over the twisted pair that uses pins 1 and 2 of an RJ-45 connector and receive data on the twisted pair that uses pins 3 and 6. Let's cable one end of a wire in pin 1 of one connector, and the other end into pin1 of the other connector. Pin 2 on one end of the cable connects to pin 2 on the other side and so on. This arrangement is called a straight-through cable. This is shown next.



Figure 13.6: Both PCs Using the Same Pair (Lane) to send data

NICs of both PCs send data at pins 1 and 2. That electricity goes over the wires and enters the other NIC on pins 1 and 2. But, the NICs aren't receiving data on pins 1 and 2! Hence, both PCs send, but neither receives data. If we connect the wire at pin 1 on one end of the cable with pin 3 on the other end; the wire at pin 2 with pin 6 on the other end; the wire at pin 3 with pin 1 at the other side; and the wire at pin 6 with pin 2 at the other side. This arrangement is called cross-over cables. The NICs of the two PCs can receive the data sent by the other device! This is shown next.



Figure 13.7: Both PCs Using Different Pair (Lane) to send data

Topic 14: Ethernet Hubs

This topic explains the basics of Ethernet hubs.

Ethernet is by far the most popular type of LAN today. Ethernet LANs are highly used at work and school. To connect to other PCs, you would need more LAN NICs, and your PC probably does not have enough room for all the cards. Also, you would need to run cables between your PC and all the other PCs. If you tried to do this for 100 PCs on the same floor of the building, and every PC wanted to connect to every other, you would have 99 cables connected to 99 NICs inside each PC!

The alternative to running a cable to every other PC is to run a cable from each PC to a wiring closet and connect the cables to a networking device, called an Ethernet hub. An Ethernet hub allows the electrician to cable each device to the hub using only a single NIC and single cable. It eliminates the cabling problem. This is shown next.



Figure 13.6: Ethernet Hub repeats everything it hears

The hub simply listens for incoming electrical signals, and when received, the hub repeats the same electrical signal to every other device that's connected to the hub. The hub expects straight-though Ethernet cabling between itself and the PCs. This is

opposite of Ethernet's pin configuration. A straight-through cable works between a PC NIC and a hub. This is shown next.



Hubs Logic:

1- Receive traffic on pins 1 and 2 on each physical interface.

2-When received, repeat the same electrical signal out all other ports, except the one in which the data was received.

3-When repeating out other ports, repeat the traffic out pins 3 and 6 so that the PCs will be listening.

Topic 15: Ethernet Frames and Collisions

In this topic, we study Ethernet frames and learn why collisions occur.

Ethernet is world's most popular LAN standard. It consists of a set of standards and protocols for LAN communication, as defined by the IEEE. For instance, Ethernet standards define how a network interface card (NIC) should encode binary 0s and 1s on a wire by varying the voltage. Before a PC could ask an NIC to send data, the PC must encapsulate the user data inside an Ethernet frame. i.e. take the data and add a header and trailer to it. Ethernet standard defines the formats of headers and trailers. The resulting bunch of bits created including the Ethernet header and trailer is called an Ethernet frame. This is shown next.



Figure 15.1: Ethernet Frame sent from Bobs to Larry

The information in header and trailer are used by the computers, NICs, and networking devices to make the Ethernet work smoothly. The first 8 bytes in an Ethernet header are called the preamble. The preamble contains alternating 1s and 0s so that the NICs receiving the data know that a new frame is being sent across the LAN. A collision occurs when two or more frames are sent over a single twisted pair at the same point in time. The result is that none of the frames is intelligible. Even when all devices follow Ethernet rules, collisions can happen. Let's focus on two key facts. **Fact1**: When two or more electrical signals travel over the same pair, both electrical signals are distorted and become a single signal. The receiving device cannot interpret the signal as 0s and 1s. **Fact2**: A hub repeats received electrical signals out on all other physical ports on the hub, except the one over which the signal was received, even if other electrical signals are already being repeated. This is shown next.



Figure 15.2: Collision between Bob's and Archie's Frames

What happens when both Bob and Archie send an Ethernet frame to Larry at the same time? The hub blindly repeats each of the frames sent by Bob and Archie out all other ports, including the one connected to Larry. Larry cannot understand either frame because the hub is trying to send both electrical signals over the cable to Larry at the same time.

Topic 16: How to avoid Collisions over Ethernet?

This topic discusses how to avoid collisions over Ethernet.

Imagine that you are connected to a hub, and you are currently receiving a frame. What would happen if you sent a frame at that time? This will cause a collision. Ethernet standards define a basic algorithm that helps reduce collisions, as well as defining what to do when collisions occur. The algorithm is called the carrier sense multiple access collision detect (CSMA/CD) algorithm

The CSMA/CD Algorithm

It is based on a simple concept: Listen before sending, and wait until you are not receiving a frame before you try to send your frame. Well, that's exactly what the algorithm calls for. Why not just wait a moment? This is shown next.



Figure 16.1: Collision Avoidance by Listening before sending

Even when you are using CSMA/CD, collisions can still occur. Why? Let's imagine Larry Bob and Archie are connected via an Ethernet. Currently Larry is sending a frame. Bob and Archie want to send frames over the LAN. Possible after Larry finishes. This is shown next.



When no one is sending anything, there is no electricity flowing over the wires and we say that LAN is silent. As soon as both Bob and Archie stop receiving an electrical signal, both try to send their frames at roughly the same time. This is shown next.



Figure 16.2: An imminent and Unavoidable Collision

The hub will repeat both frames out to Larry, so Larry will know about the collision. The hub won't forward Bob's frame back to Bob, Bob receives one frame. The same happens with Archie. Bob and Archie do not know there's a collision! When an NIC transmits a frame, it receives its own frame as well. This is called a loopback circuit. By using loopback, while Bob is sending a frame and if Archie also sends a frame at the same time, the hub forwards Archie's frame to Bob and now Bob knows there is a collision.

CSMA/CD: what to Do When a Collision Happens

1. The senders of the collided frames send a jamming signal to make sure everyone knows a collision has occurred.

2. The senders of the collided frames independently pick a random timer value.

3. Each sender waits until his own random timer has expired and then tries to send his frames again.

Topic 17: Importance of Ethernet Addresses

In this topic, we learn the importance of Ethernet Addresses.

In an Ethernet LAN with a hub, a computer receives a lot of Ethernet frames due to the hub logic. Only some of them contain data that is meant for that computer. Consider Larry, Archie and Bob are connected over an Ethernet LAN with a hub. A frame sent by Larry to Bob will be received by Bob and Archie.



Figure 17.1: Collision Avoidance by Listening Before Sending

Archie should simply ignore the frame, and Bob should examine the data and process it. Practically, this becomes possible with the help of Ethernet Addresses. Before Larry sends the frame, he puts Bob's Ethernet address into a field in the Ethernet header called the destination address field. Next, we show working of this concept:



Figure 17.2: How Bobs Decides the frame was sent to Him
Each NIC has a unique Ethernet address, which is put onto the NIC by the NIC manufacturer. Ethernet standards specify that addresses be 48 bits long (6 bytes). The Ethernet header includes both a source address field and a destination address field. The source address identifies the Ethernet address of the NIC that sent the frame. When an NIC receives a frame, it uses the destination address to decide whether to process it or ignore it.



Figure 17.3: Ethernet Source and Destination Addresses

After Larry assembles the frame, he can use CSMA/CD logic to determine when he can send the frame.

Topic 18: Frame Check Sequence

This topic describes the usage of Frame Check Sequence.

Consider Larry, Archie and Bob are connected over an Ethernet LAN with a hub. Larry sends a frame to Bob. This is shown next.



When Bob's NIC interprets the incoming electrical signal as meaning some string of binary 0s and 1s, it might misinterpret the meaning. Electrical signal has changed as it passed over the wire, affected by resistance in the wire, EMI, collisions, and other factors. Sender might send a 1 and the receiver might think it was a 0, or vice versa. Ethernet uses a field in the Ethernet trailer called the frame check sequence (FCS)

field. It holds a 4-byte number that is set by a sender and allows a receiver to determine whether a frame contains bit errors.

A sender runs a mathematical formula, with the input being the contents of the Ethernet frame to be sent, up to the Ethernet trailer. The sender places the results of the formula into the trailer FCS field. On the receiver's side, the receiver applies the same formula to the same part of the Ethernet frame (everything up to the trailer). If the result of the formula as calculated by the receiving NIC is the same as the value in the transmitted FCS, then no errors occurred. Otherwise an error has occurred.



Ethernet standards specify that the receiver should simply discard frames that have errors. The receiver does not request that the frame be re-sent, and the sender does not know that the frame was in error. This simple process is called error detection. Error recovery refers to the process whereby the receiver requests that the sender retransmit the frames that did not pass the FCS check. Transmission Control Protocol (TCP) performs error recovery, but Ethernet does not.

Topic 19: Two Ethernet Standards

This topic describes two different standards of Ethernet.

Ethernet was originally created in the 1970s by Robert Metcalfe and some others working for Xerox corporation. Later Intel got involved, and convinced them to put Ethernet logic on a computer chip, making the mass production of Ethernet cards less expensive. Also Digital Equipment Corporation (DEC), the second-largest computer maker at the time, got involved to support Ethernet. The original standard, as defined by those three companies, came to be called DIX Ethernet. The final version created by these three companies is known as Ethernet Version 2. In the 1980s, the IEEE while standardizing several LAN standards, created a committee to define the Ethernet standards and protocols. The committee was named the 802.3 committee. Another committee was created by the IEEE to define the common LAN features and was known as the 802.2 committee. The 802.3 standard is also called

Media Access Control (MAC), and the 802.2 standard is also called Logical Link Control (LLC). Next we show the header defined by 802.3 and 802.2, as well as the trailer, as defined by 802.3.



Figure 19.1: IEEE Ethernet Headers and Trailers

The preamble with starting delimiter (SD) serves to "wake up" the receiving adapters and to synchronize their clocks to that of the sender's clock. The original DIX Ethernet specifications used an 8-byte preamble field. IEEE renamed the eighth byte of the preamble to starting delimiter, with no change in the value of the field. Ethernet addresses are often called MAC addresses and are 6 bytes in length.

IEEE Ethernet Technologies

The early 10BASE-2 and 10BASE-5 standards specify 10 Mbps Ethernet over two types of coaxial cable, each limited in length to 500 meters. In mid-1990s, Ethernet was standardized at 100 Mbps, which is 10 times faster than 10 Mbps Ethernet. 100BASE-T was defined for twisted pair copper wires and 100BASE-FX was defined over fiber. 100 Mbps Ethernet is limited to a 100 meter distance over twisted pair and to several kilometers over fiber. Gigabit Ethernet offers a raw data rate of 1,000 Mbps. 10 Gbps Ethernet was initially operating over optical fiber, but is now able to run over category UTP cabling.

Topic 20: Working of a LAN Switch

In this topic, we learn how a LAN switch works.

Only one device can transmit data at a time over a LAN connected via a hub. A switch allows multiple devices to transmit at the same time, increasing the amount of data that can be sent over the LAN. Let's assume PCs of Fred, Barney, Betty and Wilma are connected over a LAN via a hub. Fred wants to send a frame to Barney while Betty wants to send a frame to Wilma. **Hub's logic:** When I receive an electrical signal, I repeat it out on all ports, except the port over which the signal was received. **CSMA/CD logic:** A PC needs to listen before sending i.e. if you're currently receiving a frame, wait until it is finished before you try to send your frame. Otherwise a collision can occur.



Figure 20.1: Betty Waiting on Fred's Frame that was sent to Barney

It's like paving a beautiful wide road and only having one lane, so everyone must wait until one car finishes driving past before the next car can use the road.

Switch

A LAN switch provides the same cabling advantage that a hub does, while providing significant performance improvements. **Switch's logic:** When receiving a frame, examine the destination Ethernet address. Forward the frame out on the one port and only that port through which that address can be reached. In comparison to a hub, a switch does not simply repeat the electrical signal out on all other ports. A switch forwards frames selectively only forwarding the frame where it really needs to go.



Figure 20.2: Switch logic

In real life, we can simply replace a hub with a switch, and use the same cables. The switch still uses a twisted pair for transmission and another for receiving traffic in each cable. It uses the same pin outs in the RJ-45 connector. The switch receives on pins 1 and 2 and transmits on pins 3 and 6 just like a hub i.e. you need a straight-through cable between the switch and each computer. The physical details can remain the same. To make a forwarding decision, a switch uses a table that lists the MAC addresses in the network. We can refer to it as either the switching table or the MAC address table.



Figure 20.3: Fred Sending to Barney, While Betty send to Wilma

LAN switching logic improves LAN performance because the frame is not repeated to every computer that is attached to the switch. Betty still uses CSMA/CD logic. With two devices sending at the same time, the capacity of the LAN to forward frames doubles! If we are using a 10 Mbps Ethernet, separate transmissions occur, each at 10 Mbps. This LAN has 20 Mbps of capacity. Now imagine a switch with 24 ports, with the device on port 1 sending to the device on port 2; the device on port 3 sending to the device on port 4; and so on. This switch supports 12 x 10 Mbps, or 120 Mbps, of capacity.

Topic 21: Switch: Collision Avoidance

In this topic, we study how a switch avoids collisions.

Let's assume PCs of Fred, Barney, Betty and Wilma are connected over a LAN via a switch. When a switch receives a frame, it examines the destination Ethernet address. It forwards the frame out on the one port and only that port through which that address can be reached.



Let's discuss the scenario: what happens if everyone wanted to talk (send) to Fred at the same time? Keeping the logic of a switch in mind, the switch would try to

forward all three frames, which would cause a collision. A switch receives on pins 1 and 2 and transmits on pins 3 and 6.



Figure 21.1: Potential collision when forwarding multiple frames

Buffers

To avoid sending all three frames at the same time, the switch uses buffers. Buffers consist of memory inside the switch that is used to store frames temporarily. The switch sends one frame, and it keeps the other two frames in buffers. After the switch finishes sending the first frame, it gets one of the frames from the buffers and sends it. Finally, the switch grabs the third frame from the buffer and sends it. By doing so, the switch usually avoids causing a collision.



Figure 21.3: Switch Avoids Collisions by Buffering the Frames

Switch Logic Updated

When receiving a frame, examine the destination Ethernet address. Forward the frame out the one port and only that port through which that address can be reached. If multiple frames need to be sent out to the same port, send one frame and buffer the rest. As soon as the port becomes available, send the other frames. By buffering frames, delay will be introduced called buffering delay. How much? For instance, if those three frames were each 1250 bytes long, it would only take 3 milliseconds (.003 seconds) for all three to be sent over the cable with a 10 Mbps rate from the switch to Fred. If collisions had occurred, each frame would have taken

longer to reach Fred, and, both the collision and the time taken to resend the frames could have prevented other user traffic from crossing the LAN. Buffering of frames definitely improves LAN performance.

Topic 22: Full Duplex and Full Switching

This topic presents concepts of full duplex and full switching.

Let's assume PCs of Fred, Barney, Betty and Wilma are connected over a LAN via a switch. With carrier sense multiple access/collision detect (CSMA/CD), if an NIC is not receiving a frame, it can send. The switch avoids collisions by buffering the frame if the output port is busy.



Figure 22.1: Potential Collision When Forwarding Multiple Frames

A computer NIC, using CSMA/CD, believes it should not send a frame: when the NIC is actually receiving a frame. Imagine that Fred is sending a frame to Barney. Barney wants to send a frame. (In this case, Barney wants to send a frame to Fred, but the destination does not really matter in this scenario.)



Figure 22.2: Barney Waiting to Send, When Fred Sends to Barney

With Barney's CSMA/CD logic enabled, he must wait before sending a frame. But can a collision really occur? Physically, there is a single cable between Barney and the switch. Barney sends on the pair using pins 1 and 2, and the switch sends to Barney on the pair using pins 3 and 6. The switch will buffer any frames if pins 3 and 6 are busy. There is truly no danger of a collision. Barney would not cause a collision, if he sends the frame. Barney can send and receive at the same instant in time as receiving the frame from Fred. This ability is called full duplex. The CSMA/CD imposes restriction of only sending or only receiving at one point in time. This is called half duplex. This is shown next.



Figure 22.3: Barney and Fred Sending and Receiving at the Same Time Full Duplex

With switches and full duplex enabled, the LAN works like you have a two-lane road between the switch and each device, plus another two-lane road between each port on the switch. This is shown next.



Figure 22.4: Full Duplex and Full Switching

Topic 23: Learning Ethernet addresses

In this topic, we study how Ethernet addresses are learnt.

A switch maintains a switching table or Media Access Control (MAC) address table. It makes its decisions about where to forward the frames based on that table. How does the switch figure out what should be in the MAC address table in the first place? The switch has no entries in the MAC address table when the switch first powers up. A switch dynamically learns MAC addresses and the corresponding ports. This process of learning MAC addresses is called learning.

Whenever an NIC sends an Ethernet frame, the NIC places its own MAC address into the frame as the source MAC address. The switch learns the MAC address of the sender of each frame by examining the source MAC address.



When a switch receives a frame, it sends the frame out on one port, based on a comparison of the destination MAC address and the MAC address table – forwarding decision. Conversely, by not sending the frame out on other ports, the switch has simply filtered the frame from exiting those ports, which is a filtering decision. What should a switch do with a frame that is sent before the switch has learned the MAC address table? The switch performs flooding.

Flooding

When a switch receives a frame whose destination address is not in the MAC address table, the switch forwards the frame out on all ports except the one over which the frame was received.



Figure 23.1: Switch Logic for Unknown Destinations

By flooding the frame, Barney, Wilma, and Betty all get a copy. Barney replies, and the switch enters Barney's MAC address in the table. Any future frames sent to Barney will be forwarded correctly.

An Ethernet or MAC address represents an individual NIC attached to a LAN –unicast MAC addresses. When a computer wants to forward a frame to all devices on the LAN, Ethernet defines a special MAC address, called the broadcast address. The value is FFFF.FFFF.FFFF.



Figure 23.2: Switch Flood Broadcast Frames

Topic 24: Ethernet Network Speeds

This topic lists the different speeds that Ethernet supports.

When IEEE started working on Ethernet standards, Ethernet used a 10-Mbps transmission rate. The actual transmission of data— the bits cross the wires at 10 Mbps. Over the years, the IEEE has created many enhancements to the Ethernet standards. Increase in speed is one of the important enhancements. The first time Ethernet speed was enhanced, the IEEE created a working group named 802.3u. A specification called Fast Ethernet was created which runs at 100 Mbps. It is abbreviated as FE. The IEEE decided that the rest of Fast Ethernet should work the same as plain old Ethernet i.e. same headers and trailers, same CSMA/CD logic and same 6-byte MAC addresses. Only differences come in how the bits are encoded and the speed at which they are encoded. Vendors only needed to make small changes to existing Ethernet products to create Fast Ethernet products. This make them to be easy accepted in the marketplace. Today's enterprise networks widely use Fast Ethernet. The older 10 Mbps Ethernet, when using twisted pair cabling, was called 10BaseT. FE was called 100BaseT. The IEEE working groups 802.3z and 802.3ab created 1000 Mbps or 1 gigabit per second (Gbps). This standard is called Gigabit Ethernet (GigE). The difference between Gigabit Ethernet and Fast Ethernet is speed. To support the speed, the standards call for better cabling, but everything else is the same. This makes it to be quickly accepted in the market. The 802.3z working group defined how to do GigE over optical cable, and the 802.3ab working group defined how to do GigE over copper cabling. Optical cabling uses glass fibers instead of copper wire. The devices that are attached to an optical cable send light across the cable, instead of electricity, to encode 0s and 1s. The physics behind optical signals over optical glass-fiber cabling allow for higher speeds, longer distances, fewer errors, and better security, but at a higher cost than copper wiring. 10 Gigabit Ethernet is defined by the IEEE 802.3ae working group. 10 GigE is used for short.

Because all the more advanced forms of Ethernet use the same header, switch forwarding and learning logic do not change based on what speed Ethernet is used. The switch can behave the same way as always, with some ports using Ethernet, some Fast Ethernet, some Gigabit Ethernet, and so on. Imagine that you want to support a bunch of users on a building floor, some with NICs that only support 10 Mbps 802.3, and some that only support 100 Mbps 802.3u Fast Ethernet. You might buy a switch that has 24 ports; 12 10-Mbps ports and 12 100-Mbps ports. Next we show the basic setup, with Fred using a 10-Mbps NIC and Wilma using a 100-Mbps NIC.



Figure 24.1: Supporting Multiple Speeds on a Single Switch

This type of switch does not provide an easy migration path from 10 Mbps to 100 Mbps. For the previous example, assume all FE ports are occupied. Can you add a new 100-Mbps NIC? Auto negotiation is an IEEE standard that allows the switch and NIC on either end of the cable to automatically negotiate to determine the speed. Full duplex or half duplex can also be negotiated. To perform auto negotiation, the switch and the NIC must support multiple speeds, as well as auto negotiation logic. "I want to use 100 Mbps," "Okay, me, too. Let's do it."



Figure 24.2: Auto negotiation with 10/100 Cards

Topic 25: Multiple Physical LANs

In this topic, we understand how physical LANs and multiple physical LANs handle broadcasts.

Hubs and switches behave differently. But process broadcast frames identically. A broadcast frame is an Ethernet frame that has a destination MAC address field set to FFFF.FFFF. When a switch receives a broadcast frame, it forwards the frame out on all ports except to the incoming port. A hub works in the same fashion for all frames including broadcasts. A broadcast domain is a group of devices for which a broadcast frame sent by one device is received by all other devices in the same group. Next, we show three examples of broadcast domains. A single hub creates a single broadcast domain. A single switch also creates a single broadcast domain. A hub and a switch, connected together, also create a single broadcast domain.



Figure 25.1: Three Broadcast Domains: A Hub, a Switch, and a Hub and Switch

A LAN consists of the devices inside a single broadcast domain. Let's imagine that you just took a job as a network engineer at a company with the small network shown next.



Figure 25.2: Small Physical LAN and You Are in Charge

Your boss told you that "We've got to get Fred and Barney on a different LAN than Betty and Wilma. They work with super-secret projects, and we can't meet our security requirements if Betty and Wilma are on the same LAN". To put Betty and Wilma on a different LAN than the boys, you have to use two switches.



Figure 25.3: Two Physical LANs: Broadcasts Do Not Leave the Originating LAN

Both broadcast frames and unicast frames from the top LAN cannot be forwarded to the bottom LAN as there is no physical cable connecting the two switches in this case.



Figure 25.4: No Forwarding between the Two Physical LANs

Each LAN has an independent MAC address table as compared to the other LANs. Broadcasts originating in one LAN are flooded inside that LAN. Broadcasts originating in one LAN are not forwarded into the other LANs. Unicasts originating in one LAN are not forwarded into the other LANs.

Topic 26: Introduction to Virtual LANs

This topic explains how to create Virtual LANs.

A physical LAN or a broadcast domain is the group of devices for which a broadcast frame sent by one device is received by all devices in the group. To create multiple LANS, multiple LAN switches are required – expensive. VLANs allow you to create multiple LANs, but without requiring extra switch hardware.

A **Virtual LAN** is a broadcast domain, created by a switch, using a subset of the physical ports on the switch. LAN switch vendors include a feature in their products that allows you to create multiple broadcast domains in a single switch. A network engineer can configure some physical ports of a switch as if they are in one broadcast domain (one VLAN) and then configure other ports to be in a different broadcast domain (a different VLAN).



Figure 26.1: Two Virtual LANs: Broadcasts Do Not Leave the Originating VLAN

This network behaves just like it would with the two physical switches. However, you get the advantage of not having to buy another switch! In the previous example, the switch does learn all four MAC addresses, but the switch does not forward broadcasts or unicasts from one VLAN to the other. It keeps a separate address table for each VLAN.



Figure 26.2: No Forwarding Between the Two VLANs

The switch knows that the frame came over port EO and that EO has been configured as part of VLAN1. The switch looks only at the VLAN1 address table and finds a match. So, the switch forwards the frame. Even if there had not been a match in the VLAN1 address table, the switch would have flooded the frame, but only out on ports in VLAN1. Therefore, neither Wilma nor Betty could get a copy of the frame. Broadcasts originating in one VLAN are not forwarded into the other VLANs. Unicasts originating in one VLAN are not forwarded into the other VLANs.

Advantages of VLAN

1. A PC must spend CPU cycles processing a received broadcast. As a LAN grows, every device has more broadcasts to process. Don't to put too many devices in a single VLAN.

2. Imagine payroll department of an organization moves to the same floor as the IT department. The payroll director is concerned that we will be on the same LAN as the IT group and sensitive payroll traffic can be disclosed.

3. Cost effective to just configure VLANs.

Topic 27: Packing VLAN's frames in a Trunk

In this topic, we discuss the need for VLAN trunking.

Let's assume 4 PCs owned by Fred, Betty, Barney and Wilma are connected via two switches with two PCs per switch. This allows devices on each switch to send frames to each other. The basic switch logic for each switch does not change when you use multiple switches. Both switches will populate their own address tables and each switch knows where to forward the frames.



Figure 27.1: Forwarding Frames Between Two Switches

In the context of connecting multiple switches, the Ethernet cable segment between switch1 and switch2 is known as a trunk.

VLAN Trunking

Let's now add VLANs to the previous network such that Fred and Barney are connected to VLAN1 and Wilma and Betty are on VLAN2. Traffic must cross over the trunk between two switches. When a switch receives a frame over that trunk, it could be confused about which VLAN's MAC address table to use when deciding how to forward the frame. Should a switch look in both address tables?



Figure 27.2: Switch2's Dilemma: Which VLAN?

To solve the problem, the switches use a mechanism called VLAN trunking. Before sending the frame over the Ethernet cable to the other switch, switch1 adds another header to the frame. An extra header identifies the frame as part of VLAN1. That tells switch2 which address table to use. Switch2 expects frames coming in over port Fa24 to have that extra header, so it can now process the frame.



Figure 27.3: Trunking Header Tells Switch2 What to Do?

When switch2 needs to send frames out on port Fa24 to switch1, switch2 also adds a VLAN trunking header, knowing that switch1 expects to be able to find the header to identify the right VLAN. Initially, Cisco created its own standard and named it Inter-Switch Link (ISL). ISL defines the type of header that should be added to the frame, including the field in which the VLAN can be numbered. Later, the IEEE 802.1Q committee defined a standard for VLAN trunking called 802.1Q trunking, or simply "dot 1 Q." 802.1Q trunking differs from ISL. As a result, you need to pick between the two options. The switches on each end of the trunk must agree to which protocol to use, or trunking will not work.

Topic 28: Email: A Network Application

This topic describes how email works.

You can write a letter on paper and put it in an envelope. Write a correct name and address on the front of the envelope and put it in a mailbox. The postal service will try its best to deliver the mail to the right place. Put a return address on the envelope so that the recipient can reply by putting your address on the front of the envelope. To further elaborate the working of postal service, let's assume that you live in a house, and a postal worker typically comes by your place every working day. He leaves your mail in your mailbox. He also picks up any outgoing mail that you left either in your mailbox or in a centralized post office box (PO box) that was set up just for outgoing mail. This is shown next.



Figure 28.1: Postal Worker Picking Up and Dropping Off the Mail

Email allows you to do the same thing, but without the paper. You type some text, put that person's e-mail address at the top of the e-mail, and send the e-mail. By sending the e-mail, you do the equivalent of giving the e-mail to the postal service. The service delivers the e-mail, and the next time your friend checks his e-mail, he receives the message that you sent. Each e-mail includes the text you typed, the recipient's e-mail address, and your e-mail address. As the recipient now knows your e-mail address, he can easily respond to your e-mail. E-mail also enables you to send a message to multiple recipients at once. When you create and send an e-mail to your friend, your PC does not actually send the e-mail to your friend's PC. Instead, you would send the message to your e-mail server, which is the equivalent of dropping off a letter at the local post office. Your e-mail server would send the e-mail to your friend's e-mail server, which is the equivalent of the postal service delivering a paper letter to the post office near your friend. Then, at some point in the future, your friend would check his e-mail and retrieve the e-mail from his local e-mail server, which is the equivalent of retrieving his paper mail from his PO box.



Figure 28.2: Sending E-Mail Using E-Mail Servers

Internet service providers (ISPs) have one (or more) e-mail servers. If you connect to an ISP from home, you would use that ISP's e-mail servers to drop off and pick up email. If you use e-mail from your corporation's enterprise network, you typically use your company's e-mail servers to drop off and pick up e-mail. E-mail addresses have two parts: the name of the e-mail user and the name of the e-mail server: <u>barney@fredsco.com</u>. Let's assume that Keith and Conner send e-mails to different people inside Fredsco. Next, we show why a two-part e-mail address is useful.



Figure 28.3: How a Two-Part E-Mail Address Is Used

The e-mail software on PCs does not think about the address at all. The PCs always send their e-mails to their respective local e-mail server. The servers look at the name after the @ sign – identifies the receiving side mail server. Local post office looks at the city and state on a letter, or the zip code, to figure out to which post office to send the letter.

Topic 29: TCP/IP Email Standards: SMTP, POP3

In this topic, we explain the use of Email standards such as SMTP and POP3.

Internet Message Format, Simple Mail Transport Protocol and Post Office Protocol Version 3 are the part of TCP/IP E-mail standard. Internet Message Format (IMF) is described in RFC 2822 and describes headers used to encapsulate the e-mail text, sender and receiver e-mail addresses. Simple Mail Transport Protocol (SMTP) is defined in 2821. It provides details about protocols for transmitting and receiving e-mails. Post Office Protocol Version 3 (POP3) and is defined in RFC 1939. It provides information about protocols for a client to retrieve e-mail from a server. A user uses e-mail client software to generate, send, receive, and read e-mails.

An **Email Client** software is a user interface, accepts text typed from the keyboard, understands what a user clicks on the screen, stores e-mails on the computer hard disk, and so on. E-mail TCP/IP protocols are implemented in the e-mail client software so that it can use the network. The e-mail client application is not the same thing as the application layer protocol.



Figure 29.1: E-Mail Client Application and Its Use of Application Layer Protocols

An e-mail client sends not only the text of the e-mail message, but also a header. The header contains several fields, such as the recipient's and sender's e-mail addresses.



Figure 29.2: Sampling of the E-Mail Header Defined by RFC 2822

All the clients and servers know where to find all the information needed to forward the e-mail correctly. E-mail clients and servers use SMTP protocols to manage the process of sending and receiving e-mail. SMTP defines messages so that the e-mail clients and servers can manage the e-mail forwarding process. The Extended Hello (EHLO) command identifies the client. The Mail command tells the server that the client wants to either send or receive e-mail. The RCPT identifies recipient of e-mail. After the server replies with an acknowledgement (ACK) message to each of these first three commands, the e-mail can be transmitted.



Figure 29.3: Simple SMTP Messages: Identifying the Client and the Recipient

The current version, POP3, was designed specifically for use between an e-mail client and its e-mail server. It cannot transfer e-mail between servers. POP3 allows for authentication, which refers to the process of one device identifying itself to another device via a name and password. The other device then decides whether the first device is allowed access.



Figure 29.4: Basic Authentication with POP3

SMTP pushes emails between two devices. POP3 can be used to only pull the e-mail from the server. E-mail server (a single computer) needs to run both the server softwares. The SMTP server and the POP3 server on the same computer store and retrieve e-mails from the same message storage location.



Figure 29.5 Typical Usages of SMTP and POP3

Topic 30: File Transfer Protocol (FTP)

This topic explains the working of file transfer protocol.

The working of FTP can be explained using a warehouse that a company might use to store goods. Imagine that a company has a bunch of stuff, and many people inside the company might need some of the stuff from time to time. Some of the stuff is big, and some of the stuff is small. The company leases a warehouse space and puts

its stuff in the warehouse. Then the company gives everyone a key, and anyone who needs some stuff can go get it when needed.



Figure 30.1: Warehousing Process for Transferring Stuff

Keith should put the widgets in the right place. Conner should know where to find Keith's widgets. Warehouse becomes useless if no mechanism to find widgets and keep track of inventory. A file transfer application called File Transfer Protocol (FTP) works by putting things on an FTP server and get them off the FTP server. Documentation is also needed as to where things are.



Figure 30.2: FTP Process for Transferring Files

FTP uses client software on the end user computer. It also implements application layer protocols. The FTP TCP/IP standard defines messages and headers to do something useful. FTP clients need to connect to a server to do anything useful. The client must identify itself to the server. FTP server software runs on a physical server.

An Example:

The sequence Keith needs to follow to stores files on an FTP server is described next.

Authentication:

1- FTP client uses the FTP USER command to provide the username to the FTP server.

2- The FTP PASS command supplies the password to the server.

Data Transfer:

3- If authenticated, then FTP client uses the PUT command to inform the server that the file's name is file1 and will be available in the upcoming messages.

4- Send the actual contents of the file.

FTP Cliet Software	① FTP USER Keith → ② ← FTP OK Message	FTP Server Software
	FTP PASS Secret	
	3 ← FTP OK Message	Files: File1
	FTP PUT File1 \longrightarrow	
	4	
	File1's Contents in Here	
	← FTP OK Message	

Figure 30.3: Stocking the FTP Warehouse

Assume Conner wants to get file1. He needs to follow the same sequence to connect to the FTP server and get file1 as followed by Keith except: Instead of issuing a PUT command, Conner issues a GET command, which refers to copying a file from the server to the client.



Figure 30.4: Copying Inventory (Files) from the FTP Warehouse

FTP is defined in RFC 959. FTP uses a different underlying TCP connection for the control messages, versus the actual file transfer.

Topic 31: World Wide Web and HTTP Protocol

This topic describes the working of the World Wide Web and the HTTP protocol.

The end user PC uses software called a web browser also known as a web client. Someone must create some content and put it on a web server before the end user can do anything useful. Content consists of individual web pages, the collection of which is called a website. As an example, let's imagine a company FredsCo wants to build a website so that clients can access it. Web server software must be installed on the computer that will be used as the web server. A number of files (content) created by someone should be put onto the web server and be displayed by the http://www.fredsco.com website. Now the web server is ready to respond to clients. A customer, Conner, opens his web browser and tries to connect to <u>http://www.fredsco.com</u>. The web server gets Conner's request and sends the web page back to Conner.



Figure 31.1: Building and Stocking the New Retail Store, AKA New Website

Rules for WWW:

When Conner types http://www.fredsco.com in his browser, the web page gets loaded. The above string is called a uniform resource locator (URL). A URL identifies the protocol in use and name of the server. HTTP stands for Hypertext Transfer Protocol and is the TCP/IP protocol you use to send the web page contents from the web server to the web browser. HTTP command GET requests the web server to send the server's default web page to the browser. When a web site is developed, one web page is defined as the default. The server then looks in its configuration and finds the name of the file that holds its home page. Using HTTP, the server sends the contents of a file called home.html back to the browser.



Figure 31.2: Connecting to the Fredsco Home Page

An .HTML file contains text that conforms to a specification called Hypertext Markup Language (HTML). HTML was the first language used to describe the contents of a web page. HTML tells the browser what to put in the browser window, what color to make it, what size, and so on, but it does not define anything about how to send and receive data. When a web page gets loaded, some parts show up right away while the rest of the page fills in slowly. A lot of files get transferred between a web client and server. The first individual file (known as object) downloaded by HTTP is an HTML file. HTML files include text that goes on the web page, formatting instructions, plus instructions to download other objects. When a web page gets loaded, some parts show up right away while the rest of the page fills in slowly. A lot of files get transferred between a web client and server. The first individual file (known as object) downloaded by HTTP is an HTML file. HTML files include text that goes on the web page gets loaded, some parts show up right away while the rest of the page fills in slowly. A lot of files get transferred between a web client and server. The first individual file (known as object) downloaded by HTTP is an HTML file. HTML files include text that goes on the web page, formatting instructions, plus instructions to download other objects. When a web page gets loaded, some parts show up right away while the rest of the page fills in slowly. A lot of files get transferred between a web client and server. The first individual file (known as object) downloaded by HTTP is an HTML file. HTML files include text that goes on the web page, formatting instructions, plus instructions to download other objects.

An Example

Assume Conner types Fredsco's URL in his browser.

1- Conner's browser gets the HTML file using HTTP.

2- Conner's browser reads the file, displaying things on the screen as a result.

3 - When Conner's browser reads the file, it might have instructions that require Conner to download other objects. If so, it uses HTTP to get those objects.

4 - Conner's browser reads the contents of the new objects, displaying them on the screen.

5- Steps 3 and 4 are repeated until all objects are downloaded.



Figure 31.3: Transferring All the Files That Make Up a Web Page

Topic 32: Main Features of TCP

This topic discusses some of the main features of TCP.

The high points of TCP include ensuring delivery through error recovery, breaking large shipments into manageable sizes using segmentation, getting the data to the right individual program, not just the right computer by using port numbers, and simplifying the creation of applications by hiding the details of data delivery from the application. Assume Wilma's computer is sending three TCP segments to Fred. Sender numbers the segments so that if a receiver does not receive a segment or receives with some error, he can request sender to send again.



Figure 32.1: Assuring Data Delivery

TCP puts some interesting information into the TCP header. This is shown next.



Figure 32.2: The Format of the Shipping Label: TCP Headers

For error recovery, it uses the sequence number and acknowledgment number fields. The sequence number identifies the segment, and the acknowledgement number is used when an error occurs. The ACK field means that this number is the number of the next segment receiver expects to receive. This procedure is called forward acknowledgment.

Fred's Old Computer			Wilma's Super- Dooper Computer
←	SEQ= 1	First Part of File	\equiv
←	SEQ= 2	Second Part of File]=
-	SEQ= 3	Third Part of File] =
_=	ACK=4	(No Data)	$] \rightarrow$

Figure 32.3: Delivery Confirmation, No Errors

TCP Error-Recovery Process

Assume while Wilma sent 3 segments, segment number 2 has some errors.



Figure 32.4: TCP Error-Recovery Process

Topic 33: Working of TCP

In this topic, we present the working of TCP protocol.

If you are using a TCP/IP application, the application layer protocol uses transport layer protocol to send their messages across the network. The application layer protocols are part of the application software. For example, HTTP is used for WWW and SMTP for email. The transport layer protocols are typically part of the operating system (OS) of the computer. The transport layer software hangs around, waiting on the application program specifically the part of the application program that implements the application layer protocols to ask it to do something.



Figure 32.5: Segmenting Data Before Sending

An application layer protocol could avoid using a transport layer protocol. The transport layer protocols provide great services that many applications need. It takes a lot less time and effort for the application to use a transport layer protocol. It's better, faster, cheaper, and simpler for an application to use a transport layer protocol. A transport layer protocol provides a logical communication between application processes running on different hosts. Transport protocols run in end systems. On the sending side, transport layer breaks application messages into segments. On the receiving side, transport layer reassembles segments into messages, passes to application layer. For the **Internet** transport layer, there are two options. Transmission Control Protocol (TCP) and User Datagram Protocol (UDP). TCP provides a reliable delivery while UDP is unreliable. The **minimal** transport-layer services are process-to-process data delivery and error checking – error detection.

TCP's Encapsulation

Assume Keith's browser requests a home page. The browser does not actually send the request over the network. It asks the TCP software on Keith's computer to send it.



Figure 32.6: Acknowledging Each Byte

TCP provides several services, so it needs a place to record some information about those services. For this purpose, TCP defines a header. The TCP header and the data are called a TCP segment.



Figure 32.7: Keith's TCP Connections

TCP doesn't have to think about what HTTP is trying to do means that the TCP software does not need to know anything about how HTTP works. Software development becomes simpler.

Figures and Material used for this topic have been adapted from Kurose and Ross's book with the title *"Computer Networking: A Top-down Approach"*, 6th Edition, 2013.

Topic 34: TCP Flow and Congestion Control

This topic describes the TCP flow and congestion control mechanisms.

Hosts on each side of a TCP connection set aside a receive buffer for the connection. When the TCP connection receives bytes that are correct and in sequence, it places the data in the receive buffer. The associated application process will read the data from this buffer, but not necessarily at the instant the data arrives. If the application is relatively slow at reading the data, the sender can very easily overflow the connection's receive buffer by sending too much data too quickly.



Figure 34.1 Receive Window

Flow Control does allow the sender to overflow receiver's buffer by transmitting too much, too fast. It is TCP's speed-matching service: matching the rate at which sender is sending against the rate at which receiving application is reading.

TCP Flow Control

Sending side of TCP maintains a variable, receive window. It gives the sender an idea about the free buffer space at receiver. TCP header contains a field named Receive Window. Receiver advertises spare room available in its receive buffer in the receive window field of every segment it sends to the sender.

Congestion means too many sources sending too much data too fast for network to handle. This can result into lost packets (buffer overflow at routers), long delays (queuing in router buffers).

TCP Congestion Control

Each sender limits the rate at which it sends traffic as a function of perceived congestion. In case the network is congestion-free, upon the arrival of ACKs, TCP sender indicates a congestion-free source-to-destination path. Occurrence of either timeout or receipt of three duplicate ACKs from receiver indicates congestion. If a sender perceives that there is little congestion on the path between itself and the destination, then it increases its send rate. In case there is congestion along the path, then it reduces its send rate.

Figures and Material used for this topic have been adapted from Kurose and Ross's book with the title *"Computer Networking: A Top-down Approach"*, 6th Edition, 2013.

Topic 35: User Datagram Protocol (UDP)

This topic describes the User Datagram Protocol (UDP).

The **Internet** Transport Layer can either use Transmission Control Protocol (TCP) or User Datagram Protocol (UDP). TCP is a reliable service while UDP provides unreliable delivery.

Household Analogy:

Consider two houses, one in Lahore and other in Islamabad. Each house is a home to a dozen kids. The kids are cousins and each kid writes letter to each cousin every week (144 letters in total). Each letter is delivered by traditional postal service in a separate envelope. In each house, one kid is responsible for mail collection and mail distribution. Hassan does this job in Lahore while Umer does it in Islamabad. Each week Hassan collects the mail and gives it to the postal service mail carrier. He also distributes the received mail. Umer performs a similar job in Islamabad. The postal service provides logical communication between the two houses and not from person to person.

Hassan and Umer provide logical communication among the cousins. Application messages are like letters in envelopes. Processes resemble cousins. Hosts (also called end systems) are same as houses. Job of the transport-layer protocol is the same as that of Hassan and Umer. The network-layer protocol is like the postal service.

Assume Hassan and Umer go on vacation, and another cousin pair -- say Fahad and Ahmed replace them. Being younger, they pick up and drop off the mail less frequently and occasionally lose letters. They do not provide the same set of services as Hassan and Umer. Transmission Control Protocol (TCP), and User Datagram Protocol (UDP) offer different service models to applications.

In contrast to TCP:

UDP provides "best effort" service. Segments may be lost, delivered out of order to application. There is no connection between UDP sender, receiver. Each UDP segment is handled independently of others. UDP does not support error recovery, congestion control, and flow control.

Why is there a UDP?

No connection establishment (which can add delay) is required by UDP. No connection state at sender, and receiver, thus making it simpler. As no congestion control is supported, UDP can blast away as fast as desired. UDP is more suitable for real-time apps (online streaming video).

	← 32 bits ───→			
Length, in	source port #	dest port #		
bytes of UDP	→length	checksum		
segment, including header				
	Application data (message)			

Figure 35.1: UDP segment structure

Usage of UDP

UDP is more suitable for streaming multimedia and Internet telephony (e.g. Skype).

Topic 36: TCP Ports Numbers

In this topic, we describe the importance of TCP port numbers.

Imagine that Keith has two browsers open: one to look at http://www.fredsco.com , and one to look at http://www.espn.com. Keith also has his e-mail client software up all the time. He also has an FTP client working, downloading some files. What happens when a new TCP segment arrives at Keith's computer? The segment gets to the right computer, but it might have data for one of the two browsers, the e-mail client, or the FTP client. Which one? This issue is depicted next.



Figure 36.1: TCP ports Number

There's a field in the TCP header called the TCP destination port that tells the receiving computer which application program needs to be given the data. Let's look at the TCP header.

	Bit 31	i	Bit	Bit 15		Bit 0		
		estination Port (16)	Source Port (16) Destination					
)	mber (3	Sequence Nu				
20 Bytes	Acknowledgment Number (32)							
		leader Length (4) Reserved (6) Code Bits (6) Window (16)						
		Urgent (16)		6)	Checksum (1)			
		iy)	or 32 lf	Options (0				
			Varies)	Data (

Figure 36.2: TCP Header

For each application program that is currently being run on a computer, a unique TCP port number is assigned to it.

Keith's Co	omputer						
Browser1 Port 1030	Browser2 Port 1031	E-Mail Client Port 1040	FTP Client Port 1045				
	TCP Software⁻				Hey, TCP destin port is 1031 - g data to Browse	ation ive the 12 !	
				TCP	Segment		
				De	st. Port= 1031	Data	Ξ

Figure 36.3: A unique TCP port number is assigned

Port Numbers range from 0-65,535 as port number field is of 16-bits. Well-known Port numbers are from 0 to 1023. These are used by well-known applications. For example, a web server uses port number 80. You can find the list of all well-known port numbers on website of www.iaina.org.

Dynamically assigned port numbers:

Typically, a client side of the application lets the transport layer automatically assign a port number.



Figure 36.4: The transport layer automatically assigns a port number

Before useful data can be exchanged between a client and a server, a TCP connection is established with a three way handshaking phenomenon. A 3-way Handshaking takes place. Both endpoints agree to communicate with each other. Also, port numbers the client and server use, what values to start with in the sequence number field, and other details are also exchanged with each other.



Figure 36.5: TCP ports

Topic 37: TCP Segmentation

This topic describes process and benefits of segmentation performed by TCP.

When a TCP/IP application such as email or web client has data to send, it gives it to transport layer protocol. The application layer protocol does not have a limit to the size of bytes it chooses to send. However, TCP segments data received from the applications to ensure that a TCP segment will fit into a single link-layer frame. The maximum amount of application-layer data that TCP can grab and place in a segment is called Maximum Segment Size (MSS). 1480 is a typical value for MSS. The process of breaking the application data into parts is called segmentation. The bytes that include the data and the TCP header are called TCP segments.

Example:

Let's assume a client contacts a web server. The home page has 3000 bytes of data. HTTP gives all 3000 bytes to TCP. MSS of TCP is 1480. Put 1460 bytes into data part of a segment.



Figure 37.1: TCP Segmentation

Segmentation brings benefits too?

Imagine you download a web page, which contains a graphics file called logo.gif. Its size is 146 KB. If TCP sends all the data in one TCP segment a single bit can get an error during transmission. This will result in TCP will have to resend the entire packet. Now let TCP sends logo.gif as 100 segments with 1460 bytes in each segment. If a single bit error occurred, only one segment would have to be re-sent. Thus, improving the network performance.

When TCP segments the application data, it uses the first byte of each segment for sequence number. TCP actually numbers the bytes, not the segments. This is shown next.



Figure 37.2: TCP segment the application Data

Topic 38: Basics of Internet Protocol (IP)

In this topic, we provide the basics of Internet Protocol (IP).

In a typical large network, many routers and LAN switches sit between a client PC and a server. **Routers** are networking devices that connect to multiple physical networks, such as multiple Ethernets. They make decisions about where to forward the data so that it reaches the correct destination.
Example

Assume Hannah on a LAN in Mason wants to get a web page from a web server http://www.cisco.com on a LAN in Cincinnati. This is shown next.



Figure 38.1: Topic 38: Basics of Internet Protocol (IP)

Routing is the complete process by which a computer sends the data, passing through all the routers and eventually arriving at the destination. Previous example shows only LANs, the routers could also be connected to a wide-area network (WAN).

Two Key functions

Forwarding is moving a packet from router's input to appropriate router output. Routing is the determination of the route taken by packets from source to destination. These ideas are shown next.



Figure 38.2: Routing with connected to wide area network (WAN)

IP provides logical communication between hosts. IP makes its best effort to deliver segments between communicating hosts, but it makes no guarantees. IP is said to be an unreliable service.



Figure 38.3: Transport Layer TCP, UDP

Topic 39: Working of IP and IP addresses

This topic explains the working of IP and usage of IP addresses.

Internet Protocol (IP) is the network layer protocol of TCP/IP model. IP defines addressing, forwarding and routing. The addressing details recommended by IP facilitate easy and efficient forwarding of IP packets. IP addresses are 32-bit binary numbers.



Each of the decimal numbers in an IP address is called an octect. Range falls in [0 255]. 2³² is equal to 4 billion possible IP addresses. Everyone should have a unique IP address to avoid confusion when trying to deliver data to that address. Each network interface on a computer needs an IP address. A network interface is simply a card that has a physical connector for some type of network. Ethernet NIC is one such example. The NIC takes care of TCP/IP network interface layer details, which are the equivalent of OSI Layers 1 and 2. Most end user computers also known as TCP/IP hosts have a single network interface. Devices that have more than one network interface have more than one IP address. Routers are such devices. IP defines a 20-byte long header, which includes a 4-byte source IP address and a 4-byte destination IP address. An IP packet includes the IP header, along with any data that follows the IP header. To send data from one computer to the other, a sender puts the destination computer's IP address. This is shown next.



Figure 39.1: IP address field

Next, we show the process of getting a home page. Application layer generates an HTTP GET request. TCP adds its header. The TCP segment gets encapsulated in an IP packet.



Figure 39.2: TCP segment gets encapsulated in an IP packet

HTTP is concerned about getting a web page to the user. TCP is concerned about segmentation, error recovery, and other things, on behalf of HTTP. IP provides an end-to-end delivery of packets service to TCP. Hannah types web server's name http://www.fredsco.com and corresponding IP address is 3.3.3.3. Names are more human friendly while addresses are more computer friendly.

Topic 40: How to run an IP Network?

This topic explains how an IP network runs.

The postal service provides a wonderful service. You put a letter into the mailbox, and it magically appears at the right address. By sending an IP packet into the network, the networking devices should collectively be able to forward the packet to the right destination. An example of an IP address is 1.1.1.1. There are 4 decimal numbers, between 0 and 255 inclusive, separated by periods. The structure and meaning of IP addresses tell us how IP addresses are used in an internetwork. To understand the concepts, let's take postal service as a reference. In addition to the number, street, town, and state, the postal service uses postal codes to make it easy

to sort the mail. Everyone in the same general geographical area such as a town has the same postal code. The only time the whole postal address is needed is when the letter gets to the final post office; the one that handles mail for that postal code.



Figure 40.1: IP network

All IP addresses on the same physical network such as one Ethernet LAN have a portion of their IP addresses in common. This common portion works like a postal code. Like postal codes, all the IP addresses on the same Ethernet are in the same general area, so routing can take advantage of that fact.



Figure 40.2: Same Ethernet are in the same general area

IP address grouping makes routing easy. Let's imagine the logic needed by R1: Packets whose destination begins with 1 should be forwarded to the left. Packets whose destination begins with 2 should be forwarded to the Ethernet on the right. Packets whose destination begins with 3 should be forwarded to R2 so that R2 can forward the packet. The routers do not have to know about every IP address in the network. IP calls the group of IP addresses that share a common beginning to part of their addresses an IP network. The common portion is called network number and is used to represent a network. For an IP network number, the bits other than the common portion have all 0s. In previous example, the network numbers are 1.0.0.0, 2.0.0.0 and 3.0.0.0. A host cannot use a network number as an IP address in the network.

Topic 41: Classes of IP Networks

This topic describes the classes of IP networks.

Let's assume a simple internetwork with two routers and three IP networks.



Figure 41.1: Classes of IP Networks

How many IP addresses can you have in network 1.0.0.0? You can't use 1.0.0.0 and 1.255.255.255 – are reserved. Valid addresses: more than 16 million addresses. IP defines three sizes of networks as different classes. The three different network classes are called Class A, B, and C. All addresses in the same Class A, B, or C network have the same numeric value for the network portion of the addresses. The rest of the address is called the host portion of the address.

Any Network of This Class	Number of Network Bytes (Bits)	Number of Host Bytes (Bits)	Number of Addresses per Network
А	1 (8)	3 (24)	2 ²⁴ - 2, or 16,777,214
В	2 (16)	2 (16)	2 ¹⁶ - 2, or 65,534
С	3 (24)	1 (8)	2 ⁸ - 2, or 254

Let's assume the same previous internetwork. By using one IP network number for each physical network, allows easy routing.



Figure 41.2: Classes of IP Networks

Class A networks allow for a ton of IP addresses in a single IP network. Class B networks allow for a pretty large number, and Class C networks allow for a smaller number of host IP addresses. The people who made up IP addressing chose three sizes of networks because one size doesn't fit all companies and organizations. Larger companies can use Class A, medium-sized companies can use Class B, and small companies can use Class C.

Class	First Octet Range	Valid Network Numbers	Total Number of Networks of This Class
А	1 to 126	1.0.0.0 to 126.0.0.0	2 ⁷ -2, or 126
В	128 to 191	128.1.0.0 to 191.254.0.0	2 ¹⁴ -2, or 16,382
С	192 to 223	192.0.1.0 to 223.255.254.0	2 ²¹ -2, or 2,097,150

Topic 42: IP Subnetting

In this topic, we explain the use of IP subnetting.

These days, single LANs with more than 1000 devices are extremely rare. Many IP addresses will go unused if you use a Class A or Class B network for the devices on a LAN. Subnetting provides a solution. Let's assume that a design requires three networks, and uses three different Class B networks. Each Class B network can host 65,534 addresses.



Figure 42.1: IP Subnetting

A subnet is just a subdivision of a larger Class A, B, or C network. Subnetting refers to the process whereby the engineer decides to create subnets. **Without Subnetting**, devices in the same Class A, B, or C network cannot be separated from each other by a router. Also, devices in different Class A, B, or C networks must be separated from each other by a router. **With Subnetting**, devices in the same subnet cannot be separated from each other by a router. Also, devices in different Subnetting, devices in different subnets must be separated from each other by a router. Also, devices in different subnets must be separated from each other by a router. Subnetting allows a network engineer to configure network devices of a class B network for example, 150.1.0.0, such that they can think that the first 3 octets of the addresses identify the network. Following subnets can be created: all addresses that begin with 150.1.1, all addresses that begin with 150.1.2, all addresses that begin with 150.1.3 and so on.

An IP subnet number is used to represent a subnet. The subnet number has the same value in the first part of the number as all the host addresses, and 0s in the last part.



Figure 42.2: IP Subnetting

This internetwork of three Ethernet LANs only uses a part of Class B network 150.1.0.0. The same internetwork without subnetting fully uses three B classes (150.1.0.0, 150.2.0.0, and 150.3.0.0).

Figures and Material used for this topic have been adapted from Kurose and Ross's book with the title *"Computer Networking: A Top-down Approach"*, 6th Edition, 2013.

Topic 43: Network Address Translation (NAT)

This topic describes explain the concept of NAT.

Small office, home office (SOHO) subnets connect devices such as cameras, computers, smartphones, game consoles. Every IP-capable device needs an IP address. A SOHO needs a range of addresses that can be allocated by the ISP. With 32-bits, possible IP addresses are equal to 2³² or 4 billion. If the subnet grew bigger, a larger block of addresses would have to be allocated. But what if the ISP had already allocated the contiguous portions of the SOHO network's current address range? Network address translation (NAT) is a simpler approach to address allocation. It is defined in RFC 2663. It has found increasingly widespread use.

An Example

Assume a home network with 3 computers. A NAT-enabled router, residing in the home has an interface that is part of the home network. All four interfaces in the home network have the same subnet address of 10.0.0/24.



Figure 43.1: Network Address Translation (NAT)

There can be hundreds of home networks, many using the same address space, 10.0.0.0/24. Devices within a given home network can send packets using 10.0.0.0/24 addressing. How is addressing handled when packets are sent to or

received from the Internet? The NAT-enabled router does not look like a router to the outside world. Instead, it behaves to the outside world as a single device with a single IP address. All traffic leaving the home router for the Internet has a source IP address of 138.76.29.7, and all traffic entering the home router must have a destination address of 138.76.29.7 NAT-enabled router hides the details of the home network from the outside world. If all datagrams arriving at the NAT router from the Internet have the same destination IP address, then how does the router know the internal host to which it should forward a given datagram?



Figure 43.2: NAT-enabled router

Motivations:

Local network uses just one IP address as far as outside world is concerned: 1) range of addresses not needed from ISP. 2) Addresses of devices in local network can change without notifying outside world. 3) ISP can change without changing addresses of devices in local network.

Figures and Material used for this topic have been adapted from Kurose and Ross's book with the title *"Computer Networking: A Top-down Approach"*, 6th Edition, 2013.

Topic 44: Dynamic Host Configuration Protocol (DHCP)

In this topic, we describe the Dynamic Host Configuration Protocol.

An organization obtains a block of IP addresses from an ISP. Then, it can assign individual IP addresses to the host and router interfaces in its organization. A system

administrator will typically manually configure the IP addresses into the router. Host addresses can be configured manually or Dynamic Host Configuration Protocol.

1- Manual configuration: manually done

- Windows: control-panel->network-> configuration-> tcp/ip-> properties
- UNIX: /etc/rc.config

2- DHCP: allows a host to dynamically obtain its IP address from network server when it joins network. Because a host gets connected automatically into a network, it is "plug-and-play" protocol.

Motivating Example1

Consider a student who carries a laptop from a dormitory room to a library to a classroom. In each location, he connects to a new subnet and hence will need a new IP address.

Motivating Example2

Consider a residential ISP that has 2,000 customers, but no more than 400 customers are ever online at the same time. The ISP does not need 2,048 IP addresses. Only get a block of 512 IP addresses. As the hosts join and leave, the DHCP server allocates an arbitrary address from its current pool of available addresses. Each time a host leaves, its address is returned to the pool.

DHCP is ideally suited to these situations, as there are many users coming and going, and addresses are needed for only a limited amount of time. DHCP is a client-server protocol. A client is typically a newly arriving host wanting to obtain network configuration information, including an IP address for itself.



Figure 44.1: Dynamic Host Configuration Protocol (DHCP)

Working of DHCP:

For a newly arriving host, DHCP protocol is a four-step process.

1- DHCP server discovery: Host generates a UDP packet with destination port 67. Broadcast to all nodes attached to subnet.

2- DHCP server offer: Server generates a message with proposed IP address for the client, the network mask, and an IP address lease time.

3- DHCP Request: Client echoes back the configuration parameters.

4- DHCP ACK: Server confirms the requested parameters.



Figure 44.2: Working of DHCP

Figures and Material used for this topic have been adapted from Kurose and Ross's book with the title *"Computer Networking: A Top-down Approach"*, 6th Edition, 2013.

Topic 45: Internet Control Message Protocol (ICMP)

In this topic, we explain the usage of ICMP.

There are three Major Components of Network Layer namely, IP protocol, routing component, and a facility to report errors in IP datagrams called ICMP.

ICMP is used by hosts and routers to communicate network-layer info to each other. Error messages: unreachable host, network, port, etc. Informational messages: echo request/reply. ICMP is often considered part of IP, but architecturally lies above IP. ICMP messages are carried in IP datagrams as IP payload. An **ICMP message** consists of type, code and first 8 bytes of IP datagram that caused the ICMP message to be generated in the first place (sender can determine the datagram that caused the error).

Тур	e Code	description
0	0	echo reply (ping)
3	0	dest. network unreachable
3	1	dest. host unreachable
3	2	dest. protocol unreachable
3	3	dest. port unreachable
3	6	dest. network unknown
3	7	dest. host unknown
4	0	source quench (congestion control)
8	0	echo request (ping)
9	0	route advertisement
10	0	router discovery
11	0	TTL expired
12	0	bad IP header

Many tools obtain information about the Internet by sending probes and waiting for ICMP responses. For example, ping , traceroute etc.

Ping and ICMP

The ping program sends a datagram to a specified destination to test if it can be reached. It then reports the results of the probe by declaring whether the destination responds. The ping uses ICMP echo messages. When a user invokes it by giving the IP address or hostname, ping sends datagram that contains an ICMP type 8 code 0 message (echo request) to the specified host. The destination host, seeing an echo request, sends back a type 0 code 0 ICMP echo reply message.

Traceroute & ICMP

It allows us to trace a route from a host to any host in the world. It employs TTL expired and a destination port unreachable ICMP messages. The source sends a series of IP datagrams to the destination. Each carries a UDP segment with an unlikely UDP port number.



Figure45.1 Traceroute & ICMP

The **Stopping Criterion** creates an UDP segment eventually arrives at destination host. Destination returns ICMP "destination port unreachable" ICMP (type 3, code 3) message. When source gets this ICMP, it stops.

Figures and Material used for this topic have been adapted from Kurose and Ross's book with the title *"Computer Networking: A Top-down Approach"*, 6th Edition, 2013.

Topic 46: End-to-End: Processing IP Packet

In this topic, we describe fragmentation i.e. processing of IP packet along the end-toend route.

At the link-layer, an IP datagram is encapsulated within a link-layer frame. All link layer protocols cannot carry network-layer packets of same size e.g. Ethernet frames can carry up to 1500 bytes of data. The maximum amount of data a link-layer frame can carry is called Maximum Transmission Unit (MTU). The MTU of the link-layer protocol places a hard limit on the length of an IP datagram.

Problem

Each of the links along route between sender and destination use different link-layer protocols with different MTUs. When a router is about to forward an IP packet, it may find that its length is greater than MTU.

Solution

Fragment data in IP datagram into two or more smaller IP datagrams, and encapsulate each of these smaller IP datagrams in a separate link-layer frame.

IP Reassembly & Fragmentation

Each of these smaller datagrams is referred to as a fragment. Fragments need to be reassembled before they reach the transport layer at the destination. Both TCP and UDP are expecting to receive complete, un-fragmented segments from the network layer. Datagram reassembly is done in the end systems. Keep the network core simple.



Figure 46.1: IP Reassembly & Fragmentation

There are three fields in IP header. **Identification:** allows a destination host to determine which datagram a newly arrived fragment belongs to. A sender increments identification number for each datagram. **MF:** stands for more fragments and is set for all fragments except the last one. **Fragmentation offset:** tells where in the current datagram this fragment belongs.



Figures and Material used for this topic have been adapted from Kurose and Ross's book with the title *"Computer Networking: A Top-down Approach"*, 6th Edition, 2013.

Topic 47: A Link State Routing Algorithm

In this topic, we explain the working of a link state routing algorithm.

When an IP packet arrives to a router, the router indexes a forwarding table and determines the link interface to which the packet is to be directed. Routing algorithms, operating in network routers, exchange and compute the information

that is used to configure these forwarding tables. The job of routing is to determine good paths from senders to receivers, through the network of routers. A "good" path is one that has the least cost, e.g. physical length of the link, link speed, or monetary cost associated with it.



graph: G = (N,E)

```
N = set of routers = { u, v, w, x, y, z }
```

 $\mathsf{E} = \mathsf{set} \; \mathsf{of} \; \mathsf{links} = \! \{ \; (\mathsf{u},\mathsf{v}), \; (\mathsf{u},\mathsf{x}), \; (\mathsf{v},\mathsf{x}), \; (\mathsf{v},\mathsf{w}), \; (\mathsf{x},\mathsf{y}), \; (\mathsf{w},\mathsf{y}), \; (\mathsf{w},\mathsf{z}), \; (\mathsf{y},\mathsf{z}) \; \}$



Figure 47.1: A Link State Routing Algorithm

Link State routing algorithm uses global information such as network topology and all link costs are known. Each node broadcasts identities and costs of its attached links to all other nodes.

Pseudo-code: 1 Initialization: 2 N' = {u} 3 for all nodes v if v adjacent to u 4 then D(v) = c(u,v)5 else $D(v) = \infty$ 6 7 Loop 8 find w not in N' such that D(w) is a minimum 9 add w to N' 10 update D(v) for all v adjacent to w and not in N' : $D(v) = \min(D(v), D(w) + c(w,v))$ 11 12 until all nodes in N' or N' = N

Notation

c(x,y): link cost from node x to y; = ∞ if not direct neighbors. D(v): current value of cost of path from source to destination v. p(v): predecessor node along path from source to v. N': subset of nodes



Figures and Material used for this topic have been adapted from Kurose and Ross's book with the title *"Computer Networking: A Top-down Approach"*, 6th Edition, 2013.

Topic 48: Distance Vector Routing Algorithm

This topic explains the working of a distance vector routing algorithm.

Bellman-Ford Equation

If dx(y) := cost of least-cost path from x to y, then $dx(y) = min \{c(x,v) + dv(y)\}$, where min is taken over all neighbors v of x.

Bellman-Ford Example



Node that achieves minimum is next hop in shortest path. Thus, the solution to the Bellman-Ford eq. provides the entries in a node's forwarding table. Each node x maintains the following routing information:

1-For each neighbor v, the cost c(x,v)

2-Node x's distance vector $Dx = [Dx(y): y \in N]$, where Dx(y) = an estimate of cost of least-cost path from itself to y.

3- The distance vectors of each of its neighbors. i.e., $Dv = [Dv(y): y \in N]$ for each neighbor v of x

Basic idea:

From time-to-time, each node sends a copy of its distance vector to each of its neighbors. When a node x receives a new distance vector from any of its neighbors v, it saves v's distance vector, and then uses the Bellman-Ford equation to update its own distance vector as follows. If x's distance vector has changed as a result of this update step, x will then send its updated distance vector to each of its neighbors, which can in turn update their own distance vectors. The estimate Dx(y) converges to the actual least cost dx(y).

An Example:





The DV algorithm continues until no update messages are sent.



Topic 49: Working of Default Gateway Router

In this topic, we explain the working of a default gateway router.

Let's assume that Hannah wants to access the webpage www.example.com residing on a webserver. The internetwork consists of three different IP networks, each with one IP network number.



Figure 49.1: Working of Default Gateway Router

After IP software of Hannah's PC has built the IP packet that needs to be sent to the server, it needs to know where to send the packet first. To send the packet to R1, Hannah needs to know R1's IP address of the router interface that's connected to the same Ethernet as Hannah. R1 is termed as Hannah's default router/gateway. A PC's default router is simply the router to which that PC sends packets when the destination is in another network or subnet.



Figure 49.2: Software of Hannah's PC has built the IP

Routers typically have one IP address per physical interface. Routers have lots of interfaces of many different types, which are labeled with a name and a number, such as Ethernet1.

Topic 50: Address Resolution Protocol

In this topic, we explain how IP packets get encapsulated in Ethernet frames and how ARP works. Let's assume that Hannah's PC is connected to an internetwork that uses three different IP networks, each with one IP network number. She accesses www.example.com from a webserver.



Figure 50.1: Address Resolution Protocol

Two issues need attention. 1 - Hannah can't send an IP packet over an Ethernet, but she can send an Ethernet frame over an Ethernet. 2- The destination address field of IP header holds IP address of the destination web server. There is no place for "default gateway IP address" in the header. Hannah must have some other means to ensure that R1 receives the packet. **Solution of issue 1:** Encapsulation. Hannah encapsulates the IP packet in an Ethernet frame for transmission over the LAN. Ethernet places IP packet between a header and trailer. The Ethernet header contains source and destination Ethernet address fields. The trailer contains a frame check sequence field (FCS) to determine whether errors occurred during transmission.







Figure 50.3: Solution of issue 1

Solution of issue 2: To deliver the packet to R1, Hannah includes R1's Ethernet address as the destination in the frame.



Figure 50.4: Solution of issue 2

Hannah learns the IP address of default router manually or via Dynamic Host Configuration Protocol (DHCP). To learn MAC address of R1, Hannah uses address resolution protocol (ARP).



Figure 50.5 Dynamic Host Configuration Protocol (DHCP)

Hannah generates an ARP broadcast "Hey, if this is your IP address, tell me your Ethernet MAC address." Everyone on the LAN gets ARP broadcast as switch forwards LAN broadcasts to all devices in network.



Figure 50.6: Ethernet MAC Address

Topic 51: Router's Routing Logic and Table

This topic describes the router's routing logic and contents of IP routing table.

Let's assume that Hannah's PC is connected to an internetwork that uses three different IP networks, each with one IP network number. She accesses www.example.com from a webserver.



Figure 51.1: Router's Routing Logic and Table

As R1 receives the packet, it needs to forward it to R2 next. This process can be broken down into 3 steps: 1. De-capsulation. 2. Decide where to forward the packet next. 3. Encapsulation.



1. De-capsulation: Check the incoming frame's FCS. If there are errors, discard it. Otherwise, remove header and trailer, leaving the original IP packet whose source and destination addresses are 8.1.1.1 and 130.4.3.3

2. Decide where to forward the packet next: A router makes a decision about where to forward the packet next by looking at the destination IP address of the IP packet. For routing to work well, the router needs to know how to reach the various IP networks and subnets in the internetwork. This info is contained in a routing table. RIP and OSPF are examples of routing protocols.



3. Encapsulation: At this point, R1 has an IP packet sitting in memory. To send the IP packet out on E2, R1 needs to encapsulate it in a frame. In the newly created Ethernet frame: source address is MAC address of my E2 interface. Destination address is the MAC address of R2. If not known, use ARP to find R2's MAC address – put it in a cache for future use.



Figure 51.2: Encapsulation

Logic of Final Router: R2 has the same logic as that of R1 except R2's routing table differs from R1's: R2 does not need to send the packet to another router, but it should instead send the packet directly to the web server.



From R2 to web server: Create a new Ethernet frame. Own Ethernet address on the outgoing interface as source MAC. Use ARP to learn web server's MAC and use it as

destination. Finally, the web server has the frame and extracts the packet. The routing of the packet is complete!

Topic 52: Routing with Subnets

This topic explains how a router's routing logic works with subnets.

Let's assume an internetwork of a single Class B IP network 150.1.0.0. This IP network is subdivided into three subnets namely, 150.1.1.0, 150.1.2.0, and 150.1.3.0. All hosts in a given IP subnet have the same value in the first 3 octets of their IP addresses. Next, we look at the routing tables.



Figure 52.1: Routing with Subnets

When Hannah sends a packet to the web server, destination address is 150.1.3.3. Hannah's logic works like it did without subnettting. Hannah's default gateway is R1. Hannah sends an ARP broadcast for IP address 150.1.1.250. R1 replies and Hannah sends a frame to R1 with the IP packet inside the Ethernet frame. R1 receives the frame, and if it is error free, it extracts the IP packet. R1 uses destination address and finds an entry for it in its IP routing table. R1 forwards packet out on its E2 interface to R2 next. Finally, when R2 receives the frame, it does the usual error check and extracts the IP packet. R2 finds an entry for destination IP address. R2 can forward the packet directly to the web server. Whether or not an internetwork uses subnetting, the basic routing logic remains the same. If Hannah sends a packet to someone who's on the same subnet, she can send it directly to the destination without involving default router.



Figures and Material used for this topic have been adapted from Kurose and Ross's book with the title *"Computer Networking: A Top-down Approach"*, 6th Edition, 2013.

Topic 53: Router Hardware Architecture

In this topic, we explain the router hardware components.

There are two key router functions: run routing algorithms (RIP, OSPF) and, forwarding datagrams from incoming links to appropriate outgoing links. A Generic router architecture is shown next.



Figure 53.1: Router Hardware Architecture

Input Ports

Input ports perform physical & data link layer functions of an incoming physical link at a router. They determine output port to which an arriving packet will be forwarded via the switching fabric.



Figure 53.2: Input Ports

Switching fabrics

Switching fabrics transfer packet from input buffer to appropriate output buffer. Rate at which packets can be transferred from inputs to outputs is called switching rate. The three types are shown next.



Figure 53.3: Switching fabrics

Output port



- buffering required when datagrams arrive from fabric faster than the transmission rate
- scheduling discipline chooses among queued datagrams for transmission



Where does queuing occur?

Packet queues can form at both input and output ports. The location and extent of queuing will depend on the traffic load, speed of switching fabric and line speed. As these queues grow large, router's buffer space will eventually be exhausted and packet loss will occur when no memory is available to store arriving packets. Suppose switch speed is N times faster than speed of line. Even then output buffering will still occur when multiple inputs send to same output. This scenario is shown next.



Figure 53.5: Packet queues

The Head-of-the-Line (HOL) blocking at Input Port is shown next.



Figure 53.6: The Head-of-the-Line (HOL) blocking at Input Port

Topic 54: Routing to Nearby Places

In this topic, we explain how a router learns routing table for networks/subnets that are directly connected to it.

At a given time, a router knows which of its physical interfaces are up and working. It knows the IP addresses used on each interface. Also, it knows about IP networks or subnets that are connected to its interfaces. The router can add a route to the subnet in its routing table.

Configuring a Router When you buy a brand new router, a network engineer connects to the router and tells the router which IP addresses to use. For each interface, use an IP Address and Subnet mask to figure out the range of valid IP addresses in each subnet.



Figure 54.1 Configuring a Router

After R1's interfaces are up, it knows about subnets: the subnet numbers, the outgoing interface it should use to forward packets to them. There is no need to send packets to another router for them. In our previous example, R1's Ethernet1 interface is connected directly to the same subnet as Hannah. As a result, R1 simply adds a route for each directly connected subnet to its routing table.



Figure 54.2: Configuring a Router

Routers always add routes for directly connected subnets and networks, as long as the interface is both configured and working. Also, they should include other routes in their routing tables. In previous example, R1 didn't have a route to subnet 150.1.3.0, so it could not forward a packet that was destined for IP address 150.1.3.3.

Static IP route: The network engineer can address this problem by configuring a static route on R1 for subnet 150.1.3.0, with outgoing interface Ethernet2, and next-hop router of 150.1.2.2.



Figure 54.3: Static IP route

Network topologies tend to change a lot. Static routes make it difficult to use all the possible routes to the same part of the network when you have multiple possible physical paths.

Topic 55: Dynamically Learning Routing Tables

This topic explains explain how routing table entries are learnt dynamically.

The most typical way a router learns all the rest of the routes in an internetwork, beyond just its directly connected routes, is by using a routing protocol. **Routing Protocols** define messages by which routers can exchange route information with other routers. If all the routers participate, all routers should have routes for all

subnets or networks. Let's assume an internetwork that consists of two routers and three subnets. Several steps that occur over time are:

- 1- Each router knows only its directly connected routes.
- 2- A router generates a routing protocol message (routing update) that contains information about IP networks and subnets and send it to another router.
- 3- The receiving router: a) puts that interface into the route in which it received the routing update. b) Also puts the IP address of the router that sent the routing update as the next-hop router.



Figure 55.1: Dynamically Learning Routing Tables

Next, we show R2 advertising its routes, with R1 learning a route to subnet 150.1.3.0.



Figure 55.2: Dynamically Learning Routing Tables

Topic 56: How to Pick the Best Route

In this topic, we describe how the best route gets selected.

Routing protocols not only help routers learn routes, but they also help routers learn the best routes to a destination when there is more than one way to get there. Let's assume an internetwork consisting of three routers. R1 can send packets to subnet 150.1.3.0 through R2 or through R3.



Figure 56.1: How the best route gets selected

Routing protocols define a number called a metric that's associated with each route in a routing update. That number represents how good or how bad that route is. When a router receives multiple routing updates, it might learn of multiple ways to reach a subnet. By looking at the metrics, the router can pick the best route.



Figure 56.2: The router can pick the best route

Imagine that someone turns off the power on the Ethernet switch between R1 and R2. That route would then fail and be unavailable. The routing protocol on R1 would remove the route from the routing table. At the same time, R1's routing protocol

would notice that another route exists the one through R3 and add that route to the routing table.

Figures and Material used for this topic have been adapted from Kurose and Ross's book with the title *"Computer Networking: A Top-down Approach"*, 6th Edition, 2013.

Topic 57: Interior & Exterior Routing Protocols

In this topic, we explain the need for interior and exterior routing protocols.

Hierarchical Routing

Considering network to be flat, taking all routers to be identical and running one single routing protocol is not realistic. At least two reasons:

1-Scale: with hundreds of millions destinations can't store all destinations in routing tables. Similarly, routing table exchange would swamp links.

2-Administritative Autonomy: internet is a network of networks. Each network admin may want to control routing in its own network

Autonomous System (AS): Organize routers into regions. An AS is under the same administrative control. For example, operated by the same ISP or belonging to same company network. All routers within the same AS run the same routing protocol (intra-AS or interior routing protocol) and have information about each other. Routers in different AS can run different intra-AS routing protocols. To connect ASes to each other, one or more of the routers in an AS will have to be responsible for forwarding packets to destinations outside the AS – Gateway routers.



If a router in AS1 receives an IP packet destined outside of AS1, then it uses an inter-AS (exterior) routing algorithm 1- to learn which destinations are reachable through AS2, which through AS3, and 2- propagate this information to all routers in AS1. This is shown next.

Forwarding table configured by both intra- and inter-AS routing algorithm



Well-known IP routing protocols are listed next.

IP Routing Protocols

Routing Protocol	Public or Proprietary?	Interior or Exterior?
Routing Information Protocol (RIP)	Public	Interior
Interior Gateway Routing Protocol (IGRP)	Proprietary	Interior
Open Shortest Path First (OSPF)	Public	Interior
Enhanced IGRP (EIGRP)	Proprietary	Interior
Border Gateway Protocol (BGP)	Public	Exterior

Topic 58: Introduction to Domain Name System

In this topic, we explain why a domain name system (DNS) is required.

People have many identifiers. For example, name, passport#, NTN#, NIC#. One identifier cannot be used everywhere. Can Hassan Hamid be used to recognize a tax payer? Similarly, Internet hosts have many identifiers. A "hostname", e.g., www.yahoo.com is human friendly but difficult for routers to process variable length alphanumeric characters. IP address (32 bit) is router friendly. If you need to call someone and you know the person's name but not the phone number, you can just look up the information in the phone book. It's simple and easy provided the info is present in the book. TCP/IP hosts can have the equivalent of a phone book in a file. The local host file contains a list of TCP/IP host computer names and their corresponding IP addresses. Assume Hannah types http://www.example.com in a

browser. Hannah's PC can send an IP packet to the web server, if its IP address is known. Look local host file (www.example.com, 150.1.3.3).



Figure 58.1: Domain name system (DNS)

A local host file cannot list the names and IP addresses of every server on the planet. If the pair changes, conveying this updated info to everyone is cumbersome. TCP/IP uses a domain name system (DNS) instead of local host file. DNS defines how to discover which names correspond to which IP address. DNS also defines the structure and format of TCP/IP host and names.

Figures and Material used for this topic have been adapted from Kurose and Ross's book with the title *"Computer Networking: A Top-down Approach"*, 6th Edition, 2013.

Topic 59: Domain Name System (DNS) Servers

In this topic, we explain the need for different domain name system (DNS) servers.

DNS is an application-layer protocol that allows hosts to query the database and to resolve names (name, address). It is employed by other application-layer protocols such as HTTP, SMTP, and FTP.

Problems with centralized DNS

Single point of failure: If the DNS server crashes, so does the entire Internet. Traffic volume: a single server would have to handle all DNS queries. Distant centralized database: a single server cannot be "close to" all the querying clients –significant delays. Maintenance: a centralized database would have to be updated frequently for every new host. In order to deal with the issue of scale, the DNS uses a large

number of servers, organized in a hierarchical fashion and distributed around the world.

Root DNS Servers org DNS servers edu DNS servers com DNS servers poly.edu umass.edu pbs.org yahoo.com amazon.com DNS DNS servers DNS servers DNS servers DNS servers servers Level 2: Top Level Domain (TLD) Level 3: Authoritative DNS servers

Distributed, Hierarchical Database

Interaction of Servers: an approximation

Let's assume a client wants IP address for www.amazon.com. 1- The client queries a root server to find com DNS server. 2- The client queries com DNS server (TLD) to get amazon.com DNS server. 3-client queries amazon.com DNS server (Authoritative) to get IP address for www.amazon.com.

Root Name Servers: in the Internet there are 13 root servers. **Top-level Domain (TLD) Servers** are responsible for com, org, net, edu, gov, and all top-level country domains, e.g.: uk, fr, ca, jp. **Authoritative Servers:** organizations with publicly accessible hosts (e.g. web servers) must provide publicly accessible DNS records that map the names of those hosts to IP addresses. **Local DNS Servers** do not strictly belong to hierarchy. Each ISP has one server which is also called "default name server". When host makes DNS query, query is sent to its local DNS server.



DNS Caching: once (any) name server learns mapping, it caches mapping.

Topic 60: Working of Domain Name System

In this topic, we explain working of domain name system (DNS) inside and outside company.

DNS defines how to figure out names and IP addresses for the entire Internet, as well as inside a single site at a single company.

Inside the Company: Let's assume that Hannah's PC is inside the example.com corporation's enterprise network. She wants to view http://www.example.com.



Figure 60.1: Working of Domain Name System

For DNS to work well inside a company, someone must be responsible for supporting it. That work includes updating and changing the list of names and IP addresses.

Outside the Company: Let's assume that Hannah wants to access the http://www.fredsco.com website, which is located in Fredsco enterprise network.



Figure 60.2: Fredsco enterprise network

١
The DNS servers must work together as shown next.



Figure 60.3: The DNS servers

Figures and Material used for this topic have been adapted from Kurose and Ross's book with the title *"Computer Networking: A Top-down Approach"*, 6th Edition, 2013.

Topic 61: DNS Resource Records

This topic describes the use of domain name system (DNS) Resource Records.

The domain name service (DNS) is a directory service, which translates hostnames into IP addresses. It is a distributed database implemented in a hierarchy of many DNS servers. The DNS distributed database is stored on servers in the form of resource records (RRs):

DNS Resource Record Format:

RR format: (name, value, type, TTL)

Time to live (TTL) – after this time a resource record becomes invalid

The meaning of **name** and **value** depend on **type**

The type field is defined as:

Type=A T • name is hostname • value is IP address • (relay1.bar.foo.com,	 Type=CNAME value is canonical name for the alias hostname Name. querying hosts learn the canonical
 145.37.93.126, A) Type=NS name is domain value is hostname authoritative name s that knows how to ol IP addresses for host this domain (foo.com, dns.foo.com) 	 (foo.com, relay1.bar.foo.com, CNAME) Type=MX of value is the canonical name of erver a mail server associated with an alias name s in (foo.com, mail.bar.foo.com, MX) m, NS)

Inserting resource records into DNS

Let's assume that Alice just created a new startup company "Network Utopia". She wants to register domain name networkuptopia.com.

1- She needs to contact a registrar (A registrar accredited by Internet Corporation for Assigned Names and Numbers (ICANN) verifies the uniqueness of the domain name, enters it into the database and collects a small fee for its services.)

2- She needs to provide registrar with names and IP addresses of her primary and secondary authoritative name servers.

3-Registrar inserts two RRs (NS and A) into the TLD com server:

- (networkutopia.com, dns1.networkutopia.com, NS) •
- (dns1.networkutopia.com, 212.212.212.1, A)

4 -She also has to make sure that Type A record for web server www.networkuptopia.com and Type MX record for her mail server mail.networkutopia.com are entered into her authoritative DNS servers.

Now suppose Bob views www.networkutopia.com. His host will first send a DNS query to his local DNS server, which will then contact a TLD com server. This TLD com server contains the NS and A RRs listed above and sends a reply back to Bob's local DNS server. This local DNS server then sends a query to 212.212.212.1, asking the type A record. This record provides the IP address, say 212.212.71.4, to Bob's host and now browser can initiate a TCP connection.

Topic 62: Introduction to Wide Area Network

In this topic, we describe what a wide area network (WAN) is.

A wide-area network (WAN) defines a type of a network, or part of a network, in which the devices are relatively far apart. A WAN is a network, or part of a network, for which the cabling must pass outside the property of one company. The distance might only be a few miles, or it might be thousands of miles! We next show a WAN.



Figure 62.1: Wide Area Network

You can run an Ethernet cross-over cable between two devices and the two devices can communicate with Ethernet. That works well, but can we use it for longer distances? If two routers in two cities need to forward packets to each other, they need some sort of physical medium over which to send the packets.



When R1 sends out an electrical signal over the cable, R2 needs to receive that same signal on the wires that it expects to receive data. Likewise, R1 needs to receive what R2 transmits. Because the physical interfaces on the routers use the same pins to transmit, the cable connects the twisted pair used for transmitting by R1 over to R2's receive pins, and vice versa. In short, all that the two routers need is a cable between them, with transmit and receive pairs of wires, so that they can send and receive anytime they want.

In the previous example, there are two problems stop you from using Ethernet:

• 1-You are not legally allowed to run a cable between Atlanta and Cincinnati.

• 2-Those who can run the cable namely, the telephone companies (telcos) do not lease or sell 450-mile Ethernet cross-over cables.

Topic 63: Different Aspects of WAN Link

In this topic, we describe different aspects of a WAN Link.

To connect two routers located let's say at Atlanta and Cincinnati (450 miles), a telephone company (telco) should offer:

1- A service between two routers that acts like a cable with four wires (two pairs) in it.

2- When a router sends on one pair, that pair is crossed to the other pair before it gets to the other end cable. That way, when one router sends on one pair, the other router receives on the other pair.

3- Your routers use a serial interface because the service requires to physically encode data a little differently than on an Ethernet.

Result is that two routers can send and receive data to and from each other. Essentially, a telco can lease you a 4-wire cable, or 4-wire circuit, between two points. Telcos have already run cables between almost every town and city. Telcos have offices, called central offices (COs), almost everywhere. In addition, telcos have the right-of-way, which is the legal right to dig up roads and put cabling in the ground.



Figure 63.1: Aspects of WAN Link

Switches in the CO are called telco switches, phone switches, or WAN switches. Telco needs to ensure path between Atlanta and Cincinnati:

1- A cable from office building in Cincinnati where other router sits and telco CO in Cincinnati.

2- Electrical signals between Atlanta CO and Cincinnati CO.



Topic 64: A Cross-over Cable versus Leased Circuit

In this topic, we describe differences between a cross-over cable and a leased circuit.

If you install an Ethernet cross-over cable between two routers by plugging the cable into Ethernet interface on the routers, the two routers can forward Ethernet frames to each other. Ethernet can support transmission speeds of 10 Mbps, 100 Mbps, 1000 Mbps (1 gigabit/second), and even 10 Gbps. If you install a WAN link between two routers, a telco creates something similar to a cross-over cable with some differences. With WAN links, many different transmission speeds are supported. Typically, WAN links use speeds that are multiples of 64 Kbps. For example, a telco offers a service called T1 line or T1 circuit whose speed is 1.5 Mbps, which is 24 times 64 Kbps. A telco can also offer links in multiples of 1.544 Mbps. You can also order even higher speeds that generally come in multiples of 51.84 Mbps.

Key points about speed of WAN links

Many different speeds are possible. You specify the speed when you order the leased line. You need to configure the routers to use the right speed.

How to control the WAN link speed

Serial links run at different speeds, and must be preconfigured. The channel service unit/data service unit (CSU/DSU) of a serial interface controls the speed. The CSU/DSU functions can be done with an external device or as a function of the serial interface card on the router. If you use a CSU/DSU that sits outside the router, you must configure the speed on that external device. If CSU/DSU is built into the router, you must configure the speed of the WAN link on the router.



A WAN Link Installation Plan

1. Contract with the phone company to provide a leased line at a certain speed.

2. Install a router at each site, near where the telco will run its cable.

3. Install an external CSU/DSU near the routers at each site, if you didn't buy routers with internal CSU/DSUs.

4. Configure the CSU/DSUs with the correct speed.

5. After the phone company runs the cables, install the cables into the CSU/DSU (external) or serial interface of the router (internal CSU/DSU) at each site.

Topic 65: Routers and WANS

This topic describes how WANs and Routers work together.

Routers use WAN links when they need to forward IP packets to a subnet at a remote site. End user devices at a company's office site use an Ethernet NIC and never directly connect to a WAN link. To send traffic to an IP host at another site, end user devices send the data to router. The router forwards the IP packet to another router at the other site, which then forwards the packet to the other IP host.



Figure 65.1: WANs and Routers work together

WAN data link framing

Let's assume that Hannah sends a packet to the www.example.com web server at IP address 150.1.3.3. The IP packet arrives to her default gateway namely R1. As R1 receives the Ethernet frame, it checks to see whether errors occurred. If no errors occurred, R1 extracts the IP destination address 150.1.3.3 and find a match in his routing table.



Figure 65.2: WAN data link framing

R1 needs to encapsulate the IP packet inside a data link layer frame. Two popular data link layer protocols for point-to-point WAN links are high-level data link control (HDLC) and Point-to-Point Protocol (PPP).



Figure 65.3: High-level data link control (HDLC) and Point-to-Point Protocol (PPP)

R1 needs to encapsulate the IP packet inside a data link layer frame. Two popular data link layer protocols for point-to-point WAN links are high-level data link control (HDLC) and Point-to-Point Protocol (PPP).

WAN data link addressing

Inside the HDLC and PPP header, there's only one address field, and it is 1 byte long. When R1 sends anything on this point-to-point WAN link, the only device that could possibly get the data is R2. So, although the address field exists, it doesn't really matter.

Differences bet. HDLC and PPP: HDLC was defined by the ITU in the 1970s. PPP was defined in RFC 1661 during the 1990s. PPP has more advanced features. Cisco uses a nonstandard version of HDLC. Cisco also conforms to the standard for PPP.

Topic 66: Frame Relay

In this topic, we explain the basics of Frame Relay.

Frame Relay is a wide area network (WAN) technology that uses one physical WAN link connected to each site, while allowing each site to send data to each other site. When you want to build a network to connect multiple remote sites, you can use a lot of serial links. Frame Relay requires less work, less new hardware and is more cost-effective solution compared to leased lines. A Frame Relay network acts like a big WAN switch, with routers connecting to it. To send data to another router, the sending router just needs to send a frame with the right address in it.

To make Frame Relay work, each router needs a physical cable between itself and a device called a Frame Relay switch. A Frame Relay switch is the equipment that understands Frame Relay and can forward traffic based on Frame Relay protocols. The telco uses Frame Relay switches in its local central offices (COs) that together switches the data to the correct sites. When a router physically connects to a Frame Relay switch, it is connecting to a Frame Relay service. The company that sells Frame Relay services is called a Frame Relay service provider. Often, a Frame Relay service

provider is also a telco. In most cases, that company works with other telcos to create the Frame Relay network.



Figure 66.1: Physical Parts of Frame Relay

Although the cabling and channel service unit/data service units (CSU/DSU) are the same as with a leased line, the telco does something different in the CO: It connects the cable to a Frame Relay switch. The provider's collective set of Frame Relay switches, along with the other equipment between them, form that provider's Frame Relay network. Frame Relay is a set of protocol specifications, all matching the functions of OSI Layer 2; data-link layer.

Topic 67: Frame Relay Switching

This topic explains the working of Frame Relay switching to multiple remote sites.

When a telco sells you a Frame Relay service, it forwards Frame Relay frames sent by one router to another router. Before a router can send a packet, it must add the correct data link header and trailer to the packet. Each Frame Relay header contains a single address field called a data-link connection identifier (DLCI). It is a 10-bit number, usually written as a decimal number between 0 and 1023. Each Frame Relay switch forwards the frame, based on the DLCI, through the network, until it gets to the router on the other side. Frame Relay switches must be configured to know where to forward frames with particular DLCIs in their headers. They cannot learn addresses automatically.



Figure 67.1: Cabling with Internal and External CSU/DSUs

To see the real advantage of Frame Relay over leased lines, let's look at an example with three Frame Relay sites. We further assume router R1 (Atlanta) wants to send data to both R2 (Cincinnati) and R3 (Boston). With Frame Relay, a router can use its single physical access link to forward traffic to multiple remote routers. This is shown next.



Figure 67.2: Three Leased Lines to Connect Three Routers

Topic 68: Virtual Circuits

In this topic, we explain the concept of virtual circuits.

Let's assume that there are three sites using Frame Relay network and frames are going from one site (R1) to the other two sites (R2 and R3).



Figure 68.1: Frame Relay Switching to Multiple Remote Sites

In Frame Relay lingo, the ability for R1 to send data to R2 over the Frame Relay network is called a virtual circuit (VC). A VC is like a leased circuit, in that exactly two devices can send and receive data using it. It is called virtual to contrast it with a physical leased circuit. As a VC is typically predefined to always be there, VCs are often called permanent virtual circuits (PVCs). It is shown next.



Figure 68.2: Frame Relay PVC Concepts

The service provider preconfigures all the required details of a PVC. Configuring the switches to forward frames with DLCI 102 to Cincinnati and frames with DLCI 103 to Boston. In the previous network, there is no PVC between R2 and R3. The choice of which sites need to have a PVC depends on where the network engineers think that traffic needs to flow in the network. When routers use Frame Relay, and there is a PVC between each pair of routers, the PVCs are in a full mesh. When not all routers have a PVC, it's called a partial mesh. A full mesh is shown next.



Figure 68.3: Frame Relay Full Mesh

Topic 69: Point-to-Point WANs & Frame Relay

In this topic, we compare point-to-point WANs and Frame Relay.

Frame Relay is considered to be faster, cheaper, and better than using leased lines for a WAN. Point-to-point WANs tend to require more hardware than does the equivalent network built with Frame Relay. Let's consider what the telco does to create three leased circuits. A router requires two serial interfaces and two separate channel service/data service units (CSU/DSU). The telco has to install two cables between the local CO and the office building in Atlanta where R1 resides.



Figure 69.1: Three Leased Lines to Connect Three Routers

Next, we show the equivalent Frame Relay network for the three sites. One cable from each router is needed to the telco, one serial interface, and one CSU/DSU at each router.



Figure 69.2: Typical Frame Relay Network with Three Sites

What happens when the company grows to 10 sites or 100 sites and employs pointto-point WANS? For each point-to-point line, R1 will need a separate physical serial interface and a separate CSU/DSU. This requires a lot of extra money for router and CSU/DSU hardware. The telco has to run many more cables and more cost.

Topic 70: Routing over PVCs

This topic explains routing over permanent virtual circuits (PVCs).

Assume an internetwork that uses three routers at three sites. Each has an access link to a local CO, and there is no PVC between R2 and R3, making the network a partial mesh of PVCs. The Frame Relay provider (the telco) tells the customer to configure R1 so that it uses data-link connection identifier (DLCI) 102 to send frames to R2, and DLCI 103 to send frames to R3.



Figure 70.1: Routing over PVCs

Now assume Hannah opens a browser to connect to the www.example.com. After using DNS to discover that www.example.com's IP address is 150.1.3.3, Hannah sends a packet to 150.1.3.3. Because 150.1.3.3 is in a different subnet, Hannah's PC sends the packet to her default gateway namely, R1. When R1 receives the Ethernet frame, it checks to see if errors occurred. If no error, then R1 performs Frame Relay encapsulation. Frame Relay header holds the 10-bit-long DLCI field that identifies the PVC. The trailer has a frame check sequence (FCS).



Figure 70.2: Forwarding an IP Packet over a Frame Relay PVC

When R1 looks at his routing table for subnet 150.1.3.0, it lists next-hop router of 150.1.2.2, which is R2's IP address, and outgoing interface S0. It does not list which DLCI to use to get to 150.1.2.2. As soon as the PVC starts working, R2 announces its IP address to R1, using the VC between the two routers. R1 also announces its IP address to R2, using that same VC. In this way, both routers learn the other router's IP address that is used on that VC. The message used to announce the IP address and DLCI is called an Inverse ARP message.

Topic 71: The Internet: A Large IP Network

This topic explains why connecting to the Internet provides a path for IP packets to be sent almost anywhere.

A company cannot run a cable between two far-away buildings for legal and practical reasons. A telco already has the ability to let you connect to its network physically,

and then deliver the data over its network. With WAN links, the telco allows two sites to communicate. With Frame Relay, many sites can communicate, but they all must use the same Frame Relay service from the same telco. The company can connect to the Internet with a WAN link or Frame Relay. With that one physical connection, that company gains access to a huge number of people who also have access to the Internet. Next, we show an example depicting reaching the world through one connection to the Internet.



Figure 71.1: Reaching the World through One Connection to the Internet

An Internet service provider (ISP) creates an IP network over which IP packets can be forwarded, similar to the way a telco creates a network over which bits (for WAN links) and Frame Relay frames (for Frame Relay) can be forwarded. Some ISPs are also telcos. The Internet consists of three major components:

- The IP networks that ISPs create.
- The enterprise IP networks that attach to one or more ISPs.
- The individuals who connect their computers to ISPs.

WAN Type	Generic Name of Provider	Physical Connectivity	Description of Service
WAN link (leased line)	Telco	4-wire cable from customer to telco CO at each site	Telco forwards bits between two sites
Frame Relay	Frame Relay service provider	4-wire cable from customer to telco CO at each site	Provider forwards frames, based on DLCI
Internet	Internet service provider	Either WAN link or Frame Relay bet. customer and ISP	ISP routes IP packets to any IP host on the Internet

Table 71.1: Comparisons of WAN Links, Frame Relay and the Internet

Topic 72: Using a Phone Line for Data

In this topic, we describe how a modem can be used to transmit data from home.

A telephone includes a microphone, which converts sound waves into an analog electrical signal. It is carried by a local loop (phone line between a house and a telco CO). The telco can send the electrical signal between one phone and another. On the receiving side, the phone converts the electrical signal back to sound waves using a speaker. The characteristics of the analog signal are:

- Frequency: the number of times a signal repeats itself, from peak to peak, in one second.
- Amplitude: how strong the signal is.



Figure 72.1: Analog Electrical Signal: Frequency and Amplitude

The sounds that your voice makes happen to be a continuously changing sound wave, so analog electrical signals that continuously change work well for voice. Modems allow two computers to send and receive a serial stream of bits over an analog phone circuit, with no physical changes required on the local loop between a residence and the telco CO. As the telephone switch in the CO expects to send and receive analog voice signals over the local loop, modems simply send an analog signal to the PSTN and expect to receive an analog signal from the PSTN. Instead of voice, a modem converts a string of binary digits on a computer into a representative analog electrical signal.



Figure 72.2: Comparing a Phone to a Modem

Modems encode a binary 0 or 1 onto the analog signal by varying frequency or amplitude. Changing these characteristics of the analog signal is referred to as modulation. Basic Operation of Modems over PSTN are shown next.



Figure 72.3: Basic Operation of Modems over PSTN

In previous example, frequency changes every .1 seconds, PC can send 10 bits/sec. If a modems sends at 9600 bps, the receiving modem needs to sample the incoming analog signal every 1/9600 of a second. The details of modem relate to OSI Layer 1, the physical layer.

Topic 73: Digital Subscriber Line (DSL)

In this topic, we describe the usage of DSL for Internet.

Modems use the phone line to make what the telco thinks is a voice call, so you cannot make another call while you are surfing the Internet. Another big negative is the speed. Digital subscriber line (DSL) is an alternative technology for sending and receiving data to and from an ISP, using the same old phone line, but running at much faster speeds. Also, you can make phones calls and surf (send IP packets) at

the same time. But it is a little more expensive. DSL allows the same old analog voice signal to be sent over the line by a phone. At the same time, DSL allows a separate digital signal to go over the same phone line. The telephone generates analog signals ranging from 0 to 4000 Hz. Most speech ranges from 300 to 3300 Hz. The digital DSL signal uses frequencies higher than 4000 Hz – do not interfere with speech/hearing. DSL uses the same local loop wiring that is already installed between the CO and your house, but now the CO connects the local loop wiring to a device called a DSL access multiplexer (DSLAM). The DSLAM splits out the digital signal and analog signal from the local loop. The DSLAM gives the analog voice signal to a telephone switch while forwards the data traffic to a router.



Figure 73.1: DSL Connection from the Home

Some DSL variants allow the local loop to be much longer than some other DSL variants, whereas others only allow for a shorter local loop. For a shorter local loop, transmission rates are much higher – design tradeoff. The speed at which data goes from the Internet to the home is faster than the other direction a feature called asymmetric transmission rates. More bandwidth is needed for downloading.

Figures and Material used for this topic have been adapted from Kurose and Ross's book with the title *"Computer Networking: A Top-down Approach"*, 6th Edition, 2013.

Topic 74: Sending Data without a Phone Line

In this topic, we describe how TV cable networks can be used for the Internet.

Cable Internet access makes use of the cable television company's existing cable television infrastructure. A residence obtains cable Internet access from the same

company that provides its cable television. A cable head end broad casts television channels.

data, TV transmitted at different frequencies over shared cable distribution network cable headend cable distribution network (simplified)

Cable Network Architecture: Overview

Figure 74.1: A hybrid fiber-coaxial access network

Fiber optics connects the cable head end to neighborhood-level junctions, from which traditional coaxial cable is then used to reach individual houses. This is called Hybrid fiber coax (HFC) and is shown next in detail.



Figure 74.2: A hybrid fiber-coaxial access network

Frequency division multiplexing: different channels transmitted in different frequency bands.



Figure 74.3: A typical home network

Topic 75: Introduction to AAA Security Model

In this topic, we list and describe the three components of the AAA security model.

AAA stands for authentication, authorization, and accounting. The AAA model refers to three areas of security. All three relate to issues regarding an individual device or user. Before you can use most servers, you need to have the right to use that server. A username and a password are required. **Authentication** is a five-step process.

- 1. Fred requests a web page.
- 2. The web server replies, asking for a username, password.
- 3. Fred types in and sends his username, password to server.
- 4. The web server checks its list of usernames and verifies that the password is correct.
- 5. If the username and password are correct, the web page returns the contents of the web page.

The working of basic authentication process is shown next:



Figure 75.1: Basic Authentication Using a Username and Password

Username, password can be listed on a server called authentication server. Application servers query authentication server. Kerberos protocol runs between application and authentication servers. This is shown next.



Figure 75.2: Basic Authentication Using a Username and Password

The second "A" in AAA is for **Authorization**. It refers to the process of figuring out what a particular user is allowed to do.



Figure 75.3: Fred Can't See Others' Payroll Information

The last "A" in AAA is **accounting**. The same servers that perform authentication and authorization services can keep a record of each request to authenticate or authorize a user. Accounting allows to record and report when users type the wrong password. Several consecutive attempts to connect to a server with an invalid password each time signals someone is trying to guess Fred's password. Accounting features can record individual attempts, generate reports, and notify personnel if too many invalid attempts happen in a short timeframe.

Topic 76: Password Authentication Protocol

In this topic, we describe the working of password authentication protocol (PAP).

Unlike those people who are attached to a LAN in a physically secure office building, people who use modems or DSL to connect to the Internet from home have to go through an additional authentication step. ISPs authenticate users before they can even use the network. This helps ISPs to determine whether a customer has paid his bill or not. Password Authentication Protocol (PAP) and Challenge Handshake Authentication Protocol (CHAP) are used by ISPs. Both are part of Point-to-Point Protocol (PPP) — a data link protocol. Let's assume that Fred dials an ISP with a modem, PAP is used between himself and the ISP router. The following sequence is used:

1-Fred uses PAP to send a username, password.

2-The router sends a request to an authentication server using RADIUS protocol messages.

3-The authentication server checks the username, password.

4-The authentication server confirms that Fred is authentic using RADIUS.

5-The router uses PAP to confirm that Fred is allowed to use the Internet.



Figure 76.1: Basics of PAP

TACACS+ is a popular proprietary protocol that Cisco developed before RADIUS existed and can be used in place of RADIUS. Also, CHAP can be used instead of PAP. PAP sends the username and password in clear-text. Anyone with the right tools can actually read your username and password. To protect against password theft, CHAP does not send the password as clear-text. Process of CHAP starts as soon as a modem calls a phone number at the ISP or DSL modem is powered on. CHAP uses six steps to authenticate user.

1- Router generates a random number and sends it in a CHAP message to the PC.

2-PC runs a math function with the random number and the password typed by the user as input.

3-The PC sends the results of the function called a message digest back to the router.

4-The router sends the username, the random number, and the message digest to the AAA server.

5a-The AAA server uses the same math that the client used in Step 2, with the same random number, plus the password associated with that username in the AAA user database.

5b- The result is another message digest.

5c-If the message digest calculated by the AAA server matches the one calculated by the PC, the password that the user typed must be the right one.

6. The AAA server tells the router that the user is authentic; the router tells the PC, and life goes on.

The password was never passed through a network connection during this procedure. This is shown next.



Figure 76.2: Hashing Passwords to Create a Message Digest

Topic 77: Virtual Private Network (VPN)

This topic explains benefits of using a virtual private network (VPN).

Encryption allows a computer to apply a mathematical formula to data, sending the results of the mathematical function over the network. The data just looks like a bunch of random bits. The computer receiving the data can then re-create the original data by decrypting the data. To decrypt the data, you need a secret password called an encryption key. These days, it is common for users to encrypt data before sending it over the Internet. Instead calling it encryption; it is called a virtual private network (VPN). The packets go across the Internet, it is still a physical public network. VPNs create a private network, but they do so logically, or virtually.

Let's assume Barney works for a company. The enterprise network at Barney's company is a private network, with all the components inside privately controlled office space. The Internet is public. For Barney to use the VPN, he must encrypt the packet as he creates it. To do this, Barney needs to have VPN software installed on his computer. He also needs to know the encryption key to use. A VPN device inside the corporate network is called a VPN concentrator, which decrypts packets received from Barney and others. This is shown next.



Figure 77.1: Encrypting IP Packets for a VPN

Topic 78: Enterprise Network and the Internet

In this topic, we list the traffic that should and shouldn't be allowed between an enterprise network and the Internet. When you connect an enterprise network to the Internet, one of the first things you must decide about is what you want to allow to pass to and from the Internet, and what you don't. Assume an enterprise network 'Internal IP Network'. There is an internal web server that has stuff only appropriate for employees who work for Fredsco. The external web server is meant for external users, but internal clients will also want to browse that web server.



Figure 78.1: An Enterprise Network Connecting to the Internet

To secure Fredsco's network, two things have to be kept in mind: Between which two hosts do packets need to flow? Which host begins that communication? To secure Fredsco's network, two things have to be kept in mind: Between which two hosts do packets need to flow? Which host begins that communication? Typical types of traffic allowed are shown next.



Figure 78.2: Typical Types of Traffic Between an Enterprise and the Internet

The bad guys on the Internet who are trying to get into Fredsco's network should be prevented. Traffic that is typically not allowed is shown next.



Figure 78.3: Traffic That's Typically Not Allowed

Topic 79: Firewalls, Demilitarized Zone, IDS

In this topic, we describe the use of firewalls, demilitarized zone (DMZ) and intrusion detection systems (IDS).

A network engineer configures the firewall with a set of rules that tells it what's legal and what isn't. Then the firewall allows some packets to pass through it and discards others to enforce the rules. The firewall needs to be in the path that is used for forwarding packets to and from the Internet. Cisco's firewall product is called a PIX firewall.



Figure 79.1: Putting Up a Wall Between the Dangerous Folks and Your Network

As a firewall watches the traffic entering the network, it knows the nature of the traffic that is allowed to flow through it. Also, it recognizes when a host is initiating a new flow. A host who is initiating a new flow can be recognized by looking at: The first TCP segment used to create a TCP connection SYN flag bit =1, and source IP address of the packet.



Figure 79.2: TCP Connections and Well-Known Ports

The scenario depicting the allowance of the TCP Connection is shown next.



Figure 79.3: Allowing the TCP Connection from Figure 79.2

The firewall uses a similar logic to stop packets that should not be allowed. This is shown next.



Figure 79.4: Disallowing a TCP Connection to an Inside Web Server

Firewalls are like routers. They forward packets based on destination IP address. They have at least two physical interfaces. They can have more than two interfaces. Outside interface connects to Internet. Inside interface connects to internal network.

A third interface connected to a LAN called a demilitarized zone (DMZ), somewhere bet inside & outside interfaces. With a DMZ, Internet-accessible servers can be placed on a different LAN. A stronger firewall rule: No TCP connections can be initiated from outside to the inside. The only flows allowed are for servers in the DMZ.



Figure 79.5: A Safe but Potentially Risky Place: The DMZ

Intrusion detection systems (IDSs): watch packets that a firewall allows through. They look for things in the packets to determine if someone is cheating firewall and do bad things to servers in network. Some IDS devices sit in the network, watching packets that pass over a LAN and are called network-based IDSs. Those IDS softwares that sit on the servers are called host-based IDSs. A network-based IDS is shown next.



Figure 79.6: Watching for Patterns with a Network-Based IDS

Handouts for Part 2 (Topics 80 to 120) CS206: Introduction to Network Design and Analysis

Figures and Material used for this topic have been adapted from "Wireshark User's Guide", U. Lamping, R. Sharpe, and E. Warnicke, 2014, https://www.wireshark.org/docs/wsug_html/.

Topic 80: Introduction to Wireshark

This topic provides introduction to Wireshark.

Wireshark is a network packet analyzer which captures network packets and displays that packet data as detailed as possible. Wireshark is a free open source software program available at <u>www.wireshark.org</u>. G. Combs is the original creator of Ethereal (Wireshark's development name prior to May 2006). As an open source project, it's source code is open.

Intended Purposes: When run on a host connected to a wired or wireless network, Wireshark captures and decodes the network frames. People use it to learn network protocol internals. Network administrators use it to troubleshoot network problems. Network security engineers use it to examine security problems. Developers use it to debug protocol implementations.

Wireshark's Features

- 1- Available for UNIX and Windows.
- 2- Capture live packet data.
- 3- Display packets with very detailed protocol information.
- 4- Filter packets on many criteria.
- 5- Save packet data captured.

Wireshark's Installation

Get Wireshark installer from: www.wireshark.org/download.html and execute it.



Figure 80.1: Wireshark's Installation wizard

Installation Components

- Wireshark The network protocol analyzer.
- TShark A command-line network protocol analyzer.
- **Plugins & Extensions** Extras for the Wireshark and TShark dissection engines.
- Tools Additional command line tools to work with capture files
- User's Guide



Figure 80.2: choose components

🧲 Wire	eshark 1.12.3 (64-bit) Setup 🛛 – 🗖 🗙
Choose Components Choose which features of Wires	shark 1. 12.3 (64-bit) you want to install.
The following components are a	available for installation.
Select components to install:	···· ✔ Wreshark ··· ✔ TShark @·· ✔ Plugins / Extensions @· ✔ Tools ··· ✔ User's Guide
Space required: 105.5MB	Description Position your mouse over a component to see its description,
Wireshark Installer (tm) ————	< Back Next > Cancel

Figure 80.3: choose components

Installing WinPcap: With WinPcap installed you would be able to capture live network traffic.

stall WinPcap?	in a patrical data. Chardd WinDena ha installad?
inecap is required to capture i	
Currently installed WinPcap ve	ersion
WinPcap is currently not insta	alled
Install	
✓ Install WinPcap 4.1.3	
✓ Install WinPcap 4.1.3 (Use Add/Remove Program	ms first to uninstall any undetected old WinPcap versions)
✓ Install WinPcap 4.1.3 (Use Add/Remove Program	ms first to uninstall any undetected old WinPcap versions)
✓ Install WinPcap 4.1.3 (Use Add/Remove Program)	ms first to uninstall any undetected old WinPcap versions)
 Install WinPcap 4.1.3 (Use Add/Remove Program) 	ms first to uninstall any undetected old WinPcap versions)
 Install WinPcap 4.1.3 (Use Add/Remove Program) 	ms first to uninstall any undetected old WinPcap versions)
✓ Install WinPcap 4.1.3 (Use Add/Remove Progra What is WinPcap?	ms first to uninstall any undetected old WinPcap versions)
Install WinPcap 4.1.3 (Use Add/Remove Program) What is WinPcap?	ms first to uninstall any undetected old WinPcap versions)

Figure 80.4: Installing WinPcap



Figure 80.5: Installing WinPcap

The main window is shown next.



Figure 80.6: The Main window

Area 1 shows the Title bar — trace file name. Area 2 represents Main menu — standard menu. Area 3 is the Main toolbar — provides quick access to frequently used items from menu. Area 4 depicts Display filter area — reduce amount of traffic you see. Area 5 shows Packet List pane — summary of frames. Area 6 is the Packet Details pane — dissected frames. 7. Packet Bytes pane — hex/ASCII details. Area 8 represents Status Bar — access to the Expert, annotations, packet counts, and profiles.

Figures and Material used for Part2 (from Topics 81 to 172) have been adapted from "Troubleshooting with Wireshark: Locate the Source of Performance Problems", 2014, by L. Chappell.

Topic 81: Five Common Network Problems

This topic discusses 5 common Network problems.

TCP Connection Refused by Server

A service is not running on a server or perhaps a firewall is preventing the connection. TCP connection refusals may also be due to a closed port. Possible Symptoms: Client's SYN followed by Reset (RST)/ACK. Client SYN followed by an ICMP Destination Unreachable/Port Unreachable response from a host-based firewall.

Connection Blocked by a Host-Based or Network Firewall

Ideally, hosts would not even attempt to communicate with firewalled resources. Such an attempt could be due to a misconfiguration, malware, malicious user, or other issue. Possible Symptoms: No response to a SYN packet, RST/ACK response to a SYN packet.

Slow Application at Server

The good news is that the server did not refuse to provide a desired service. The bad news is that the server is slow. This may be due to: 1- lack of processing power at the server, 2-a poorly behaving application, or 3-even, in a multi-tiered architecture, a slow upstream server that actually provides the data. Possible Symptoms: TCP-based app: Large delay between the server ACK to a client request and the response data. UDP-based app: Large delay between a request and the response data.

Slow Load of Remote Content

Many networks are designed in a multi-tiered fashion. For example, consider a client that sends a request to Server 1. That server may need to obtain information from Servers 2 through 9 before answering the client. We need to capture this multi-tiered traffic to determine which server is actually responding slowly. Possible Symptoms: TCP-based app: Large delay between the server ACK to a client request and the response data. UDP-based app: Large delay between a request and the response data.

Topic 82: Next 5 Common Network Problems

This topic describes next 5 common Network problems.

Server Application Fault

The server is up and running, but not responding to requests. The server responds to a SYN with a SYN/ACK, but when the client makes a request, no response is received. When the TCP Retransmission Time Out is reached, client retransmits request. If no ACK is received, it continues to retransmit request using an exponential backoff time until it finally gives up and sends a TCP Reset. Possible Symptoms: Retransmissions.

Content Redirection

Upon arriving at a store to buy something, we might be told that the store does not have the item in the stock; please go to another store to get it. If an application supports redirection (such as HTTP) and the target knows where the information actually resides, it may send you a redirection response. Possible Symptoms: In HTTP communications, response codes 300-399. A sudden "name resolution process" and traffic to another host.

TCP Receive Buffer Full

When advertised TCP receive buffer value drops to zero, data transfer will stop. A low Window Size value can also cause data flow to stop. The only recovery is a Window Update. Possible Symptoms: Unusually high TCP delay before a Window Update and resumption of data flow.

TCP Send Buffer Full:

A limited send buffer space can negatively impact network performance even if network can handle a high data transfer rate and the receiver has plenty of buffer space. Possible Symptoms: High TCP delay for no apparent reason.

Altered TCP Attributes along a Path

Routers and other interconnecting devices can bring havoc on a network if they change the attributes of a TCP connection as they forward the traffic. Possible Symptoms: Low TCP calculated window size and significant delays before Window Update packets.

Topic 83: Next 6 Common Network Problems

In this topic, we provide description of next 6 Common Network Problems.

Mismatched TCP Parameters across a Proxy Device

Connection parameters on one side of a proxy device do not match the connection parameters offered in connection on other side of proxy device. Possible Symptoms: Delays in data forwarded through proxy (proxy queuing).

Routing Loops

Routing loops occur when a packet is routed back onto a network over and over again. Possible Symptoms: Identical packets listed but no TCP Retransmission indications (they are not identical packets—their TTL value decrements).

Weak Signal (WLAN)

If a wireless signal degrades substantially, it may not be interpreted properly when captured. Possible Symptoms: Low Signal Strength value.

Asymmetric Routing

This is a situation when traffic flowing from Host A to Host B flows along a different path than the traffic flowing from Host B to Host A. This can be a problem, if there are devices that must see every packet in order to function. For example, network path must be symmetrical from a client to a proxy host. An IDS box must see every packet. Possible Symptoms: ACKed segment that was not captured.

Packet Loss

Typically, this occurs at an interconnecting device such as a switch, a router, a NAT device, or a network firewall. If the number of packets dropped is small and the recovery process is quick, the packet loss may go unnoticed. If many sequential packets are lost, however, users will likely feel the impact and complain. Possible Symptoms: Fast Retransmission. Duplicate ACKs.

High Path Latency

A single low speed (high delay) link along a path or the delay between geographically disbursed peers can inject a level of path latency that affects performance. Possible Symptoms: Large delays between the outbound SYN and the inbound SYN/ACK of a TCP handshake.
Topic 84: Next 4 Common Network Problems

This topic discusses Next 4 Common Network Problems.

Lousy Routing Path:

When construction occurs in a major city, traffic is a nightmare as drivers are rerouted along less efficient and less direct routes. This is also true of network traffic. If a target is 10 blocks away and yet for some reason the packets must travel through 17 routers to get there, performance may be unacceptable. **Possible Symptoms:** Large delays between the outbound SYN and the inbound SYN/ACK of a TCP handshake.

Bandwidth Throttling

Transmitting data along a bandwidth throttling link is like driving a car during rush hour. You move along bumper-to-bumper at speeds below your lowest speedometer indications.

Delayed ACK

It is defined in RFC 1122 as a method to "increase efficiency in both the Internet and the hosts by sending fewer than one ACK segment per data segment received." This delay can cause issues for the host that is sending data if it times out waiting for an ACK or it cannot send another data packet until an ACK is received. Possible Symptoms: 200ms delay before ACK packets.

Queued Packets (Overloaded Router)

TCP peers and UDP-based applications may detect sudden queue delays and think packets have been lost. This may be caused by an overloaded router or perhaps prioritization at a router (e.g., video streaming first, email traffic last). Possible Symptoms: Decrease in throughput followed by equal increase in throughput.

Topic 85: Next 5 Common Network Problems

This topic studies Next 5 Common Network Problems.

Route Redirections

When a host sends packets to one local router when another local router exists with a preferred path, a route redirection may take place. The receiving router may respond with an ICMP packet that includes IP address of the recommended router. Rare - we see only one router on the local network. Possible Symptoms: ICMP Type 5 packets with either Code 0 (Redirect for Host) or Code 1 (Redirect for Network) indicate route redirection is taking place.

Broadcast or Multicast Storms

Broadcasts are typically not forwarded so you should be capturing on and dealing with hosts on a single network. Multicasts can be forwarded through an internetwork therefore they can cause a greater problem if something goes wrong and they storm the network. Possible Symptoms: High rate of packets addressed to the all nets broadcast (255.255.255.255). High rate of packets addressed to a multicast address (224-239.x.x.x).

Switch Loop

Older switches that have not been restarted or rechecked every once in a while may begin to malfunction. If there is a loop—network will become overwhelmed with looped packets. Possible Symptoms: High rate of identical packets.

Virus/Malware on Network Hosts

When a compromised host begins performing port scans on other network hosts or it begins broadcasting discovery packets, the overhead may be felt. Possible Symptoms: Local broadcasts. Unusual internal targets (such as a local client trying to connect to the Accounting server).

Network Name Resolution

This is imperative to connecting to a target. A user may receive an application error such as "Server not found" as in the case of a DNS Name Error when browsing. Possible Symptoms: No response to a name query. An error response to a name query (such as a DNS Name Error or Server Error).

Topic 86: Next 8 Common Network Problems

This topic studies Next 8 Common Network Problems.

Network Address Resolution

Incorrect subnet addressing can cause a client to send packets that are destined to a local device to a router. Possible Symptoms: ARP Requests are sent for remote targets. Traffic destined to local hosts are sent to the router.

Hardware Address Resolution

This allows obtaining MAC address of a local target or local router. Network address resolution problems may cause a host to think local devices are remote (or vice versa). Possible Symptoms: No response to an ARP query for a local host. No response to an ARP query for a local router.

No Support for Selective ACK

When packet loss is detected by a receiving TCP host that supports SACK, that host can acknowledge data received after the missing packet(s). Without SACK, the first packet lost and every subsequent data packet will be retransmitted even when those subsequent packets were received successfully. Possible Symptoms: No SACK Option in the TCP header of a SYN packet. No SACK Option in the TCP header of a SYN/ACK packet when there was a SACK Option in the TCP header of a SYN packet.

No Support for Window Scaling

Window Scaling is used to increase the advertised TCP receive buffer size past the 65,535-byte limit caused by the 2-byte Window Size field. During the TCP handshake, peers indicate that they support Window Scaling and provide a Window Scale Shift Count. Reduces the delays caused by a low window or Zero Window condition. Possible Symptoms: No Window Scaling Option in TCP header of a SYN/ACK packet when there was a Window Scaling Option in TCP header of a SYN packet.

Client Misconfiguration

A misconfigured client may have the wrong DNS address, the wrong router address, incorrect port numbers defined in a services file, or other problems. Possible Symptoms: The client sends traffic to the wrong target. The client receives service refusals or no answer. Numerous other symptoms may appear depending on the misconfiguration.

Low Packet Size/Low MTU Size

Amount of data that can be carried in a frame is limited by MTU, MSS, or even an inefficient application. Performance pains when sending large files. Possible Symptoms: The MSS value defined in the TCP SYN or SYN/ACK packet is illogically small.

TCP Port Number Reuse

Port numbers can be reused without any problem if previous TCP connection is terminated. Otherwise, the new connection is refused. Possible Symptoms: TCP RST sent in response to a SYN packet to a server port that is known to be open.

Slow Application

Slow applications can be due to poor or bloated coding, internal errors, or even manmade timers defining the application's performance speed. Possible Symptoms: Large delay in the application response time value.

Topic 87: A Four-Part Analysis Methodology

In this topic, we explain an analysis technique for the troubleshooting of networks.

A Network Analyst's Perspective includes Task 1: Define the problem, Task 2: Collect system, application and path information, Task 3: Capture and analyze packet flows, and Task 4: Consider tools.

Define the Problem

A common complaint from network users: "The network is slow". This is a vague description. Need more information from the user. Q1. What type of traffic will you be looking for? A file upload, file download, login, email send/receive or something else? Q2. What server target were you communicating with? Q3. What are the symptoms? Q4. Did you receive any error message? Q5. Is this happening all the time?

Collect System, Application and Path Information

Obtain as much system, network and infrastructure information as possible. Ask the customer about their configuration. Q1. What operating system is running on the client/server? Q2. What application (and version) is running? Q3. Describe the network path that the traffic must traverse.

Capture and Analyze Packet Flows

Capture Location Tips: Capture as close as possible to the complaining host so that you can observe the traffic from that host's perspective, examine the roundtrip time to the target(s) etc. Capture Tool Tips: Prefer using a tap or dedicated capture device rather than spanning a switch port for the capture process. Analysis Process Tips: Know what is "normal". Remove unrelated traffic from view. Focus on traffic related to the complaining user's machine etc.

Consider Other Tools

Once you have hit a Wireshark limitation, you may need to work with another tool. If size of trace files becomes larger than 100 MB, Wireshark becomes too slow.

Topic 88: Using a Troubleshooting Checklist

This topic describes the usage of a Troubleshooting Checklist.

Captured traffic is saved in a trace file. When you open a trace file, use a basic troubleshooting checklist. The order in which you go through the checklist may change depending on the troubleshooting issue.

Verify Trace File Integrity and Basic Communications

Look for ACKed Unseen Segment. Verify traffic from the complaining user's machine is visible. Verify resolution process completion such as DNS/ARP.

Focus on Complaining User's Traffic

Filter out unrelated traffic. Export related traffic to a separate trace file.

Detect and Prioritize Delays

Sort and identify high delta times. Measure path latency (Round Trip Time) using delta times in TCP handshake. This can be achieved by measuring delta from TCP SYN to SYN/ACK. Measure server response time. This can be obtained by measuring from request to response.

Look for Throughput Issues

Click on low throughput points to jump to problem spots in the trace file. Look at traffic characteristics at low throughput points.

Check Miscellaneous Traffic Characteristics

Check packet sizes during file transfer. Check for ICMP messages. Check for IP fragmentation.

TCP-Based Application: Determine TCP Connection Issues/Capabilities

Look for unsuccessful TCP handshakes. Examine the TCP handshake Options area, e.g. MSS, SACK.

UDP-Based Application: Identify Communication Issues

Look for unsuccessful requests.

Spot Application Errors

Filter for application error response codes.

Trace Files used in Topics from 89 to 172 are available at http://www.wiresharkbook.com/troubleshooting.html.

Topic 89: Wireshark Lab 1

This topic describes how to create a Troubleshooting Profile in Wireshark.

Until you create a new profile, you are working in Wireshark's Default profile. The profile you are working in is shown in the right side column of the Status Bar. This is shown next.



Figure 89.1: Create Your Troubleshooting Profile

You can create profiles to customize Wireshark with buttons, colors, and more. You can create separate profiles for different needs. For example, you may want to make a VoIP profile, a WLAN profile, and a general troubleshooting profile. You can quickly switch between profiles depending on your needs.

Step1: Right-click the Profile column on the Status Bar.

Step2: In the Configuration Profile window, select New.

Step3: Click the **arrow** in the **Create from** area, expand the **Global section** and select **Classic**. This profile uses the most vibrant colors.

Step4: Enter Troubleshooting Book Profile in the Profile Name area. Click OK.

	Classic									
Profile name: Troubleshooting Book Profile	Troubleshooting Book Profile									

Figure 89.2: Troubleshooting Book Profile

As soon as you create your new profile, the Wireshark Status Bar indicates that you are working in the Troubleshooting Book Profile, as shown next.



Figure 89.3: Wireshark Status Bar indicates the Troubleshooting Book Profile

You will be able to add capabilities and customization to this new profile. Wireshark also allows download/import a predefined profile for immediate use.

Topic 90: Wireshark Lab 2

This topic discusses how to enhance the Packet List Pane Columns in Wireshark.

By default, the Packet List pane contains: No. (number), Time, Source, Destination, Protocol, Length, and Info columns. This is shown next.

No.		Time	Source	Destination	Protocol	Length	Info
	1	0.00000000	24.6.173.220	96.17.148.114	ТСР	66	523
	2	0.017118000	96.17.148.114	24 6.173.220	ТСР	66	80-
	3	0.017290000	24.6.173.220	96.17.148.114	ТСР	54	523

Figure 90.1: Enhance the Packet List Pane Columns

You can add columns to display additional information about packets to speed up your analysis process.

Step 1: Open *tr-httpdelta.pcapng*. This trace file contains traffic to/from a user's machine that is checking for Windows updates as well as virus detection updates.

Step 2: Packets 1-3 are TCP handshake packets. Packet 4 is an HTTP GET request for a file called *minitri.flg.* Packet 5 is an ACK for GET request. Packet 6 is the HTTP 200 OK. This is shown next.

1 0.00000000	24.6.173.220	96.17.148.114	TCP	66 52382→80 [SYN] Seq=0
2 0.017118000	96.17.148.114	24.6.173.220	TCP	66 80→52382 [SYN, ACK]
3 0.017290000	24.6.173.220	96.17.148.114	TCP	54 52382→80 [ACK] Seq=1
4 0.01818600	24.6.173.220	96.17.148.114	HTTP	343 GET /minitri.flg HTT
5 0.036422000	96.17.148.114	24.6.173.220	TCP	60 80→52382 [ACK] Seq=1
6 0.037222000	96.17.148.114	24.6.173.220	HTTP	317 HTTP/1.1 200 OK (te

Figure 90.2: Enhance the Packet List Pane Columns

Select Packet 6 in the Packet List pane and then, in the Packet Details pane, expand the Hypertext Transfer Protocol.



Figure 90.3: Add and Use a Custom Column to Locate HTTP Delays

Step 3: Scroll to the bottom of the HTTP section and right-click on the [Time since request: 0.019036000seconds] line. Select Apply as Column.



Step 4: Wireshark places the new Time since Request column to the left of the Info column. Right-click on this new column heading and select **Edit Column Details**. Enter **HTTP Delta** in the Title area. Click **OK**.

Title:	HTTP Delta
Field type:	Custom
Field name:	http.time
Occurrence:	0

Step 5: Click twice on your new **HTTP Delta** column header to sort the column data from high to low. Wireshark indicates there is a 2.807332 second delay before one of the HTTP 200 OK responses (Packet 49).

File	Edit View Go	<u>Capture</u> Analyze	Statistics Telephony Tools	s Internals Hel	p	Click twic	e on the coli	umn
0	• 🖌 🔳 🔬	B & X 2	0,0071		199	header to so	ort from high	to lov
Fit	ter			• Expre	ssion Cle	ar jay	Save	
Vo.	Time	Source	Destination	Protocol	Length HT	TP Delta 🔹 🔻	Info	
49	2.675406	216.115.74.	202 24.6.173.220	HTTP	544 2.	807332000	HTTP/1.1	200
33	0.823550	216.115.74.	202 24.6.173.220	HTTP	543 0.	958153000	HTTP/1.1	200
5	0.000005	96.17.148.1	14 24.6.173.220	HTTP	147 0.	037208000	HTTP/1.1	200
105	0.000005	96.17.148.1	14 24.6.173.220	HTTP	90 0.	035879000	HTTP/1.1	200
j	0.000800	96.17.148.1	14 24.6.173.220	HTTP	317 0.	019036000	HTTP/1.1	200
)6	0.000860	96.17.148.1	14 24.6.173.220	HTTP	317 0.	016734000	HTTP/1.1	200
1	0.000778	96.17.148.1	15 24.6.173.220	HTTP	317 0.	015392000	HTTP/1.1	200
i3.	0-015217	96.17.148.1	15 24.6.173.220	ATTPA	202 0.	015217000.	HTTP/1.1	-204

Step 6: Right-click on your **HTTP Delta** column heading and select **Hide Column**. You can restore this hidden column at any time. Right-click on any column header, select **Displayed Columns** and select column to restore.

Topic 91: Wireshark Lab 3

In this topic, we learn how to change the Time Column settings in Wireshark.

Packets are time stamped at the moment they are captured. By default, Wireshark sets the Time column to Seconds Since Beginning of Capture. In addition, the resolution is set to nanoseconds regardless of whether the packet timestamps contain that level of precision.

Step 1: Open *tr-australia.pcapng*. This trace file contains a web browsing session and was captured at the client. We want to change the Time column setting so we can quickly measure the delta time between displayed packets.

Step 2: This trace file begins with a DNS query and response. TCP connection establishment begins in Packet 3. Select **View | Time Display Format | Seconds since Previous Displayed Packet**.

Step 3: To change the precision, select **View | Time Display Format | Milliseconds: 0.123**. Now, look at the time from the SYN (Packet 3) to the SYN/ACK (Packet 4). It appears the round trip time is 192 (ms).

		172										6
tr	-australia.pci	apng		LONG TO LOST								
File	Edit View	<u>G</u> o <u>(</u>	apture A	nalyze	Statistics Telephony	<u>I</u> ools Inter	nals <u>H</u> elp)				1
0	۰ 🖌 🔳			XZ	0 💠 🔿 🖄	7 1 1		QC	2		2 👧	X 1
Fi	lter:						Expres	sion	Clear	Ar	nlv	Save
No	Time	Source			Destination	Protocol	Length	Info	cicui	1.1	-19-0	Sare
1	0.000	192.	168.1.	72	192.168.1.25	4 DNS	76	Stan	dard	auer	•v 0x	4158
2	0.302	192.	168.1.	254	192.168.1.72	DNS	92	Stan	dard	que	y re	spons
3	0.001	192.	168.1.	72	101.234.75.2	0 TCP	66	9872	> h	ttp	[SYN]	Seq=
4	0.192	101.	234.75	.20	192.168.1.72	ТСР	66	http	> 9	872	SYN,	ACK]
5	0.000	192.	68.1.	72	101.234.75.2	0 тср	54	9872	> h	ttp	[ACK]	Seq=
6	0.000	_				HTTP	583	GET	/ HT	TP/1.	.1	1
7	0.198	1	he roun	d trip	time to the HITP	TCP	1514	[TCP	seg	ment	of a	reas
8	0.002		server	is jus	st over 190 ms	TCP	1514	[TCP	seg	ment	of a	reas
9	0.000	101.	234.13	.20	192.100.1.72	TCP	1514	[TCP	seg	ment	of a	reas
10	0.005	192.	168.1.	72	101.234.75.2	20 ТСР	54	9872	> h	ttp	[ACK]	Seq=
11	0.000	101.	234.75	.20	192.168.1.72	ТСР	1514	[TCP	seg	ment	of a	reas
12	0.000	101.	234.75	.20	192.168.1.72	тср	60	[TCP	Dup	ACK	11#1	1 ht
120	10000		4. 7.5	00	107	THE OWNER OF THE OWNER	77 4	and not		-Ann	100	-

Figure 91.1: Set the Time Column to Detect Path Latency

Topic 92: Wireshark Lab 4

This topic describes how to apply a filter on a host, subnet or conversation in Wireshark. When you capture traffic at the server or inside the network infrastructure: your trace file may contain conversations between many hosts on the network. Are you interested in the traffic between a specific client and server? If yes, you can apply a display filter based on a host address, a subnet address or a conversation.

Step 1: Open tr-cnn.pcapng.

Step 2: This trace file contains numerous conversations. Let's extract conversations between the local client and cnn.com servers. We begin with looking at the name resolution information that Wireshark extracted from the trace file. Select **Statistics | Show Address Resolution**.



Figure 92.1: Extract and Save a Single Conversation

Servers in cnn.com domain all begin with 157.166. Click **OK** to close the Address Resolution window.

Step 3: In the display filter area, enter **ip.addr==157.166.0.0/16**. Click **Apply**. The Status Bar indicates that 360 packets match the filter. This lists all conversations to and from the cnn.com servers.

Step 4: Select **File | Export Specified Packets**. The **Displayed** radio button is selected by default, as shown next. Name your file *tr-cnntraffic.pcapng*. Click **Save**.

File name:	tr-contraffic.pcapng			
Save as type:	Wireshark/ • pcap	ing ("pcaping," pcaping gz;	"ntar;" ntar.gz)	
Packet Range				Comments with a
		Captured	Displayed	Compress with g
Al packets		4730	360	
Selected participation	acket	1	1	
🔿 Marked par	ckets	0	0	
First to last	marked	0	0	
O Range:		0	0	
Renovalo	nored nackets	0	n i	

Step 5: We can also use right-click method to apply a display filter to a single conversation. Let's do this on **Packet 3** in the Packet List pane. Select **Conversation Filter | TCP**. 55 packets appear.

Step 6: Select **File | Export Specified Packets**. Name your file *tr-cnnconv1.pcapng*. Click **Save**.

Step 7: Click the **Clear** button to remove your display filter. Avoiding unrelated traffic and potential distractions makes analysis easier. Therefore, save the traffic from an interesting conversation to a separate trace file.

Topic 93: Wireshark Lab 5

This topic describes how to apply a filter using port number in Wireshark.

There are two ways to define a display filter on an application in a trace file—you can filter based on the application name (if known to Wireshark), or the port number in use.

Step 1: Open *tr-twohosts.pcapng*. Let's look at the FTP data transfer connection established by 192.168.1.119.

Step 2: Enter **ip.addr==192.168.1.119** in the display filter area and click **Apply**. This filters all traffic to and from 192.168.1.119. Look for response to the PASV command (Packet 3,959). Expand FTP section in the Packet Details pane to see port no. that the server will be listening on for the FTP data channel (port 39,757).

Filter:	ip.addr	==192.168.1.119			← Expression Clear Apply Save
No.	Time	Source	Destination	Protocol	Length Info
2732	0.000	192.168.1.119	200.236.31.1	TCP	66 50743→21 [SYN] Seq=0 Win=8192 Len=0 MS
2975	0.224	200.236.31.1	192.168.1.119	TCP	66 21-50743 [SYN, ACK] Seq=0 Ack=1 win=26
2976	0.000	192.168.1.119	200.236.31.1	TCP	54 50743→21 [ACK] Seq=1 Ack=1 Win=65536 L
3221	0.220	200.236.31.1	192.168.1.119	FTP	366 Response: 220 Welcome to Pur
3222	0.000	192.168.1.119	200.236.31.1	FTP	70 Request: USER anonymous
3463	0.217	200.236.31.1	192.168.1.119	TCP	54 21-50743 [ACK] Seq=313 Ack=17 Win=2688
3464	0.002	200.236.31.1	192.168.1.119	FTP	84 Response: 230 Anonymous user logged in
3465	0.000	192.168.1.119	200.236.31.1	FTP	62 Request: TYPE I
3713	0.218	200.236.31.1	192.168.1.119	FTP	84 Response: 200 TYPE is now 8-bit binary
3714	0.000	192.168.1.119	200.236.31.1	FTP	60 Request: PASV
3959	0.217	200.236.31.1	192.168.1.119	FTP	103 Response: 227 Entering Passive Mode (2
2060	0 000	100 160 1 110	100 116 11 1	TCD	66 50744 30757 [CVAI] Cog 0 Min 0103 Lon 0
<					
🗄 Fr	ame 39	59: 103 bytes o	n wire (824 bits	5), 103	bytes captured (824 bits) on interface 0
🗄 Et	hernet	II, Src: Cisco	-Li_d9:94:c0 (00):1d:7e:	:d9:94:c0), Dst: IntelCor_06:e2:8f (4c:80:93:
+ In	ternet	Protocol Versi	on 4, Src: 200.2	236.31.1	1 (200.236.31.1), Dst: 192.168.1.119 (192.168
🗄 Tra	ansmis	sion Control Pro	otocol, Src Port	:: 21 (2	21), Dst Port: 50743 (50743), Seq: 373, Ack:
🖯 Fi	le Tra	nsfer Protocol	(FTP)		
-	227 En	tering Passive I	Mode (200,236,31	,1,155	,77)\r\n Port for the ETP file
	Resp	onse code: Ente	ring Passive Mod	le (227)) FOILIOI LIE FIF IIIE
	Resp	onse arg: Enter	ing Passive Mode	e (200,2	236,31,1,155,77) transfer from server
	Pass	ive IP address:	200.236.31.1 (2	200.236.	. 31.1)
	Pass	ive port: 39757			
1					

Figure 93.1: Filter Traffic Based on a Port Number

Step 3: Replace your address display filter with **tcp.port==39757**. Click **Apply**. The Status Bar indicates that 28,020 packets match this filter. Next, we contrast this with a display filter based on an application name.

Step 4: Replace your TCP port filter with **ftp-data** and click **Apply**. Now you won't see the TCP handshake, ACKs, or the connection teardown packets.

4 tr	-twohosts.pca	ipng			
Ele O Filte	Edit <u>V</u> iew	Go Capture Ana	lyze Statistics Telephony 😥 🔍 🗢 🏟 🏟 🕉	This application connection establis	name filter does not display any TCP shment, management or tear down traffic
No.	Time		Source	Destination	Protocol Info
22	1.6	66742000	200.236.31.1	192.168.1.72	FTP-DFTP Data: 429 bytes
39	11.	026613000	200.236.31.1	192.168.1.72	FTP-DFTP Data: 215 bytes
57	18.	775188000	200.236.31.1	192.168.1.72	FTP-DFTP Data: 1460 bytes
62	18.	777706000	200.236.31.1	192.168.1.72	FTP-D[TCP Out-Of-Order] FT
63	18.	777713000	200.236.31.1	192.168.1.72	FTP-D[TCP Out-Of-Order] FT
80	39.	235186000	200.236.31.1	192.168.1.72	FTP-DFTP Data: 1460 bytes
81	39.	236447000	200.236.31.1	192.168.1.72	FTP-DFTP Data: 1460 bytes
82	39.	236457000	200.236.31.1	192.168.1.72	FTP-DFTP Data: 1460 bytes
83	39.	236463000	200.236.31.1	192.168.1.72	FTP-DFTP Data: 1460 bytes
85	39.	237499000	200.236.31.1	192.168.1.72	FTP-DFTP Data: 1460 bytes
86	39.	237502000	200.236.31.1	192.168.1.72	FTP-DFTP Data: 1460 bytes
88	39.	238822000	200.236.31.1	192.168.1.72	FTP-DFTP Data: 1460 bytes
-			Annana and	A 46 1 23	A A second during a line of

Step 5: Click the Clear button to remove your display filter.

Topic 94: Wireshark Lab 6

In this topic, we apply a filter using a field value in Wireshark.

We often need to identify packets that contain a specific field or a specific field value. For example, the display filter **http.request.method** can be used to view all HTTP client request packets using **http.request.method** field. **dns.flags.rcode > 0** is used to identify DNS error responses with code greater than 0. If you have a packet that contains the field name in which you are interested, you can right-click on the field and select **Apply as Filter**.

Step 1: Open *tr-winsize.pcapng,* HTTP clients send commands such as GET and POST in the HTTP Request Method field. Let's display all such packets.

Step 2: Right-click on the **Hypertext Transfer Protocol** section in the Packet Details pane in **Packet 4** and select **Expand Subtrees**. When you click on the **Request Method: GET** line, the Status Bar provides the name of this field—http.request.method.

Step 3: Type **http.request.method** in the display filter area and click **Apply**. One packet matches the filter. We can use this filter to determine how many HTTP requests were sent to a server. For example, **ip.addr==10.1.1.1 && http.request.method** would display all HTTP requests to or from 10.1.1.1. If 10.1.1.1 is a client, you would only see HTTP requests sent from this host.

Step 4: Click the Clear button to remove the display filter.

Topic 95: Wireshark Lab 7

This topic describes how to locate Buffer Problems using the Calculated Window Size Field in Wireshark.

Every TCP packet sent by a host contains that host's available receive buffer space in the Window Size field. If Window Scaling is in use, this field value is multiplied by a scaling factor. When the advertised buffer space becomes equal to zero, the host cannot accept any more data—a Zero Window condition. Even a low Window Size value can stop a TCP peer from transmitting data.

Step 1: Open tr-winsize.pcapng.

Step 2: Expand any **TCP header** in the Packet Details pane. Right-click on the **Calculated window size** field and select **Prepare a Filter | Selected**.

Step 3: Change the display filter value to **tcp.window_size < 1000** and click **Apply**. This displays Packet 374 in which the client is advertising a 536-byte receive buffer. What will happen if the TCP peer has more than 536 bytes of data queued to transmit? This low Window size value will stop data transmission. Next, we look at this phenomenon.





Step 4: Click Clear to remove your filter. In this trace file, we next view the delay that occurs before the Window Update packet (Packet 375).

📕 tr-winsize.pcapng				1
Eile Edit View Go	Capture Analyze Statistics Telepho	ony Iools Internals Help		5
0.0414	Notice the delay immedia 536-byte Window Size y	tely following the R G		1 🖻 🥵 🗱 🗋
Filter			lear Apply Sa	ve
No. Time 300 1.033	50 52.104.134	Destination LU.U.JZ.LO4	Protocol	LICP Segment
369 1.853	0.52.164	204.152.184.134	TCP	and-lm > http
370 1.854	4.152.184.134	10.0.52.164	TCP	[TCP segment
371 1.855	204.152.184.134	10.0.52.164	TCP	[TCP segment
372 1.855	10.0.52.164	204.152.184.134	TCP	and-lm > http
373 1.856	204.152.184.134	10.0.52.164	TCP	[TCP segment
374 2.049	10.0.52.164	204.152.184.134	TCP	and-lm > http://www.and-lm.
375 4.802	10.0.52.164	204.152.184.134	TCP	[TCP Window
376 4.803	10.0.52.164	204.152.184.134	1	[TCP Window U
377 4.984	204.152 184 134	10 0 52 164	TCP	[TCP segment
378 4.984	10.0.52 Hinting	of a Window Size probl	em, the	and-1m > htt
379.4.985	204,152 delay	y precedes a Window Up	odate	[TCP segment

Essentially, the server could not transmit the full-sized packet because the client only had 536 bytes of buffer space available. The server had to wait until the client's buffer size increased.

Topic 96: Wireshark Lab 8

In this topic, we discuss how to filter OUT "Normal" Traffic to focus on anomalies in Wireshark.

To filter out traffic based on an application name, simply precede the application display filter name with an exclamation sign (!). For example, to remove ARP from view, use **!arp**. There are two methods you can use to exclude traffic based on a field value. One method uses the **!** or **not** operator with **==** or **eq**. The other uses the **!=** or **ne** operator.

Examples

- !ip.addr==10.10.10.10
- http.request.method != "GET"

When to Use ! and == and When to Use !=

Use the first method when you filter on a field name that matches two fields such as **ip.addr**, **tcp.port**, or **udp.port**. Use the second method when you refer to a field name that only matches one field such as **dns.flags.rcode** or **tcp.dstport**.

Correct Display Filter

- !ip.addr==10.1.1.1
- dns.flags.rcode!=0

Incorrect Display Filter

- ip.addr != 10.1.1.1
- !dns.flags.rcode==0

This lab removes a set of applications and protocols from view to determine what other traffic is seen on this network.

Step 1: Open tr-general.pcapng.

Step 2: In the display filter area, type **!tcp && !arp** and click **Apply**. There are 40 packets that match this filter.

Step 3: To remove DNS and DHCP from view, expand the filter by adding **&& !dns && !bootp.** Eight packets are displayed.

The displayed packets indicate that there are two hosts running Dropbox on the network and are sending packets to the broadcast address (255.255.255.255).

Topic 97: Wireshark Lab 9

This topic presents how to Create Filter Expression Buttons in Wireshark.

Filter Expression buttons are based on display filters. These buttons can be created and used to quickly apply display filters to your traffic to identify common network problems. Let's create a button to quickly identify TCP handshake packets that do not offer Selective Acknowledgment (SACK) or Window Scaling functionality. To do this, we need to combine an inclusion filter for the SYN bit set to 1 with an exclusion filter for the SACK and Window Scaling options.

Step 1: Open tr-smbjoindomain.pcapng.

Step 2: Packet 11 is the first SYN packet in the trace file. Right-click on the TCP header of this packet and select Expand Subtrees. Let's first build the filter and then turn the filter into a filter expression button. Right-click on **SYN: Set** line and select **Prepare a Filter | Selected**. Wireshark places first part of filter in the display filter area.

Filter: tcp.flags.syn == 1

Figure 97.1: Create a Button to Detect Missing TCP Functionality

Step 3: Scroll down to the TCP Options area. Click on the **TCP SACK Permitted Option: True** line. This displays **tcp.options.sack_perm** in the Status Bar area. Expand the filter by typing **&& !tcp.options.sack_perm** so that TCP handshake packets that do not contain this value can be observed.



Step 4: Now add **|| ! tcp.options.wscale.multiplier** and put parentheses around the options, as shown next.



Step 5: Click the Save button and name your new button TCP-HS and click OK

Wireshark: Save Filter		• X		
Save Filter as				
tcp.flags.syn==1 && (!tcp.options.	sack_perm ! tcp.options.wscale.multiplier)	TCP-HS		
Help	QK	Cancel		

Step 6: Click your new TCP-HS button. There are 38 packets that match the TCP-HS button filter. The connections established by these packets will not support all the desired TCP functions.

	smbjoin	aoma	minpea	iping							L			0000								
File	Edit 1	lew	<u>G</u> 0	Captu	ire	Analy	ze S	tatisti	CS I	ele	phon	iy ⊥o	ools įr	ntern	als <u>H</u> e	lp						
0	۱		Ø			×	2	9	\$	\$		7	₽ [Ð G		. 🖭			8 %	f 🕱
Filte	r: s.sy	n==1	8.8. (tcp.op	tions	.sack	_perm	! tc	p.opti	ons	.wsca	ale.mu	ltiplier)	-	Express	on	Clea	r 4	Apply	Save		P-HS
о.	Time	S	ource				Dest	inatio	n				Protoc	ol	Length	Info				1	1	
1	0.00	0	192.	168	.0.	88	19	2.1	68.	0.	99		TCP		62	sg	i-s	tor	man		icro	soft-
2	0.00	0	192.	168	.0.	.99	19	2.1	68.	0.	88		TCP		62	mi	cro	sof	-	> 5	sqi-	storn
7	14.6	06	192.	168	.0.	.88	19	2.1	68.	0.	9	Clial		(0)	r now	тор	це	hutt	onto	data	at	crose
8	0.00	0	192.	168	.0.	99	19	2.1	68.	0.	8	CIICI	(OII)	you	Thew	IGF	-по	bull	onic	uele	GL	rfup-
4	0.00	12	192.	168	.0.	88	19	2.1	68.	0.	9	the	38 5	YN	or SY	N/AC	Kp	acke	ets th	at are	9	beros
5	0.00	0	192.	168	.0.	99	19	2.1	68.	0.	8	m	issin	g ti	ne Wir	ndov	Sca	aling	fund	ction		uster
05	0.42	5	192.	168	.0.	88	19	2.1	68.	0.	99		TCP	-	62	CO	mm L	Inx	-av	>	dap	[SYN
															1	11						
-	-		C 7				-		(10		1.1		62		13.11				(100			

If you need to edit, reorder or delete Filter Expression buttons, select Edit | Preferences | Filter Expressions.

Topic 98: Wireshark Lab 10

This topic discusses how to launch and navigate through the Expert Infos. in Wireshark.

Wireshark's Expert Infos can help you quickly detect network problems as well as obtain basic information about network communications and view/jump to packet comments.

Step 1: Open tr-twohosts.pcapng.

Step 2: Click the **Expert Infos** button in the bottom left corner of the Status Bar. This will open the **Expert Infos** window.



Figure 98.1: Use Expert Infos to Identify Network Problems

The Expert Infos window is divided into six tabs: **Errors**: Checksum errors, dissector failures. **Warnings**: Potential problems detected. **Notes**: Symptoms of problems; typically recovery processes. **Chats**: TCP connection overhead (handshake, Window updates, disconnects). **Details**: Summary of Errors, Warnings, Notes and Chats. **Packet Comments**: List of all packet comments in trace file.

Step 3: If IPv4 checksum validation is disabled, there are no Expert Infos Errors in this trace file. Otherwise, you will see 6,767 IPv4 Bad Checksum Errors. To disable IPv4 checksum validation, toggle back to Wireshark, rightclick on the **Internet Protocol Version 4** line in the Packet Details Pane, select **Protocol Preferences** and toggle off the *Validate IPv4 checksum if possible*. Click the Warnings tab of **Expert Infos** window. Click on a packet to jump to that location in the trace file.



Step 4: Click the **Notes** tab. As you scroll the list, you will see 589 Duplicate ACKs sent by receiver to recover a missing packet. This can be caused by a very high latency path or a brief connection outage.

Errors: 0 (0)	Warnings: 2 (473)	Notes: 591 (3819)	Chats: 15 (197)	Details: 4489	Packet Comments: 0			
Group	Protocol	 Summary 				 Count 	•	
E Sequence	TCP	Duplicate ACK (#	582)				1	
E Sequence	TCP	Duplicate ACK (#	583)				1	
E Sequence	TCP	Duplicate ACK (#	584)				1	
E Sequence	TCP	Duplicate ACK (#	585)	At one po	int in the trace fi	le the	1	
B Sequence	TCP	Duplicate ACK (#	586) re	ceiver ask	ed more than 50	0 times	1	
E Sequence	TCP	Duplicate ACK (#	587)	for th	e missing packe	t	1	
E Sequence	TCP	Duplicate ACK (#	588)				1	
E Sequence	TCP	Duplicate ACK (#	589)				1	
Sequence	TCP	Fast retransmissi	on (suspected)				5	

Step 5: Click **Close** to shut down the Expert Infos window.

Topic 99: Wireshark Lab 11

This topic describes how to change the Dissector Behavior in Wireshark.

Some of Wireshark's predefined preference settings are not ideal for troubleshooting. For example, the *Allow subdissector to reassemble TCP streams* preference setting.

Step 1:	Open	tr-youtube	bad.pcapng
---------	------	------------	------------

tr	-youtu	pebad.p	pcapng				*	95	1	-		2) (P			2	- 54	æ	1
<u>F</u> ile	Edit	View	<u>G</u> o <u>C</u>	apture	Analyze	Statis	ics Tel	ephon	<u>T</u> ools	Intern	als <u>H</u> elp							
0	۰ (1		XR	0	¢ ¢	•	7 1			Q Q 🗹		1 🔞	% 0			
Fi	lter:										 Expressi 	on Clear	App	ly	Save			-
No.	Time		Source				Destinati	ion			Protocol	Length	Info					
1	0.	000	24.4	4.7.3	217		208.	117	.232	102	TCP	66	56770) >	http	[S)	'N]	Se
2	0.	029	208	.117	.232.	102	24.4	.7.	217		TCP	66	http	> 5	6770	[S)	Ϋ́Ν,	AC
3	0.	000	24.4	4.7.3	217		208.	117	.232.	102	TCP	54	56770) >	http	[AC	К]	SC
4	0.	000	24.4	4.7.3	217		208.	117	.232.	102	HTTP	1323	GET /	vic	leop1	ayba	ick	?spl
5	0.	024	208	.117	.232.	102	24.4	.7.	217		TCP	60	http	> 5	6770	[AC	к]	Sed
6	0.	276	208	.117	.232.	102	24.4	.7.	217		TCP	379	[TCP	seg	gment	of	a	rea
7	0.	000	208	.117	.232.	102	24.4	.7.	217		TCP	6	[TCP	seg	ment	of	а	rea
8	0.	000	208										[TCP	sec	ment	of	a	reà
9	0.	000	208		If the I	CP re	assen	nbly s	setting	is en	abled, H	112	[TCP	sec	ment	of	a	rea
10	0.	000	208	re	spons	e pac	kets th	nat co	ontain	data a	are mark	ed as	[TCP	sec	ment	of	a	real
11	0.	000	208	pa	rt of a r	eass	emble	d PD	J - the	200 O	K is sho	wn on	[TCP	sec	ment	of	a	red
12	0.	000	208		the la	ast pa	icket c	of the	down	loade	d elemer	nt	TCP	sec	ment	of	a	reas
13	0.	000	208	.117	.232.	102	24.4	./.	217		TCP	/3	TCP	sec	ment	of	a	re
14	0.	000	208	.117	.232.	102	24.4	.7.	217		TCP	310	ГТСР	sec	ment	of	a	re
-	MA	000-	and	1.2.7	man	ann	2 4-	7	Jam.	-	TOP	.138	-ACP	000	movie	No f	-	mark

Figure 99.1: Change the TCP Dissector Reassembly Setting to Properly Measure HTTP Response Times

Response Code in the Info column for Packet 6 cannot be seen because the **Allow subdissector to reassemble TCP streams** preference setting is enabled.

Step 2: In the Packet Detail pane of **Packet 4**, right-click the **Hypertext Transfer Protocol** heading and select **Expand Subtrees**. A hyperlink to the response packet is located at the bottom of the HTTP section. Double-click the **hyperlink** to jump to **Packet 29,259**.



Step 3: Scroll to the bottom of the HTTP header in Packet 29,259. The Time Since Request (http.time) field indicates the HTTP response time was over 276 seconds.



This is not correct. The HTTP response time is measured from the HTTP request packet to the HTTP response packet that contains the 200 OK response.

Step 4: To find the actual HTTP response time, in the Packet Details pane of any packet, right-click the **TCP header**, select **Protocol Preferences** and toggle off the *Allow subdissector to reassemble TCP streams* preference setting.

Step 5: Now click the Go to First Packet button. Notice we see that Packet 6 actually contains the 200 OK response. Examine the HTTP response time value in Packet 6. It is just over 300 ms.

Topic 100: Wireshark Lab 12

This topic provides a way to find the top talkers in Wireshark.

We can use the Conversations window to determine the most active conversation based on hardware address, network address, or even the port numbers in use.

Step 1: Open tr-general.pcapng.

Step 2: Select **Statistics | Conversations**. The tabs indicate the number of each type of conversation seen in the trace file.

				13	Ethernet Conv	ersations					
Address A 4	Address B 4	Packets 4	Bytes 4 Pa	ackets A-B	Bytes A-B	Packets A+B 1	Bytes A+-B 1	Rel Start 4	Duration 4	bps A-B + b	ps A+B 4
PaceAmer_11:e2:b9	Hewletta7:bf:a3	1 004	423 541	561	345 830	443	77711	0.000000000	46.3653	59670.45	13408.47
Micro-St_3c:a9:84	Broadcast	7	420	7	420	0	0	2.167656000	28.8674	116.39	N//
PaceAmer_11:e2:b9	Broadcast	27	1 620	27	1 620	0	0	4,439383000	41.0899	315.41	N/4
Aicro St_3d:0b:40	Broadcast	9	1 104	9	1 104	0	0	4,977944000	38.6096	228.75	N/A
Micro-St_3d:10:d3	Broadcast	9	1 650	9	1 650	0	0	6.639666000	37.3467	353.44	N//
sewiellarchicas	Broadicast		082	11	002		0	15.431845000	SUM97	101.01	N/A
Name resolution	Limit to displ	ay filter									

Figure 100.1: Find the Most Active Conversation (Byte Count)

Step 3: We are interested in the most active TCP conversation (in bytes) in this trace file. Click the **TCP** tab and then click the **Bytes** column heading twice to sort from high to low. The most active conversation is between 192.168.1.72 on port 32313 and 192.87.106.229 on port 80 (listed as *http*).

Step 4: Right-click on this top conversation and select Apply as Filter | Selected A <->B. Wireshark applies the filter and displays the 123 packets of this conversation.

thernet: 0	Hbre Chann	ei FDDI IPv4: 30 1	PMB JPX JX	IA NCP	RSVP SCTP	TCP: 30 10	ken King L	JDP: 18 USB WL	AN	
	_			TCP Co	nversations					
Address A	Port , Rig	ht Click ss B 4	Port B + Par	ckets + B	ytes - Packet	s A→B 4 Byt	es A-B + I	Packets A-B + By	tes A+B 4	4
192.168.1.72	32313	197.87 106 229	htto	123	124 180	- 28	4.054	85	120 126	
192.168.1.72	32290	Apply as Filt	er	,	Selected		•	A - B	74 096	1
192.168.1.72	32318	Prepare a Fil	ter		Not Select	ted	•	A → B	40 857	۲
192.168.1.72	32296	Find Packet			and Sel	ected		A B	24 141	
192.168.1.72	32303	Colorize Cor	versation		or Selec	ted		A → Any	14 697	
192.168.1.72	32298	65.254.248.200	http	28	and not	Selected		A - Any	19 657	
192.168.1.72	32323	23.43.181.210	https	19	or not S	elected	•	A - Any	7 905	
192.168.1.72	32315	140.211.11.131	http	18	942/	/	9	Anv + B	8 4 5 4	
192.168.1.72	32289	199.59.148.92	https	26	6727	11	41	Any - B	2 576	
192.168.1.72	32291	65.254.248.200	http	18	5 460	10	26	Amy - B	2 783	
192.168.1.72	32309	143.127.102.125	http	14	4 304	7	11/0-	Any - b	3 1 2 6	
192.168.1.72	32320	192.35.244.50	ftp	47	4 276	18	1 233	29	3 0 4 3	
4			m						÷.	
										_

Step 5: When you are done, click **Clear** to remove your display filter, toggle back to the Conversations window and click **Close**.

Topic 101: Wireshark Lab 13

This topic provides how to build a basic IO graph in Wireshark.

The basic IO Graph can be used to view throughput levels for all traffic in a trace file, or plot a subset of traffic (based on display filters). The IO Graph can help you prioritize your troubleshooting tasks.

Step 1: Open *tr-winsize.pcapng*. This trace file contains a single TCP conversation. An HTTP client is downloading a large file from a web server.

Step 2: Select **Statistics | IO Graph**. By default, Wireshark displays the packets per second rate (packet per tick with a default tick rate of one second).

Wireshark IO Graphs: tr-winsize.pcap We have two distinct th during this file dow	roughput probl vnload process	ems]		J.		
< [. III	1		()s	20:	5
Graphs	C 1				X Axis	augh 1 cos	-
Graph I Color Filter:	style:	Line	-	Smooth	Divels ner	tick:	5 -
Graph 2 Color Filter:	Style:	Line	-	Smooth	I View	as time of	dav
Graph 3 Color Filter:	Style:	Line	-	Smooth	V Axis		,
Graph 4 Color Filter:	Style:	Line	-	Smooth	Unit:	Packets/	Tick 💌
Graph 5 Color Filter:	Style:	Line	-	Smooth	Scale:	Auto	-
					Smooth:	No filte	r 💌
<u>Н</u> еlp <u>С</u> ору					Save		<u>C</u> lose

Figure 101.1: Quickly Spot a Throughput Problem in an IO Graph

Let's investigate both of them.

Step 3: Click on the **first drop in throughput** in the graph. Wireshark jumps to that point in the trace file so we can investigate the problem further.

🖌 tr-w	vinsize.pcapng	- 1 A.	1. 1.						
<u>F</u> ile	<u>E</u> dit <u>V</u> iew <u>G</u> o	<u>Capture</u> <u>Analyze</u> <u>Sta</u>	tistics Telephony	Tools Inter	nals <u>H</u> elp				
0 0			. 🗢 🔶 7	20		QQ	•	1	6 📓
Filter				-	Expression	Clear	Apply	Save T	CP-HS
Vo. 🔺	Time	Source	Destination		Protocol	Length	Info		N 1050
370	0.000869		LUTILULI	164	HTTP	1230	Conti	nuation	or no
371	0.001347	There is a signif	icant delay at	.64	HTTP	1514	Conti	nuation	n or no
372	0.000013	the first drop in	throughput	.84.134	TCP	54	and-1	m > htt	p [ACK
373	0.001043	207.1 107.1.	10.0.32.	164	HTTP	1514	Conti	nuation	or no
374	0.193043	10 .52.164	204.152.	184.134	TCP	54	and-1	m > htt	p [ACK
375	2.753091	10.0.52.164	204.152.	184.134	TCP	54	[TCP	Window	Update
376	0.000921	10.0.52.164	204.152.	184.134	TCP	54	[TCP	Window	Update
377	0.180846	204.152.184.13	4 10.0.52.	164	HTTP	1514	Conti	nuation	n or no
378	0.000292	10.0.52.164	204.152.	184.134	TCP	54	and-1	m > htt	p [ACK
379	0.000703	204.152.184.1	4 10.0,52.	164	HTTP	1514	Conti	nuation	or no
mag /	magza	paran 6	the second	1.94 13/	TCA	gun	Au	um/hor	

If your Time column is set to *Seconds Since Previous Displayed Packet* (**View | Time Display Format**), you will see a delay of over 2.75 seconds in the trace file. Packet 375 is marked as a TCP Window Update packet. This indicates that the delay may have something to do with the Window Size value advertised by the client (10.0.52.164).

Step 4: Go to the IO Graph. Click on the **second problem point** in the graph. From Wireshark's window, we can see what is happening at this point in the file download process.

Eile Edit View	Go Capture Analyz	e Statistics Telephony	second problem point in the IO Graph							
Filter:			Expre	lear	Apply	Save T				
lo. 🔺 Time	Source	Destination	Protocol	th	Info					
56100.0011	17 204.152.18	4.134 10.0.52.	164 НТТР	14	Conti	inuation				
56110.0000	18 10.0.52.16	4 204.152.	184.134 TCP	66	[TCP	Dup ACK				
56120.0008	24 204.152.18	4.134 10.0.52.	164 нттр	1230	Conti	inuation				
56130.0000	18 10.0.52.16	4 204.152.	184.134 ТСР	66	[TCP	Dup ACH				
56140.0014	96 204.152.18	4.134 10.0.52.	164 нттр	1514	Conti	inuation				
56150.0000	24 10.0.52.16	4 204.152.	184.134 TCP	66	[TCP	Dup ACK				
56160.0010	33 204.152.18	4.134 10.0.52.	164 НТТР	1514	Conti	inuation				
6170.0000	31 10.0.52.16	4 204.152.	184.134 TCP	66	[TCP	Dup ACK				
6120-0008	6 204, 152, 18	4.134-10.0.5	64	230	Conti	inuation				

The client appears to be in the middle of a packet loss recovery process.

Step 5: Go to the IO Graph and click **Close**.

Topic 102: Wireshark Lab 14

This topic discusses the addition of a coloring rule in Wireshark.

Coloring rules can be used to quickly identify packets in the Packet List pane. Let's build a coloring rule to highlight DNS errors.

Step 1: Open tr-chappellu.pcapng.

Step 2: Type **dns** in the display filter area to capture all DNS traffic and then click **Apply**. Forty-three packets will match your filter. The client asks two DNS servers (75.75.75.75 and 75.75.76.76) to resolve www.chappellU.com nine times before receiving an answer (Packet 12).

🖌 tr-	chappellu.pi	apng	T											,
File	Edit View	Go	Capture	Analyze	Statistics	Telephony	Tools	Inte	rnals <u>H</u> elp					
0	9 🧉 📕	Å		* 2) Q (•	7 2			0 🔟	🏽 🔁 😽	\$ <mark>0</mark>		
Filter	dns							•	Expression Clea	r Apply S	ave			1
No.	Time	Sourc	:e		Destinatio	n	Prote	ocol	Info					
	0.251	24	.6.173	.220	75.7	5.75.7	5 dn	S	Standard	query	0x5451	A WWW.C	nappel	lu.com
3	1.252	24	.6.173	.220	75.75	5.76.7	6 DN	S	Standard	query	0x5451	A www.c	nappel	lu.com
4	1.253	24	.6.173	.220	75.75	5.75.7	5 DN	s,	Standard	query	0x5451	A www.c	nappel	lu.com '
5	2.252	24	.6.173	.220	75.75	5.76.7	6 J	5	Standard	query	0x5451	A www.c	nappel	lu.com
6	2.252	24	.6.173	.220	75.75	5.75.7	DN	S	Standard	query	0x5451	A www.c	nappel	lu.com
8	4.252	24	.6.173	.2.	These or	a all rene	ated	5	Standard	query	0x5451	A www.c	nappel	lu.com
9	4.252	24	.6.173	.2.	DNS	roquoete	alou	S	Standard	query	0x5451	A www.c	nappel	lu.com
10	8.253	24	.6.173	.2	DNO	requests	<u> </u>	5	Standard	query	0x5451	A www.c	nappe]	lu.com *
11	8.253	24	.6.173	.220	75.75	5.76.7	6 DN	S	Standard	query	0x5451	A www.c	nappel	lu.com
12	8.298	75	.75.75	.75	24.6	173.2	20 DN	S	Standard	query	response	0x5451	A 19	8.66.23
24	8.449	24	.6.173	.21	Thereau	anaa fr	ally	И	Standard	query	0xc16e	A www.g	oogle-	analyti
25	8.465	75	.75.75	.75	meres	Jonse III	any	6	Standard	query	response	e Oxc16e	CNAM	E www-q
26	asoc	-2/		~~	a	rrives			A COMPANY	-	and an and		000	John Come

Figure 102.1: Build a Coloring Rule to Highlight DNS Errors

Step 3: Right-click on the **Domain Name System (query)** section in the Packet Details window of Packet 12. Select **Expand Subtrees**. When the Reply Code field inside the Flags section contains a 0, the DNS response was successful. If it contains any other value, the response indicates there is as DNS error.

Step 4: Right-click on the Reply Code field and select Colorize with Filter | New Coloring Rule.

Step 5: Enter **DNS Errors** as the name of your new coloring rule. Change the String value to dns.flags.rcode > 0.

Frame 12: 93 bytes on wire (744 bits) Ethernet II, Src: Cadant_31:bb:c1 (00 Internet Protocol Version 4, Src: 75. User Datagram Protocol, Src Port: do Domain Name System (response) [Request In: 10] [Time: 0.044976000 seconds] Transaction ID: 0x5451 Flags: 0x8180 standard query respon 1 = Response: Me	Egand All Collapse All Apply as Column Apply as Column Colorize with Filter Colorize with Filter Follow UDP Stream Follow UDP Stream Follow UDP Stream Follow SLS. Stream Copy Esport Selected Packet Bytes Wikis Protocol Page Finter Field Reference Protocol Help Protocol Heferences Protocol Heferences	Dits) on interfac vletta7:bf:a3 st: 24.6.173.220 (54817) Color1 Color3 Color3 Color4 Color5 Color6 Color6 Color7 Color6 Color7 Color7 Color7 Color6 Color7 Color7 Color6 Color7 Color6 Color7 Color6 Color7 Color6 Color7 Color6 Color7 Color6 Color7 Color6 Color7 Color6 Color7 Color6 Color7 Color6 Color7 Color6 Color7 Color6 Color7 Color6 Color7 Color6 Color7 Color6 Color7 Color6 Color7 Color6 Color7 Color7 Color6 Color6
	Disable Protocol Resolve Name to Corresponding Packet to Vacket Reference in New Window or Packet Reference in New Window	/ portion was not
Questions: 1 40 Answer RRs: 1 Authority RRs: 0 Additional RRs: 0 @ Queries @ Answers		

Step 6: Click the **Background Color** button and type **orange** in the Color Name field. When you tab away from the Color Name field, Wireshark changes the word "orange" to the hex value #FFA500. Then, Wireshark shows the new color in the color preview area. Click **OK** to close the background color window and click **OK** to close the Edit Color Filter window.



Your new coloring rule appears at the top of the list of color filters. Packets are processed in order through this list. Now, DNS errors will appear with an orange background.

ait	Filter			Order
N		List is processed in order until match is four	nd	
New	Name	String	*	
	DNS Errors	dns.flags.rcode > 0		
<u>E</u> drt	Bad TCP	tcp.analysis.flags හිහි !tcp.analysis.wind	ow_update	Щр
	HSRP State Change	hsrp.state != 8 && hsrp.state != 16		
Enable	Spanning Tree Topology Cl	hange_stp.type == 0x80	New coloring rule	s are
	OSPF State Change	ospf.msg != 1	placed at the top of	the list
Disable	ICMP errors	icmp.type eq 3 icmp.type eq 4 icmp	biaced at the top of	une nat
	ARP	arp		Move
Delete	ICMP	icmp icmpv6		selected filte
anage	TCP RST	tcp.flags.reset eq 1		up or down
	SCTP ABORT	sctp.chunk_type eq ABORT		
mport	TTL low or unexpected	(! ip.dst == 224.0.0.0/4 && ip.ttl < 5 &&	k !pim) (ip.dst == 224.0.0.0/2	
	Checksum Errors	cdp.checksum_bad==1 edp.checksur	n_bad==1 ip.checksum_bad	
xport	SMB	smb nbss nbns nbipx ipxsap ne	tbios	Davia
	HTTP	http tcp.port == 80		Down
Clear	IPX	ipx spx	-	
-	4 [III		*	

Step 7: Click OK to close the Coloring Rules window.

Step 8: With **dns** filter still in place, scroll through the packets to see if you notice the two DNS errors in the trace file. Packets 83 & 84 appear with orange.

🚺 tr-c	happellu.pcap	ng					
File	<u>E</u> dit <u>V</u> iew	<u>Go Capture A</u> nalyze St	atistics Telephony <u>T</u> ools	internals <u>H</u> elp			
0 (9 🖌 🔳	(E 🗎 X 🔗	् 🗢 🕸 🕤 🖥 🖞		0, 🗹 📓 🛙	1 😽 🕺 🔯	
Filt	en dns			Expression	Clear App	ply Save	
No.	Time	Source	Destination	Protocol Info			
79	8.829	75.75.75.75	24.6.173.220	DNS Standar	d query i	response Oxel	18 CNAME cf-ss]1
80	8.830	24.6.173.220	75.75.75.75	DNS Standar	d query	Oxfd77 AAAA	www.nitroreader.com
81	8.838	75.75.75.75	24.6.173.220	DNS Standar	d query i	response 0xe4	dd AAAA 2001:4860
82	8.860	24.6.173.220	75.75.75.75	DNS Standar	d query (Oxfd77 AAAA	www.nitroreader.co
83	8.869	75.75.75.75	24.6.173.220	DNS Standar	d query i	response Oxfd	77 Server failure
84	8.873	75.75.75.75	24.6.173.220	DNS Standar	d query I	response Oxfd	77 Server failure 🤇
162	28.664	24.6.173.220	75.75.75.75	DNS Standar	d query (Oxa9b6 A vuz	it.com 🔰
164	29.665	24.6.173.220	75.75.76.76	DNS Standar	d query (0xa9b6 A vuz	it.com 🔰
165	29.665	24.6.173.220	75.75.75.75	DNS Standar	d avera	0xa9b6 A vuz	it.com

Step 9: Expand the **Frame** section in the Packet Details pane of Packet 83. You will see your Coloring Rule Name and Coloring Rule String listed in this area.

Topic 103: Capture Options for a Switched Network

This topic explains tips on choosing a capture location for a switched network.

These days, almost all network clients are connected to the network through a switch. Four types of packets are forwarded by a switch. Broadcast multicast, traffic to your hardware address and traffic to an unknown hardware address.

Because of this, when you connect a system running Wireshark directly to a switch port, you cannot listen to other users' traffic. In order to capture the traffic between the client and the upstream switch (and ultimately a remote host), you need to either

- (a) Install Wireshark or another capture tool on the user's machine,
- (b) Make the switch send a copy of the traffic down your analyzer port, or
- (c) Tap in and obtain a copy of the traffic between the client and the switch.

Install Wireshark on User's Machine: This is a great option—if you possible.

Switch Port Spanning: The next option to consider is to make the switch send a copy of the traffic down to your analyzer port (aka "spanning"). Not all switches support this feature. If there are corrupt frames traveling from the user's host, those corrupt frames won't be forwarded down the spanned port by the switch. Furthermore, avoid "oversubscribing" your switch port – spanning a level of traffic that cannot "fit" down the pipe to your Wireshark system. Switch will drop excess packets and trace file will be incomplete.

Use a Test Access Port ("Tap"): A tap is a simple device that copies all the traffic flowing through it (including those corrupt packets) out to a monitor port.

Topic 104: Wireshark Lab 15

In this topic, we describe the Wireless Capture Options in Wireshark.

You are lucky if your native WLAN adapter can capture WLAN Management and Control traffic. In Monitor Mode, you should be able to see traffic from any network as well. To test the capture capabilities of your WLAN Native Adapter, follow the given steps.

Step 1: Launch Wireshark and click the **Interface List** button on the Main Toolbar.

Step 2: Select the checkbox in front of your WLAN adapter and click **Start**.

Step 3: Go to a browser window and visit <u>www.wireshark.org</u>. Go to Wireshark and examine the packets you have captured. If your native adapter is suitable for network capture, you should see some WLAN management and Control traffic (such as Beacon packets and Probe Request/Probe Response packets).



Figure 104.1: Test Your WLAN Native Adapter Capture Capabilities

Also, when you look at the data packets you should see an 802.11 header on the data packets. If your adapter strips off the 802.11 header, Wireshark will apply an Ethernet header. In case, you do not see these traffic types or characteristics, consider another solution for WLAN capture.

Visit: wiki.wireshark.org/CaptureSetup/WLAN for additional options for WLAN capture.

Topic 105: Wireshark Lab 16

This topic provides how to capture to a File Set in High Traffic Rate Situations in Wireshark.

Capturing to file sets is an important task when you are working in high traffic situations. File sets are groups of trace files that are linked based on their file name. In this lab exercise, we will use an autostop condition to only capture three files.

Step 1: Click the Capture Options button on the Main Toolbar.

Step 2: In the Capture Options window, set path and file name for your file set. Enable Use Multiple Files. Set Next file every 1 Minute(s). Set Stop capture after 3 Files. Click Start.

Step 3: Open a browser and visit several web sites. Browse for at least **3 minutes** and go to Wireshark. Your capture process will automatically stop after 3 minutes. The third file will be displayed.

Step 4: To move from one file to the next file in a file set, select **File | File Set | List Files**, select a file in the list and Wireshark will load that file. You can quickly locate specific packets in the file set by applying a display filter to one of the files and then clicking subsequent files in the file list. The display filter will remain in place as you open each file. When you are finished navigating through files, click **Close** on the File Set window.

Topic 106: Wireshark Lab 17

This topic describes how to create and apply a MAC address filter in Wireshark.

Capture filters can reduce the traffic that you need to examine. Here, we will create and use a capture filter based on the MAC address. This will enable us to see all of the traffic to or from our machine.

Step 1: Obtain the MAC address of your host using either *ipconfig* or *ifconfig* as supported by your machine's OS.

Step 2: Click the Capture Options button on the Main Toolbar.

Step 3: Enter **ether host xx:xx:xx:xx:xx** (replacing the **x** indications with your MAC address). Uncheck **Use multiple files**. Click **Start**. If you need this filter again, then click the **Capture Filter** button. Click **New** and name your filter *MyMac* and click **OK**.

Step 4: Open a browser window and visit www.wireshark.org.

Step 5: Go to Wireshark and click the Stop Capture button on the Main Toolbar.

Step 6: Your trace file will contain the HTTP traffic from your browsing session to www.wireshark.org.

Step7: Clear the display filter when you are finished.

Topic 107: Verify the Target Host Traffic

This topic describes how to verify the capture process in Wireshark.

The first step of any analysis process is to verify that the hosts are able to communicate and you can see their traffic in the trace file. If hosts are not able to communicate, there can be several reasons for this.

Check Your Capture Process: If you do not see any traffic, something may have gone wrong during the capture process. A) Wireshark had a capture filter in place and the complaining user's traffic did not match the capture filter. B) If you are using a tap, the user's network cable isn't plugged into the tap. C) If you are spanning a switch port, the switch is not spanning the traffic down your analyzer port.

Consider the TCP/IP Resolution Process: All applications must go through a basic resolution process to build the packet to communicate with another host on a TCP/IP network.

1. Port Resolution: The user specifies Port number in the URL, in the application code.

2. Name Resolution: Examine local cache to locate the information or send a name resolution request on the network.

3. Location Resolution – Local or Remote: Compare target address to local host subnet mask.

4. MAC Address Resolution – Local Target: Examine local cache to locate the target MAC address or send an ARP Request.

5. Route Resolution: Examine local cache to locate the best route for target host.

6. MAC Address Resolution – Remote Target: Examine local cache to locate the target MAC address or send an ARP Request.

Topic 108: Wireshark Lab 18

This topic describes how to identify name Resolution Problems in Wireshark.

If a client does not know the IP address of a target (either in cache or a local hosts file), the client can send a DNS query to obtain this information. What if the client does not know of a DNS server to ask, the client cannot send out a DNS query. The resolution process ends there. If a client sends a DNS query and does not get a response or receives an error in the DNS response, the client can't talk to the target.

Step 1: Open tr-nameresolution.pcapng.

Step 2: Type **dns** in the display filter area and click **Apply**. You can observe from the Status Bar that 32 packets match this filter. From the Info column, you can see a number of **No Such Name** responses indicating the name resolution process failed. This can be due to DNS issues or user issues. In this trace file, the user typed the wrong URL.

Step 3: Clear the display filter when you are finished.

Topic 109: Wireshark Lab 19

This topic describes how to identify local address Resolution Problem in Wireshark.

Before a client can send a packet to a local target or a local router, it must obtain the MAC (Media Access Control) address of that local target or router. If the client does not have the MAC address information in cache, the client sends out an Address Resolution Protocol (ARP) request. If no response is received when trying to acquire the local target's MAC address, the client is done. It cannot send out a packet to the target.

Step 1: Open tr-localresolution.pcapng.

Step 2: Scroll through this trace file. Look at the ARP requests sent to discover the MAC address of 192.168.1.45. There are no responses.

d ti	-localresolution	on.pcapng										
File	<u>E</u> dit <u>V</u> iew	<u>G</u> o <u>C</u> apture	<u>A</u> nalyze <u>S</u> tatisti	cs Telephony <u>T</u> oc	ls Internals	<u>H</u> elp		You	should never	have to	ask for	
0	• 🧉 🔳	 B B	X 🔁 🔍	🗢 🛸 📦 🖥 🛓		Ð D	Q	any	rthing twice o	n the ne	twork	
F	ilter:				•	Expression	on	Clear	Apply	Save		
No.	Time	Source		Destination	Protocol	Info						
1	0.000	Hewlett-	_a7:bf:a3	Broadcast	ARP	Who	has	192.	168.1.45?	Tell	192.168	.1.72
2	0.863	Hewlett-	_a7:bf:a3	Broadcast	ARP	Who	has	192.	168.1.45?	Tell	192.168	.1.72
3	0.999	Hewlett-	_a7:bf:a3	Broadcast	ARP	Who	has	192.	168.1.45?	Tell	192.168	.1.72
4	1.002	Hewlett-	_a7:bf:a3	Broadcast	ARP	Who	has	192.	168.1.45?	Tel1	192.168	.1.72
5	0.997	Hewlett-	_a7:bf:a3	Broadcast	ARP	Who	has	192.	168.1.45?	Tell	192.168	.1.72
6	1.000	Hewlett-	a7:bf:a3	Broadcast	ARP	Who	has	192.	168.1.45?	Tell	192.168	.1.72
7	1.001	Hewlett-	a7:bf:a3	Broadcast		Who	has	192.	168.1.45?	Tell	192.168	.1.72
8	0.998	Hewlett-	27.hf. 2	Buandene	400	Who	has	192.	168.1.45?	Tell	192.168	.1.72
9	1.000	Hewlett-	If the client	can't resolve th	ne MAC	Who	has	192.	168.1.45?	Tell	192.168	.1.72
10	1.001	Hewlett-	address	for 192 168 1 4	5 it	Who	has	192.	168.1.45?	Tell	192.168	.1.72
11	0.998	Hewlett-	aannat	and a neekati	a 14	Who	has	192.	168.1.45?	Tell	192.168	.1.72
12	1.000	Hewlett-	cannot	senu a packet	IO IL	Who	has	192	168.1.45?	Tell	192.168	.1.72
13	4 464	Jonletto	Ribtian	Buorleast	ARRAA	Wha	hm	1.02	16m11452	- John	12.168	th Zan

Figure 109.1: Find Local Address Resolution Problems

Without resolving MAC address issue, 192.168.1.72 will not be able to communicate with 192.168.1.45. Maybe 192.168.1.45 is not currently up. Maybe that is the wrong IP address. We can't tell.

Topic 110: Wireshark Lab 20

In this topic, we describe the "No server Response to TCP Connection Request" situation in Wireshark.

There are several reasons why a server may not respond to a TCP connection attempt. The TCP handshake request packet (SYN) may not arrive at the server. Maybe the SYN packet was lost. A firewall along the path dropped the SYN packet, or a host-based firewall on the server blocked access to the port. Alternately, the response (SYN/ACK) may not arrive at the client. The SYN/ACK packet was lost along the path, or a firewall along the path blocked the SYN/ACK.

Step 1: Open tr-noserver.pcapng.

Step 2: Scroll through this trace file. This trace file contains only SYN packets from 192.168.1.72 to 192.168.1.66. None of the SYN packets have received SYN/ACK responses. Expand the TCP header in Packet 1. Right-click on the **[Stream index: 0]** line and select **Apply as Column**. Wireshark shows that there are 6 separate connections (TCP Stream Index: 0-5).

U	nuser	rer.pci	sping				19 B								
Eile	Edit	View	<u>G</u> o	Capture	Analyze	Statistics	Telephony	Tools	Internals	Help					
0	•	(=	1	8	* 2	19.4		7 1		0	0		0 🕵 🕯		1
Fi	iter:								- E	Expression	Clear	Ap	ply	Save	T
ю,	Time		Source			Destination			Protocol	Stream in	idex So	urce Port	Info		1
1	0.0	000	192	.168.	1.72	192.16			TCP			5538	5538		tt
2	2.0	042	192	168.	1.72	192.16	8.1.66		TCP	1		5537	5537	> h	tt
3	0	249	192	168.	1.72	192.16	8.1.66		TCP	0		5538	[TCP	Ret	rar
4	5.7	750	192	.168.	1.72	192.16	8.1.66		TCP	1		\$\$37	[TCP	Ret	ra
5	0.1	249	192	168.	1.72	192.16	8.1.66		TCP	0		5538	[TCP	Ret	ra
5	11.	. 327	192	.168.	1.72	192.16	8.1.66		TCP	2		5539	5539	> h	tt
7	0.7	251	192	.168.	1.72	192.16	8.1.66		TCP	3		5540	5540	> h	tt
3	0.4	425	192	.168.	1.72	192.16	8.1.66		TCP	4		5541	5541	> h	tti
)	2.	321	192	168.	1.72	192.16	8.1.66		TCP	2		5539	[TCP	Ret	rai
10	0.1	257	192	.168.	1.72	192.16	8.1.66		TCP	3		5540	[TCP	Ret	ra
11	0.4	420	192		The				TOD	4		5541	[TCP	Ret	rar
12	5.	314	192	1	Ine	client atte	empts nu	merou	SILP	2		5539	[TCP	Ret	ran
13	0.7	261	192	.1	CO	nnection	s on diffe	rent p	orts	3		5540	[TCP	Ret	rar
14	0.4	424	192	108	1.72	192.10	8.1.00		The T	4		5541	[TCP	Ret	rar
15	11.	. 580) 192	168.	1.72	192.16	8.1.66		TCP	- 5	8	5545	5545	> h	ttr
16	2.9	999	192	168	1.72	192.16	8.1.66		TCP	5		5545	[TCP	Ret	rar
7	5.0	996	192	168.	1.72	192.16	8.1.66		TCP	5		5545	[TCP	Ret	na

Figure 110.1: No Response to TCP Connection Request

Step 3: Right-click on the TCP **Source Port** field in any packet and select **Apply as Column**. This shows that the client has set up numerous ports for these connections.

Step 4: Some of these SYN packets match the HTTP coloring rule while others match the Bad TCP coloring rule because the SYN packets are Retransmissions.

Topic 111: Wireshark Lab 21

This topic discusses how to analyze the "No Response to Service Request" situation in Wireshark.

Step 1: Open tr-serverresponse.pcapng.

Step 2: Scroll through this trace file to get familiar with the traffic pattern. Use the Stream Index column to differentiate the separate connections bet. 24.6.173.220 and 50.62.146.230. Right-click on Packet 1 in the Packet List pane and select **Conversation Filter | TCP**.

Eile Fi	Edit ()	View (ip.a	<u>G</u> o Ø ddr eq 2	Capture	Analyze	Statistics	Telephony	Loois Int Loois Int Loois Int I (tcp.port eq	The HTTF send a prop	o port is open, response coo perly - this is a	but tl le - T(n app	he serv CP is pe lication	er doe: erformi i issue	s not ng
Vo.	Time		Source	e		Destination		Protocol	Stream index	Info		CO.V.	7	
l	0.0	000	24.	6.173.	.220	50.62.	146.230) TCP	0	44043 > h	ttp	[SYN]	1/	15658
)	0.0	035	50.	62.140	5.230	24.6.1	73.220	TCP	0	http > 44	043	[SYN	K]	Seq=
3	0.0	000	24.	6.173.	220	50.62.	146.230) TCP	0	44043 > h	ttp	[ACK	Seq=	15658
1	0.0	000	24.	6.173.	220	50.62.	146.230) HTTP	0	GET / HTT	P/1.	1		
5	0.0	036	50.	62.146	5.230	24.6.1	73.220	TCP	0	http > 44	043	[ACK]	Seq=	15884
5	7.0	631	24.	6.173.	220	50.62.	146.230) TCP	0	44043 > h	ttp	[FIN,	ACK]	Seq=
1	0.0	075	50.	62.146	5.230	24.6.1	73.220	TCP	0	http > 44	043	[ACK]	Seq=	15884
21	110	9.99	24.	6.173.	220	50.62.	146.230) TCP	0	44043 > h	ttp	[RST,	ACK]	Seg=

Figure 111.1: No Response to Service Request

The TCP handshake completes successfully in Packets 1-3. The client requests the default file from the web site's root directory in Packet 4. Packet 5 contains acknowledgment no. 288, which indicates that the server has received every sequence no. up to 287, and it expects sequence no. 288 next. So, server received the request. Packet 5 contains acknowledgment no. 288, which indicates that the server has received every sequence no. up to 287, and it expects sequence no. 288 next. So, server received the request. So, server received the request. The client's browser appears to time out and sends a FIN/ACK after almost 8 seconds. The client begins an implied connection termination. The server sends an ACK. We would expect the server to begin closing its side of the connection, but it does not. The client waits almost 120 seconds before sending a RST/ACK. TCP appears to be functioning properly in this trace file. The symptoms indicate the application has failed at the server side.

Topic 112: Do not Focus on Acceptable Delays

This topic discusses the worth of normal delays.

You can safely ignore some delays in your trace files. For example, a delay before a TCP RST packet would likely not be felt by the end user. Such acceptable delays should not raise an alarm.

Delays before DNS Queries

A DNS query will be triggered when we enter a URL in a browser and then press Enter. When we see an interesting link on a web page and we click on it. Open *trdelays.pcapng*.

You will find that the delay before Packet 29 is caused by an eventual time out of a connection. Packet 32 is a DNS query and the delay before it is because the user did not straightaway click on <u>www.wireshark.org</u>.

Delays before TCP FIN or Reset Packets

Delay introduced by an application, which sends TCP FIN/RST packets to close the connection after waiting for specified time or some task is completed.

Delays before a Client Sends a Request to a Server

Filling out a form and pressing the Submit button, or clicking the next link on a web page. In all such cases, an application requires user interaction.

Delays before Keep-Alive or Zero Window Probes

Keep-Alives and Zero Window packets are sent during a Zero Window situation to determine if more buffer space is available at the target. Hosts also send "Keep-Alives" periodically to maintain a TCP connection.

Delays before TLS Encrypted Alert

An application sends a Transport Layer Security (TLS) Encrypted Alert to eventually close an encrypted connection; we can't see the Close command.

Delays before a Periodic Set of Packets in a Connection that is Otherwise Idle

Applications can define their own keep alive packet types to keep the connection active.

Topic 113: Watch for the Delays that DO Matter

This topic describes various delays that do matter for networks.

On your network, knowing what "normal" delay times are will help you identify unusually high delay times. We next provide a list of delays that should be examined.

Delays before a Server Responds with a SYN/ACK

A client can use the time between the SYN and SYN/ACK in the TCP handshake to determine the round trip time between the hosts. A large delay before the SYN/ACK is an indication of a high round trip time between the hosts.

Delays before a Client Completes the 3-Way TCP Handshake

Time between SYN/ACK and client's ACK to finish TCP handshake can be used by server to determine the round trip time.

Delays before a Server Sends a Response

In a trace file, if you observe that a server quickly sent an ACK to a client's request, but there is a long delay before server's response, then this is not path latency issue. We need to consider why a server might be slow. For example, let's look at the case given in *tr-http-pcaprnet101.pcapng*. The client's GET request (GET /home) in packet 18 followed by the server ACK takes 17 ms. We can see a delay of almost 1.8 seconds before the requested data begins. The trace was taken at the client. The round trip time between the hosts is acceptable, but the server response time is not.

Delays before the Next Packet in a Data Stream

There can be several reasons for delays that occur during a file download or file upload process. The sender became busy with other processing. There can be a

device along the path, which may be buffering the data for higher priority traffic. Lack of receive buffer space at a receiver.

Delays before an ACK from a TCP peer

This situation arises when ACKs from a receiver get delayed because of the path latency or delayed ACK function. Delayed ACK timers are often set at 200 ms.

Delays before a Window Update

A host must wait for a Window Update before sending a packet, if it finds that Window Size advertised is too small to fit a full-sized data segment.

Topic 114: Wireshark Lab 22

In this topic, we discuss how to detect delays in UDP Conversations in Wireshark.

The User Datagram Protocol (UDP) is a connectionless transport layer protocol with an 8-byte header. Unlike TCP, UDP has no sequencing or acknowledgement capability. Delays between requests and responses are a measure for UDP-based applications. UDP conversation statistics such as packet rate, bps rate etc. can be obtained with Conversations Window.

Step 1: Open tr-voip-extensions.pcapng.

Step 2: Select Statistics | Conversation.

Step 3: Click the UDP tab. Uncheck the Name Resolution option, if you are interested in seeing port numbers rather than port names.

Step 4: Click twice on the **bps A** -> **B** column heading. This will sort UDP conversations based on traffic flowing from Address A/Port A to Address B/Port B. In the list, Wireshark will view the conversation between 192.168.5.11/port 25426 and 192.158.5.10/port 8000 first.

culenes 2	rible chai		V4. 2 11		ALINCE	
Address A 4	Port A 4	Address B 4	Port B 4	Packets 4	Bytes 4	Packets A→B ◀
192.168.5.11	25426	192.168.5.10	8000	575	117 426	289
192.168.5.10	5060	192.168.5.11	5060	2	1 973	2
192.168.5.11	25427	192.168.5.10	8001	5	542	5
192.168.5.10	5060	192.168.5.11	1073	12	6 1 5 2	0
192.168.5.11	7818	192.168.5.10	5060	8	4 1 2 3	0
192.168.5.20	5060	192.168.5.11	1033	3	2 254	0
192.168.5.11	7818	192.168.5.20	5060	4	2 471	0
Nama	-Lution II	🗏 Limitto dia	n lav filta			

Figure 114.1: Obtain UDP Conversation Statistics and Filter on a UDP Conversation

Step 5: This conversation is also the most active UDP conversation in terms of the Bytes column value. Right–click on this conversation line and choose Apply as Filter |
Selected | A <-> B.

Step 6: Select File | Export Specified Packets and name your new file *udpconv1.pcapng*. When you are done, clear the display filter and close the Conversations window. Removal of unrelated traffic from view, allows Wireshark to list down statistics related to that traffic set. This makes identification of performance issues easier.

Topic 115: Wireshark Lab 23

This topic describes how to Add/Sort a Delta Time Column in Wireshark.

In Wireshark, the default Time column setting is **Seconds Since Beginning of Capture**. It becomes easier to locate delays when a time column displays delta times. We can locate the largest delays in a trace file by sorting the delta time column.

Step 1: Open tr-malaysianairlines.pcapng

Step 2: Expand the Frame section of any packet.

Step 3: Right-click on the "Time delta from previous captured frame" line and select **Apply as Column**. This will create a **frame.time_delta** column.

Step 4: The new column appears to the left of the Info column. Click and drag it to the right of the existing Time column.

Step 5: Rename the column by right-clicking on the column heading and selecting **Edit Column Details**. Let's change the Title to **Delta**. Click **OK**.
Step 6: Click your new Delta column heading twice to sort from high to low. Go to the first packet. You can see packets with the largest delays at the top of the list.

0 🛞	/ = <u>/</u> = =	X 🕄 🔍 🗢 🗢 🖥	Expression	be careful by "r	normal" delays
No.	Delta	Source	Destination	Protocol	Info
1296	3.703247000	66.235.138.19	192.168.1.72	TCP	http > 18204 [RST,
1285	3.613323000	50.116.194.21	192.168.1.72	TCP	http > 18195 [FIN,
594	2.904176000	63.234.226.59	192.168.1.72	HTTP	HTTP/1.1 302 Moved
1283	0.524120000	192.168.1.72	192.168.1.254	DNS	Standard guery 0x1
2	0.492624000	192.168.1.254	192.168.1.72	DNS	Standard query res
598	0.421065000	63.234.226.59	192.168.1.72	HTTP	НТТР/1.1 200 ОК (
1281	0.366142000	192.168.1.254	192.168.1.72	DNS	Standard query res
11	0.199064000	63.234.226.59	192,168,1,72	HTTP	HTTP/1.1 200 OK (
-	0 105000000	C2 324 336 E0	102 100 1 72		······
Fra	me 1296: 60 b	oytes on wire (480	bits), 60 byte	es capt	ured (480 bits) on

Figure 115.1: Add/Sort a Delta Time Column

In this trace file, delays such as delays before DNS queries, TCP RST packets, or TCP FIN packets are "normal". We do not care about them. Focus on delays before DNS and HTTP responses.

Topic 116: Wireshark Lab 24

This topic describes how to add/sort a delta displayed time column in Wireshark.

Delays between displayed packets can be identified by using a delta displayed time column. Let's create such a time column to show the delta times of DNS traffic only.

Step 1: Open tr-malaysianairlines.pcapng.

Step 2: Click the **Preferences** button on the Main Toolbar and then select Columns.

Step 3: Click the **Add** button. In the drop-down "Field Type" list, select *Delta time displayed*. Click on the column name and update the name to **Delta Displayed**.



Figure 116.1: Add/Sort a Delta Displayed Time Column

Step 4: Click and drag your **Delta Displayed** column above the "Source" column. Click **OK**. If desired, right click on your newly created column heading to set left alignment.

Step 5: Now apply a filter for DNS traffic. For this purpose, enter **dns** in the display filter area. Click **Apply.**

Step 6: Clicking twice on your new **Delta Displayed** column will sort from high to low. Let's not focus on delays before DNS queries, and care about delays before DNS query responses.

4	1	tr-malaysianairlines.pcap	ng [Wireshark 1.12.3 (v1.12.3-)-gbb3e9a0 from mas
[ile [dit View Go Capture Analyze	<u>Statistics</u> Telephony <u>T</u> o	ols <u>i</u> nternals <u>il</u> elp		
0 0 1 1 1 1 x 2	9 0, 4 4 4 🖗 7	🛓 🔳 🖬 Ə, Ə, G) 🖂 🖉 🕅 🥵 🛠 🗎	
Filter: dns		✓ Expression Cle	ar Apply Save TCP-HS	
No. Delta Delta Displayed 🔻	Source De	stination Length	Protocol Info	
942 0.0002584.294	192.168.1.72 1	92.168.1.25 88	DNS Standard query	0x2f26 A 232
17 0.0010810.614	192.168.1.72 1	92.168.1.25 76	DNS Standard query	Oxf2dc A s3.
1283 0.5241200.524	192.168.1.72 1	92.168.1.25 78	DNS Standard query	0x10c7 A www
2 0.4926240.492	192.168.1.254 1	92.168.1.72 306	DNS Standard query	response Ox8a
1229 0.0078310.396	192.168.1.72 1	92.168.1.25 76	DNS Standard query	0x533f A www
1281 0.3661420.366	192.168.1.254 1	92.168.1.72 222	DNS Standard query	response OxbO
365 0.020205 0.359	192.168.1.72 1	92.168.1.25 88	DNS Standard query	Oxflc6 A mal
410 0.0014790.306	192.168.1.72 1	92.168.1.25 89	DNS Standard query	0xdbc1 A d3e
1139 0.0012880.256	192.168.1./2 1	92.168.1.25 88	DNS Standard query	0x9c36 A mal
1257 0.0178620.235	192.168.1.254 1	92.168.1.72 104	DNS Standard query	response Oxbr
1/7 0 0010030 118	102 168 1 72 1	07 168 1 75 88	NNS Standard many	0v16d7 A 232

We find that two packets (Packet 2 and Packet 1,281) are really very slow. **Cause for Delays:** If a local DNS server does not have these names in its cache, it needs to perform recursive queries to obtain the data.

Topic 117: Wireshark Lab 25

In this topic, we describe how to plot UDP Delays in Wireshark.

Using Wireshark's Advanced IO Graph with a filter and a reference to the maximum **frame.time_delta_displayed** value, we can have a pictorial idea about the delays in a trace file.

Step 1: Open tr-queuing.pcapng.

Step 2: Select Statistics | IO Graph.

Step 3: In the Y Axis Unit area, select Advanced...

Step 4: As this trace file contains less than 2 seconds of traffic. Select **0.01 sec**, in the X Axis Tick Interval area.

Step 5: Select the **MAX(*)** Graph 1 Calc option and enter **frame.time_delta_displayed** in the Calc area.

Step 6: Click the **Graph 1** button to plot the results. If your trace file contains both UDP and TCP-based traffics and you want to plot UDP delays, then enter **udp** in the Graph 1 filter area before you click the Graph 1 button.

					A	- 5
0.00s « Graphs	0.20s 0.40s	0.60s 0	1.00s m	1.20s	1.40s	1.60s 1.80s
		And an other states of the state of the stat				Tick intervals 0.01 cec
Graph1 C	olor Filter:	Calc: MAX(*)	 frame.time_delta_dis 	played Style: Line	 Smooth 	TICK IIILEIVAL 0.01 SEC
Graph 1 C	olor Filter:	Calc: MAX(*) Calc: SUM(*)	 frame.time_delta_dis 	played Style: Line Style: Line	 Smooth Smooth Smooth 	Pixels per tick: 5
Graph 1 C Graph 2 C Graph 3 C	olor Filter: olor Filter: olor Filter:	Calc: MAX(*) Calc: SUM(*) Calc: SUM(*)	frame.time_delta_dis v	Style: Line Style: Line Style: Line	Smooth Smooth Smooth Smooth Smooth Smooth	Pixels per tick: 5
Graph 1 C Graph 2 C Graph 3 C Graph 4 C	olor Fjiter: olor Fjiter: olor Fjiter: olor Fjiter:	Calc: MAX(*) Calc: SUM(*) Calc: SUM(*) Calc: SUM(*)	frame.time_delta_dis	style: Line Style: Line Style: Line Style: Line Style: Line	V Smooth V Smooth	Pixels per tick: 5

There is a sudden increase in the delta time at approx. 1.2 seconds of the trace file. On clicking these points, Wireshark jumps to that point in the trace file and enables to do additional analysis.

Topic 118: Wireshark Lab 26

This topic describes how to obtain TCP conversation statistics in Wireshark.

Transmission Control Protocol (TCP) is a connection-oriented protocol. TCP is a stream based communication. Each separate TCP conversation is numbered by Wireshark with a TCP Stream Index (**tcp.stream**) value starting with 0. Let's find the

most active TCP conversation. If you have a large trace file, you can apply a filter from within Conversations window.

Step 1: Open tr-chappellu.pcapng.

Step 2: Select Statistics | Conversations and click the TCP tab.

Step 3: Click twice on the Bytes column to sort from high to low.

Ethernet: 1	Fibre Channe		v4: /	IPV0 IPX	JATA N	CP KSVP	SCIP	TCP: 23
Address A	◆ Port A	Address	B ◀	Port B 4	Packets	 Bytes 	▼ Pa	ackets A→B
24.6.173.220	35642	207.171.	187.117	http	12	27 126	605	4
24.6.173.220	35643	207.171.	187.117	http	12	22 119	496	3
24.6.173.220	35641	207.171.	187.117	http	8	85 85	381	2
24.6.173.220	35644	207.171.	187.117	http	5	56 54	884	1
24.6.173.220	35640	207.171.	187.117	http	5	54 49	399	1
24.6.173.220	35629	184.73.2	50.227	http	6	6 26	339	34
24.6.173.220	35623	69.59.18	0.202	http	2	22 12	078	1
24.6.173.220	35627	141.101.	125.193	http	1	.4 9	402	1
Name res	olution 📃	Limit to di	splay filt	ter				

Figure 118.1: Obtain TCP Conversation Statistics

Step 4: Right-click on the top entry and select **Apply as Filter | Selected | A <-> B.** Wireshark creates a filter based on the source/destination address and source/destination port fields.

								TCP Conv	ersations	
Address A 🕯	Port A 🕻	Address B 📢	Port B 🕈 Pa	ckets €	Bytes 🔻	Packets A-	+B 🕇 Byte	s A→B ∢	Packets A←B ◀	Bytes A
24.6.173.220	35642	207.171.187.117	80	127	126 605		41	3 703	86	1
24.6.173.220	35643	207.171.187.117	80	122	19 496		38	3 0 1 5	84	1
24.6.173.220	35641	207.171.187.117	80	85	85 381		27	2 421	58	
24.6.173.220	35644	207.171.187.117	80	56	54 884		18	1 409	38	
2/1 6 172 220	25640	207 171 197 117	<u> </u>	5/	/0 200		10	1 025	26	
I	CP Conve	ersations								
; A→B ◀ Bytes	A→B ◀ F	Packets A⊢B ◀	Bytes A⊢B ◀	Rel Sta	rt 📢 🛙)uration (bps A→B	↓ bps A+	-8	1
	3 703		122 902	32.822	986000	1.5346			640685	.19
38	3 0 1 5	84	116 481	32.827	897,000	1.1597	20798	.30	803518	.12
27	2 421	58	82 960	32.613	12/000	1,2735	15208	.74	521155	45
18	1 409	38	53 475	32,860	271000	1.2247	9204	.26	349324	.30
18	1 935	36	47 464	32,546	263000	1.1913	12994	.01	318732	.70

	tr-chappellu.pcapng [Wireshark 1.12.3 (v1.12.3-0-gbb3e9a0 from master-1.12)]									
<u>F</u> ile	<u>E</u> dit	<u>V</u> iew <u>G</u> o	<u>C</u> apture <u>A</u> nalyze <u>S</u> tatisti	ics Telephon <u>y</u> <u>T</u> ools <u>I</u> nter	nals <u>H</u> elp					
0	0		⊨) x ∂ Q	💩 🔿 🛪 🕹 🔳		M 🛛 🖪 💥	B			
ř										
Fi	ter:	ip.addr==24	4.6.173.220 && tcp.port==356	42 && ip.addr==207.171.187.1	 Expression Clear 	Apply Save	TCP Dealy			
No.		Time	TCP Delta	Source	Destination	Length Protocol	Stream index Info			
	216	32.822	0.00000000	24.6.173.220	207.171.187.117	66 TCP	20 35642→80	[SYN] Seq=0 V		
	220	32.857	0.034746000	207.171.187.117	24.6.173.220	66 TCP	20 80→35642	[SYN, ACK] Se		
	221	32.857	0.000197000	24.6.173.220	207.171.187.117	54 TCP	20 35642→80	[ACK] Seq=1 /		
	222	32.858	0.001048000	24.6.173.220	207.171.187.117	479 HTTP	20 GET /publ	ic/5tort/1/p		
	231	32.934	0.075583000	207.171.187.117	24.6.173.220	60 TCP	20 80→35642	[ACK] Seq=1 /		
	321	33.372	0.437920000	207.171.187.117	24.6.173.220	1514 HTTP	20 HTTP/1.1	200 OK (JPE)		
	322	33.372	0.000004000	207.171.187.117	24.6.173.220	1514 TCP	20 80→35642	[ACK] Seg=140		
	323	33.372	0.000308000	24.6.173.220	207.171.187.117	54 TCP	20 35642→80	[ACK] Seq=426		
	324	33.373	0.000747000	207.171.187.117	24.6.173.220	1514 TCP	20 80→35642	[ACK] Seq=292		
	345	33.423	0.049643000	207.171.187.117	24.6.173.220	1514 TCP	20 80→35642	[ACK] Seq=438		
	346	33.423	0.000245000	24.6.173.220	207.171.187.117	54 TCP	20 35642→80	[ACK] Seq=420		
	347	33.424	0.000855000	207.171.187.117	24.6.173.220	1514 TCP	20 80→35642	[ACK] Seq=584		
	348	33.424	0.000002000	207.171.187.117	24.6.173.220	1514 TCP	20 80→35642	[ACK] Seq=730		
	2/10	22 121	0 000128000	2/ 6 173 220	207 171 187 117	5/1 TCD	20 356/12-380	[ACK] Son-126		

Step 5: Click **Clear** to remove your filter when you are finished. If there are many TCP conversations contained in your trace file, the method we learnt in this topic can be used to find the most active conversation and then quickly apply a filter on that conversation.

Topic 119: Wireshark Lab 27

In this topic, we describe how to filter TCP conversation with the stream index field in Wireshark.

What is a Stream?

Wireshark assigns each unique TCP connection attempt a separate TCP stream based on the source/destination addresses and source/destination port numbers. Let's create a filter based on the **tcp.stream** field.

Step 1: Open tr-chappellu.pcapng.

Step 2: Expand the TCP header in Packet 59.

Step 3: Right-click on the **[Stream index: 7]** field in the TCP header. Select **Apply as Filter | Selected**.

eliter: tcp.stream == 7	Expression Clear Apply Save	TCP-HS
No. Time Delta Delta Displayed	Source Destination Length Protocol	Info
598.677 0.000180.000	24.6.173.220 184.73.250. 66 тср	35629→80 [SYN] Seq=0 \
70 8.772 0.06178 0.094	184.73.250.2224.6.173.22 66 TCP	80→35629 [SYN, ACK] S
718.772 0.00032.0.000	24.6.173.220 184.73.250. 54 TCP	35629→80 [ACK] Seq=1 /
76 8.818 0.00880 0.045	24.6.173.220 184.73.250. 654 HTTP	GET /documents.json?k
85 8.913 0.04028 0.095	184.73.250.2224.6.173.22 60 TCP	80→35629 [ACK] Seg=1 /
89 9.153 0.13256 0.240	184.73.250.2224.6.173.22 885 HTTP	HTTP/1.1 200 OK (app
90 9.353 0.19953 0.199	24.6.173.220 184.73.250. 54 TCP	35629→80 [ACK] Seg=60:
112 10.7770.15622 1.424	24.6.173.220 184.73.250. 654 HTTP	GET /documents.json?ke
113 10.8720.094270.094	184.73.250.2224.6.173.22 60 TCP	80→35629 [ACK] Seg=83
114 10,9210,04991,0,049	184,73,250,2224,6,173,22 884 HTTP	HTTP/1.1 200 OK (app
115 11.1280.20632 0.206	24.6.173.220 184.73.250. 54 TCP	35629→80 [ACK] Seg=120
<		
Frame 59: 66 bytes on wire	e (528 bits), 66 bytes captured (528	bits) on interface 0
Ethernet II, Src: Hewlett-	_a7:bf:a3 (d4:85:64:a7:bf:a3), Dst:	Cadant_31:bb:c1 (00:0
Internet Protocol Version	4, src: 24.6.173.220 (24.6.173.220)	, Dst: 184.73.250.227
Transmission Control Proto	ocol, Src Port: 35629 (35629), Dst P	ort: 80 (80), Seq: 0,
Source Port: 35629 (35629	9)	
Destination Port: 80 (80)		
[Stream index: 7]		
[TCP Segment Len: 0]		
Sequence number: 0 (re	elative sequence number)	
and the second		
· · · · · ·		
	. <u>.</u>	
0000 00 01 5c 31 bb c1 d4	85 64 a7 bf a3 08 00 45 00\1.	dE.
0010 00 34 29 99 40 00 80	06 00 00 18 06 ad dc b8 49 .4).@	I
0020 fa e3 8b 2d 00 50 3f	97 e1 9d 00 00 00 00 80 02	P?
0030 20 00 79 36 00 00 02	04 05 b4 01 0 <u>3 03 02 01 01</u> v6.	
File: "C:\Users\kashif\Desktop\Teaching\VU\inc	daa\my teaching\slides\Tra Packets: 724 · Displayed: 66 (9.1%) · Load	me: 0:00.015

Figure 119.1: Filter on a TCP Conversation Using the Stream Index Field

We can see that Wireshark creates a filter for **tcp.stream==7** in the filter display area and applies it to the trace file. There are 66 packets matching this filter as indicated on the Status Bar. You can save this TCP conversation in a separate trace file. Select **File | Export Specified Packets** and provide a file name.

Step 4: Once you are finished, click Clear to remove the filter.

Topic 120: Wireshark Lab 28

This topic shows how to add a TCP stream index column in Wireshark.

There can be numerous TCP conversations in a trace file. Wireshark calls each unique connection attempt a TCP stream. To differentiate the TCP conversations, Wireshark assigns each unique connection attempt a separate TCP stream index based on the source/destination addresses and source/destination port numbers. Wireshark numbers each separate TCP conversation with a TCP Stream Index (**tcp.stream**) value starting with 0. Next, we show that packet, which is part of the first TCP conversation in a trace file.



Figure 120.1: Add a TCP Stream Index Column

Step 1: Open tr-chappellu.pcapng.

Step 2: Expand the TCP header in Packet 1. Right-click on the **[Stream index: 0]** line and select **Apply as Column**.

<u>File Edit View Go C</u> a	apture <u>A</u> nalyze <u>S</u> tatistics Telephony <u>T</u> ools	Internals <u>H</u> elp		
•••	Expand Subtrees	- 🗖 🖬 । €, Q, Q, 🖭	🏽 🖻 😽 💥	I
Filter:	Collapse Subtrees	✓ Expression Clear	Apply Save	TCP Dealy
No. Time	Expand All	Destination	Length Protocol	Info
$1 \ 0.000$	Collapse All	198.66.239.146	66 TCP	35621→80 [SYN] Se
2 0.251	Apply as Column	75.75.75.75	77 DNS	Standard query 0x
31.252	Annalysis Eller	75.75.76.76	77 DNS	Standard query 0
41.253	Apply as Filter	75.75.75.75	77 DNS	Standard query 0x
5 2.252	Prepare a Filter	75.75.76.76	77 DNS	Standard query 0>
6 2.252	Colorize with Filter	75.75.75.75	77 DNS	Standard query 0
7 3.007	Follow ICP Stream	198,66,239,146	66 TCP	[TCP Retransmissi
8 4, 252	Follow UDP Stream	75,75,75,75	77 DNS	Standard query 0
9.4.252	Follow SSL Stream	75 75 76 76	77 DNS	Standard query ()
10.8 253	Сору	75 75 75 75	77 DNS	Standard query 0
11 8 253	Export Selected Packet Bytes	75 75 76 76	77 DNS	Standard query 0
12.8.208	Edit Packet	24 6 173 220		Standard query of
12 0.290	Wiki Protocol Page	108 66 220 146	95 DN3	
14.9.217	Strate State (190.00.239.140	00 TCP	33022+00 [STN] SE
< 1/1 × 317	Filter Field Reference	p // p 1/3 //0	nn 11 U	
Erame 1.66 h	Protocol Help	hytes cantured (5)	28 hits) on t	interface 0
Ethornot II	Protocol Preferences	(5:64:a7:hf:a2)	st: Cadant 31	(
Ethernet II, Totomot Doot	दे Decode As	12 220 (24 6 172 2	30) Dc+, 100	1.00.CI (00.01.JC.
Thernet Prot	Disable Protocol	3.220 (24.0.1/3.2	20), DSL. 190	3.00.239.140 (198.0
I transmission	Resolve Name	33021 (33021), US	L POPL: 80 (a	50), Seq: 0, Len: 0
Source Port	Go to Corresponding Packet			
Destination	Show Packet Reference in New Window			
[Stream inde	Show Facket hereference in New Window			

Step 3: Click on your **Stream index** column once to sort the trace file by conversations. Jump to the end of the trace file and you find that there are 23 TCP conversations. Counting TCP streams starts at 0.

								t	r-chapp	ellu.pca	png (V	Vireshar	k 1.12.3	(v1.12.3-	0-gbb	3e9a0 f	rom ma	ster-1.12)]	
Eik	<u>E</u> di	t <u>V</u> iew	<u>G</u> o	<u>Capture</u>	<u>A</u> nalyze	$\underline{S} tatistics$	Telephony	Tools	Internals	<u>H</u> elp									
0	۲		₫	6	X 🕄	0, ¢	🖻 🌍 🖥	F 🕹		⊕(ર્ ભ્	E 🖉	i 🗵 🍯	% 5					
Fi	ter:								✓ Exp	ression	Clear	Apply	Save	TCP-HS					
No.		Time	0)elta	Delta	Displayed	Source		Desti	nation		Length	Protocol	stream in	ndex 🔻	ofo			
	693	34.	084(0.0001	17:0.0	000	24.6.1	.73.2	20 20	7.171	187	54	TCP	(22	3564	4→80	[ACK]	Seq=4
	692	34.	084(0.000	0.0.0	000	207.17	1.18	7.124	.6.17	3.22	359	TCP		22	80→3	5644	[PSH,	ACK]
	691	. 34.	084(0.0058	32:0.0)05	207.17	1.18	7.124	.6.17	3.22	1514	TCP		22	80→3	5644	[ACK]	Seq=4
	690	34.	078(0.0001	130.0	000	24.6.1	.73.2	20 20	7.171	187	54	TCP		22	3564	4→80	[ACK]	Seq=4
	689	34.	078(0.000	0.0.0	000	207.17	1.18	7.124	.6.17	3.22	1514	TCP		22	80→3	5644	[ACK]	Seq=4
	688	34.	078(3000.0	39:0.0	000	207.17	1.18	7.124	.6.17	3.22	1514	TCP		22	80→3	5644	[ACK]	Seq=4
	687	34.	077(0.0027	/10.0)02	207.17	1.18	7.124	.6.17	3.22	1514	TCP		22	80→3	5644	[ACK]	Seq=4
	686	34.	075(0.0001	L8 0.0	000	24.6.1	.73.2	20 20	7.171	187	54	TCP		22	3564	4→80	[ACK]	Seq=4
	685	34.	074(0.000	0.00	000	207.17	1.18	7.124	.6.17	3.22	1514	TCP		22	80→3	5644	[ACK]	Seq=4
	684	34.	074(0.0008	3210.0	000	207.17	1.18	7.124	.6.17	3.22	1514	TCP		22	80→3	5644	[ACK]	Seq=4
	683	34	0741	0 0001	90.0	000	24 6 1	73 2	20.20	7 171	187	54	TCP		22	3564	4→80	[ACK]	Son=4

Figures and Material used for Part2 (from Topics 121 to 172) have been adapted from "Troubleshooting with Wireshark: Locate the Source of Performance Problems", 2014, by L. Chappell.

Topic 121: Wireshark Lab 29

This topic describes how to add/sort a TCP Delta Time column in Wireshark.

The **tcp.time_delta** shows the time from the end of one packet in a TCP stream to the end of the next packet in that same TCP stream. To add the **tcp.time_delta** column, you must enable the TCP *Calculate conversation timestamp* preference. Click the **Preferences** button, expand **Protocols** and choose **TCP**.



Figure 121.1: Add/Sort a TCP Delta Time Column

Step 1: Open tr-chappellu.pcapng.

Step 2: Expand the **TCP header** in Packet 1. Right click anywhere on the TCP header, select **Protocol Preferences** and ensure that *Calculate conversation timestamps* is enabled.

Step 3: At the end of the TCP header, go to the **[Timestamps]** section, locate and right-click on the **Time since previous frame in this TCP stream** field. Select **Apply as Column**.

Step 4: The newly created column appears to the left of the Info column. Click and drag your new column to the right of the existing **Delta Displayed** column.

Step 5: Update the column name to **TCP Delta** by right-clicking on the column heading and selecting **Edit Column Details**. Click **OK**.

Step 6: Click the new **TCP Delta** column heading twice to sort from high to low. The packets with the largest delays before them in a TCP conversation appear at the top of the list.

tr-c	happellu.pca	ipng				
File	Edit View	<u>Go</u> <u>Capture</u> <u>Analyz</u>	e <u>Statistics</u> Telephon	y <u>T</u> ools Internals <u>H</u>	elp	- (
0 0	9 📕 📕	<u> </u>	2 9, 4 4 4	7 1 1	ର୍ ପ୍ 🗹 🛛 🕷	🖻 褐 💥
Filte	er:			▼ Expr	ession Clear Ap	ply Save
No.	Time	Delta	Delta Displayed	TCP Delta 🔹	Source	Destination
155	24.321	0.713963000	0.713963000	15.757807000	24.6.173.220	198.66
149	23.421	0.000615000	0.000615000	12.832118000	69.59.180.202	24.6.1
146	23.420	0.012845000	0.0128450	12.830206000	69.59.180.202	24.6.1
86	9.004	0.090737000	0.090737 0	5.996430000	24.6.173.220	198.66
706	39.581	P 000116000	0 0001 00	5 808675000	24.6.173.220	184.73
704	39.581	Sort on this r	new column to fin	d 794852000	24.6.173.220	184.73
703	39.581	any significa	ant delays in your	789487000	24.6.173.220	184.73
705	39.581	TCD	nvereations	788723000	24.6.173.220	184.73
702	39.581		inversau0115	788390000	24.6.173.220	184.73
118	14.321	0.044485000	3.729232000	5.547701000	24.6.173.220	184.73
698	39.580	5.223137000	5.537854000	5.546025000	24.6.173.220	184.73
700	39.580	0.000089000	0.000089000	5.540093000	24.6.173.220	184.73
701	39.581	0.000096000	0.000096000	5.538159000	24.6.173.220	184.73
699	39.580	0.000100000	0.000100000	5.537954000	24.6.173.220	184.73
119	14.321	0.000127000	0.000127000	5.300347000	24.6.173.220	198.66

Step 7: Scroll through the packets to determine where delays have incurred in this communication.

Topic 122: Wireshark Lab 30

This topic describes how to add a TCP Delay Button in Wireshark.

Using a "TCP Delay" button (filter expression) will allow us to efficiently locate delays. By pressing button TCP packets that are preceded by a noticeable delay will be displayed.

Step 1: Open tr-chappellu.pcapng.

Step 2: In the display filter area, enter the following filter: tcp.time_delta > 1

Step 3: Click the **Save** button on the display filter toolbar. Enter **TCP Delay** as the label when prompted. Click **OK** to save your new button.

Step 4: Click your new **TCP Delay** button. You will find that 37 packets match the filter. There are numerous TCP FIN packets in the list – we do not care about it.

4	tr-chappellu.pcapng [Wireshark 1.12.3 (v1.12.3-0-gbb3e9a0 from master-1.12)]
<u>File Edit View Go Capture Analyze Statistics Telephony</u>	ols Internals <u>H</u> elp
0 0 🖉 🖩 🔏 📄 🗙 🕾 I Q 👙 🤤 🏹	👱 🗏 📴 Q, Q, Q, 🗹 🐺 🔟 🥵 🔆 💢
olter: tcp.time_delta>1	Expression Clear Apply Save TCP-HS TCP Dealy
e since previous trame in this TCD shown	Destination Length Protocol Stream index Info
004658000 24.6.1/3.220	184./3.250. 66 TCP 8 [TCP Retransmission] 35630→80 [SYN] Seq=0 W
289570000 184.73.250.2	24.6.173.22 66 TCP 8 80+35630 [SYN, ACK] Seq=0 Ack=1 Win=5840 Le
993251000 24.6.173.220	184.73.250. 66 TCP 9 [TCP Retransmission] 35631→80 [SYN] Seq=0 W
995812000 24.6.173.220	184.73.250. 66 TCP 11 [TCP Retransmission] 35633→80 [SYN] Seq=0 W
999584000 24.6.173.220	184.73.250. 66 TCP 12 [TCP Retransmission] 35634-80 [SYN] Seq=0 w
002086000 24.6.173.220	184.73.250. 66 TCP 13 [TCP Retransmission] 35635-80 [SYN] Seq=0 w
003337000 24.6.173.220	184.73.250. 66 TCP 10 [TCP Retransmission] 35632-80 [SYN] Seq=0 w
994007000 24.6.173.220	184.73.250. 66 TCP 17 [TCP Retransmission] 35639-80 [SYN] Seq=0 w
002635000 24.6.173.220	184.73.250. 66 TCP 14 [TCP Retransmission] 35636-80 [SYN] Seq=0 w
003177000 24.6.173.220	184.73.250. 66 TCP 15 [TCP Retransmission] 35637+80 [SYN] Seq=0 w
002364000 24.6.173.220	184.73.250. 66 TCP 16 [TCP Retransmission] 35638→80 [SYN] Seq=0 W
546025000 24.6.173.220	184.73.250. 54 TCP 17 35639→80 [FIN, ACK] Seq=1 Ack=1 Win=65700 L
537954000 24.6.173.220	184.73.250. 54 TCP 16 35638→80 [FIN, ACK] Seg=1 Ack=1 Win=65700 L
540093000 24.6.173.220	184.73.250. 54 TCP 15.35637→80 [FTN. ACK] Seg=1 Ack=1 Win=65700 L

Figure 122.1: Add a "TCP Delay" Button

Step 5: Select Edit | Preferences | Filter Expressions , update TCP Delay filter expression to: tcp.time_delta > 1 && tcp.flags.fin==0 && tcp.flags.reset==0 and then Click OK.



Step 6: Click your TCP Delay button again. 23 packets are displayed because TCP FIN and RST packets have been removed. Let's further remove HTTP GET requests from the TCP Delay button. Add the following string to the end of your filter: **&& !http.request.method=="GET".** The highest TCP Delta delay is under 6 seconds and is a SYN retransmission pkt. There are 12 SYN retransmissions between the client and 184.73.250.227. There is one SYN/ACK as the RTT is 1.28957 seconds.

4	tr-chappellu.pcapno) [Wireshark 1.12.3 (v1.12.3-0-gbb3e	9a0 from master-1.12)]
<u>File E</u> dit <u>V</u> iew <u>G</u> o <u>C</u> apture <u>A</u> nalyze <u>S</u> ta	tistics Telephon <u>y T</u> ools <u>I</u> nternals <u>H</u> elp		
0 0 🛋 🔳 🔬 🖻 🗎 🗶 🤤	् 🗢 🔿 🖥 🕹 🗐 📑 🗨 Q	0, 🖭 🎬 🗹 🍢 🎉 💢	
Filter: tcp.time_delta>1 && tcp.flags.fin==0) && tcp.flags.reset==0 && !http. ♥ Expression	Clear Apply Save TCP Dealy	
No. Time TCP Delta 🔻	Source Destination Le	ngth Protocol Stream index Info	
86 9.004 5.996430000	24.6.173.22(198.66.239	62 TCP 0 [TCF	Retransmission]
7 3.007 3.007726000	24.6.173.22(198.66.239	66 TCP 0 [TCF	Retransmission]
189 31.41(3.004658000	24.6.173.22(184.73.250	66 TCP 8 [TCF	Retransmission]
554 33.69(3.003337000	24.6.173.22(184.73.250	66 TCP 10 [TCF	Retransmission]
637 33.94 3.003177000	24.6.173.22(184.73.250	66 TCP 15 [TCF	Retransmission]
636 33.94(3.002635000	24.6.173.22(184.73.250	66 TCP 14 [TCF	Retransmission]
638 33.943.002364000	24.6.173.22(184.73.250	66 TCP 16 [TCF	Retransmission]
553 33.69(3.002086000	24.6.173.22(184.73.250	66 TCP 13 [TCF	Retransmission]
551 33.69 2.999584000	24.6.173.22(184.73.250	66 TCP 12 [TCF	Retransmission]
550 33.68(2.995812000	24.6.173.22(184.73.250	66 TCP 11 [TCF	Retransmission]
635 33.93(2.994007000	24.6.173.22(184.73.250	66 TCP 17 [TCF	Retransmission]
535 33.67 2.993251000	24.6.173.22(184.73.250	66 ТСР 9 ГТСР	Retransmission

Topic 123: Wireshark Lab 31

This topic discusses how to obtain the round trip time (RTT) using TCP handshake in Wireshark.

A file transfer process will be very slow if RTT is extremely high and everything else is functioning properly. Wireshark allows us to estimate the RTT by looking at the **tcp.time_delta** value. We can capture at client side and look at the **tcp.time_delta** value between the client's TCP SYN packet and the server's TCP SYN/ACK response.

Step 1: Open tr-cnn.pcapng

Step 2: Enter tcp.flags.syn==1 in the display filter area and then click Apply.

Step 3: Click your TCP Delta column heading twice to sort from high to low. We are interested in the delays preceding SYN/ACK packets.

4			tr-cr	n.pcapng [Wiresł	nark 1.12.3 (v1.1	2.3-0-gbb3e9a0 from ma	ster-1.12)]		
<u>F</u> ile <u>E</u> d	lit <u>V</u> iew <u>G</u> o	<u>Capture</u> <u>Analyze</u> <u>Statistic</u>	s Telephon <u>y T</u> ools <u>I</u> nternal	; <u>H</u> elp					
• •	()	E 🗎 X 🔁 Q, 4	è ⇔ � 주 Ł 🗉 E	Q Q Q F	🖉 🖻 😽 💈	% 🔀			
Filter:	tcp.flags.sy	n==1	v	Expression Clear	Apply S	ave TCP Dealy			
No.	Time	TCP Delta	▼ Source De	stination Le	ngth Protocol	Stream index Info			
4730	25.598	3 6.002244000	63.166.98.11	92.168.1.	66 TCP	98 [TCP R	etransmission] 80→56049 [SYN, ACK
4101	L9.967	3.480712000	157.166.226.1	92.168.1.	66 TCP	83 [TCP S	purious Retra	nsmission] 8	0→56015
4594	19.596	0.339694000	63.166.98.11	92.168.1.	66 TCP	98 [TCP R	etransmission] 80→56049 [SYN, ACK
2559	97.214	0.184279000	199.7.71.72 1	92.168.1.	60 TCP	88 80→560	21 [SYN, ACK]	Seq=0 Ack=1	. Win=819(
2568	37.235	0.181178000	199.7.71.72 1	92.168.1.	60 TCP	89 80→560	22 [SYN, ACK]	Seq=0 Ack=1	. Win=819(
94	50.957	0.176830000	46.51.168.4/1	92.168.1.	66 TCP	25 80→559	57 [SYN, ACK]	Seq=0 Ack=1	. Win=146(
1949	95.462	0.113300000	216.38.164.11	92.168.1.	62 TCP	61 80→559	93 [SYN, ACK]	Seq=0 Ack=1	. Win=4923
1420) 4.392	0.111955000	216.38.164.11	92.168.1.	62 TCP	48 80→559	80 [SYN, ACK]	Seq=0 Ack=1	. Win=4923
2061	L 5.655	0.110960000	50.31.185.441	92.168.1.	66 TCP	72 80→560	04 [SYN, ACK]	Seq=0 Ack=1	. Win=146(
2389	96.412	0.103735000	54.243.175.11	92.168.1.	66 TCP	82 80→560	14 [SYN, ACK]	Seq=0 Ack=1	. Win=146(
2454	46.501	0.101873000	54.243.175.11	92.168.1.	60 TCP	84 80→560	16 [SYN, ACK]	Seq=0 Ack=1	. Win=146(

Figure 123.1: Obtain the Round Trip Time (RTT) Using the TCP Handshake

We can see that there are three packets that are marked as Retransmissions. These are because of connection establishment problems. We can also observe at the RTT to various servers.

Topic 124: RTT: Packets 2 and 3 of TCP Handshake

This topic describes how to use Packets 2 & 3 of TCP handshake for obtaining RTT in Wireshark.

Packets 2 and 3 of the TCP handshake can be used to measure the path latency or round trip time (RTT). We need to run a packet capturing tool such as Wireshark on the server host, and look at the **tcp.time_delta** value between the server's TCP SYN/ACK packet and the client's TCP ACK response. The SYN/ACK packet is easy to locate with a display filter, but the ACK (third packet of the handshake) is a bit trickier. The second (the SYN/ACK) packet of the TCP handshake can be detected by applying the filter: **tcp.flags.syn==1 && tcp.flags.ack==1**.

Detection of the third packet of the handshake is difficult. Use the following characteristics:

- a) tcp.seq==1 (Required) TCP Seq. Number 1 (Relative Seq. Number)
- **b) tcp.ack==1** (Required) TCP Ack. Number 1 (Relative Ack. Number)
- c) tcp.len > 0 (Optional) data in the third packet of the handshake
- d) tcp.push==1 (Optional) PUSH bit set

The filter then becomes (tcp.flags.syn==1 && tcp.flags.ack==1) || (tcp.seq==1 && tcp.ack==1).

Topic 125: Wireshark Lab 32

This topic obtains the round trip time (RTT) using Display Filters in Wireshark.

Step 1: Open tr-chappellu.pcapng.

Step 2: Enter the filter **tcp.flags.syn==1** in the display filter area and then click **Apply**. You will find that 58 packets match this filter. The first two packets are sent from the client port 35,621. Packet 3 and Packet 4 are the first two packets of a new TCP connection. The TCP Delta column indicates the time from the TCP SYN from port 35,622 and the SYN/ACK to that same port, RTT is about 17 ms.

									tr-chappe	llu.pcapn	ng (M	/ireshark 1.12	.3 (v1.12	.3-0-gbb	3e9a0 from mas	ter-1.12)]			
Eile	e <u>E</u> dit	View	<u>G</u> o	<u>C</u> apture	<u>A</u> nalyze	<u>Statistics</u>	Telephon	<u>y T</u> ools	Internals	<u>H</u> elp									
0	۲	(۵	B	X 🔁	0, 🖕	۵	7 1		θQ	0	• 🖉 🕅	8 🕺	Ħ					
F	Filter:	tcp.fla	gs.syn:	==1					✓ Exp	pression	Clea	ar Apply	Save	TCP De	aly				
No.		Time		TCP	Delta		Source		Destin	ation	l	ength Protoc	ol Stream	m index	Info				
	1	0.0	00	0.	0000	00000	24.6	. 173	. 22198	3.66.	23	66 TCF	2	C	35621→80) [SYN]	Seq=0	Win=	8192
	7	3.0	07	3.	0077	26000	24.6	.173	. 22198	3.66.	23	66 TCF	P	C	[TCP Re1	ransmi	ssion]	3562	1→80
	13	8.3	00	5.	0000	00000	24.6	.173	. 22198	3.66.	23	66 TCI	7	1	. 35622→8() [SYN]	Seq=0	Win=	8192
	14	8.3	17	0.	0175	99000	198.	66.2	39.24	.6.17	73.:	66 TCF	>	1	. 80→35622	? [SYN,	ACK]	Seq=0	Ack=
	18	8.3	91	0.	0000	00000	24.6	.173	.2269	.59.1	L80	66 TCF	þ	2	35623→80) [SYN]	Seq=0	Win=	8192
	19	8.4	09	0.	0176	23000	69.5	9.18	0.224	.6.17	73.:	66 TCF	Þ	2	80→35623	S [SYN,	ACK]	Seq=0	Ack=
	30	8.5	60	0.	0000	00000	24.6	.173	. 2269	. 59.1	L80	66 TCF	P	3	35625→80) [SYN]	Seq=0	Win=	8192
	32	8.5	75	0.	0155	73000	69.5	9.18	0.224	.6.17	73.:	66 TCF	Þ	3	80→35625	SYN,	ACK]	Seq=0	Ack=
	50	8.6	54	0.	0000	00000	24.6	.173	.2214	1.101	L.1	66 TCF	>	4	35626→80	[SYN]	Seq=0	Win=	8192
	51	8.6	55	0.	0000	00000	24.6	.173	. 2214	1.101	1.1	66 TCF	P	5	35627→80) [SYN]	Seq=0	Win=	8192
	52	8.6	75	0.	0207	36000	141.	101.	12524	.6.17	73.	66 TCF	>	5	80→35627	′[SYN,	ACK]	Seq=0	Ack=
				•												- Farmel			1.1

Figure 125.1: Obtain RTT using Display Filters

This trace file, which was captured at the client can be used to locate the second and third packet of the TCP handshake. These packets can be used to determine RTT when capturing at the server.

Step 3: Enter the following filter in the display filter area: **(tcp.flags.syn==1 && tcp.flags.ack==1) || (tcp.seq==1 && tcp.ack==1).** Click **Apply** and let's examine the results. There are 69 packets match this filter. Several packets from this set are not of interest to us.

Step 4: Let's enhance the filter with the following conditions: **(tcp.flags.syn==1 && tcp.flags.ack==1) || (tcp.seq==1 && tcp.ack==1) && tcp.len==0 && tcp.flags.fin==0.** Click **Apply**.

<u> </u>	tr-chappellu.pcapng [W	ireshark 1.12.3 (v1.12	.3-0-gbb3e9a0 from master-1.12)]	
<u>F</u> ile <u>E</u> dit <u>V</u> iew <u>G</u> o <u>C</u>	Capture <u>A</u> nalyze <u>S</u> tatistics Telephony <u>I</u> ools <u>I</u> nternals <u>H</u> elp			
0 0 🗸 📕 🔬	🖹 🖹 🗶 🕄 । 🔍 🗢 🔿 春 👱 🗐 📴 । 🕀 २. Q. (• 🎬 🕅 🕵 🎉		
Filter: =1) (tcp.seq=	<mark>== 1 &&tcp.ack== 1) && tcp.len== 0 && tcp.flags.fin== 0 v</mark> Expression Clea	r Apply Save	TCP Dealy	
No. Time	TCD Dalla Causa Davisation I	angth Destand Chess	er inden	
148.317	0.017599000198.66.239.24.6.173.	66 TCP	180→35622 [SYN,	ACK] Se <mark>q=0 Ack=1 Wi</mark>
158.318	0.00018600024.6.173.22198.66.23	54 TCP	1 35622→80 [ACK]	Seq=1 Ack=1 Win=657
198.409	0.01/62300069.59.180.224.6.1/3.	66 ICP	200→35623 [SYN,	ACK] Seq=0 Ack=1 Wi
208.409	0.00015900024.6.173.2269.59.180	54 TCP	2 35623→80 [ACK]	Seq=1 Ack=1 Win=662
328.575	0.01557300069.59.180.224.6.173.	66 TCP	380→35625 [SYN,	ACK] Seq=0 Ack=1 Wi
338.575	0.00012600024.6.173.2269.59.180	54 TCP	3 35625→80 [ACK]	Seq=1 Ack=1 Win=662
528.675	0.020736000141.101.12524.6.173.	66 TCP	580→35627 [SYN,	ACK] Seq=0 Ack=1 Wi
538.675	0.020893000141.101.12524.6.173.	66 TCP	480→35626 [SYN,	ACK] Seq=0 Ack=1 Wi
548.676	0.00013700024.6.173.22141.101.1	54 TCP	5 35627→80 [ACK]	Seq=1 Ack=1 Win=657
558.676	0.00017900024.6.173.22141.101.1	54 TCP	4 35626→80 [ACK]	Seq=1 Ack=1 Win=657
708.772	0.094363000184.73.250.24.6.173.	66 TCP	780→35629 [SYN,	ACK] Seq=0 Ack=1 Wi
				and the second second

We now see only the SYN/ACK and ACK packets of the handshakes in the trace file. Using the TCP Delta column, we can determine the time between each of these packets in each of TCP conversations. For future use, we can save the filter expressions as buttons so that high path latency can be identified.

Step 5: Click Clear to remove the filter.

Topic 126: Wireshark Lab 33

In this topic, we discuss how to plot TCP delays in Wireshark.

Wireshark's advanced IO Graph can be used to plot the maximum **tcp.time_delta** value to locate TCP conversation delays in a trace file.

Step 1: Open tr-chappellu.pcapng.

Step 2: Select Statistics | IO Graph.

Step 3: In the Y Axis Unit area, select Advanced...

Step 4: Select the **MAX(*)** Graph 1 Calc option and enter **tcp.time_delta** in the Calc area.

Step 5: Click the **Graph 1** button to graph your results.

		(Wireshark jumps to packet when you cl high point in the	the linked ick on the graph	۸ A	
					0s	20s 40s
-						V Auis
iraphs Graph 1 Cold	or Filter:	Calc: MAX(*)	▼ tcp.time_delta	Style: Line	▼ 🖉 Smooth	X Axis Tick interval: 1 sec
iraphs Graph 1 Colo Graph 2 Colo	or Filter:	Calc: MAX(*) Calc: SUM(*)	tcp.time_delta	Style: Line Style: Line	▼ V Smooth	X Axis Tick interval: 1 sec Pixels per tick: 5
Graphs Graph 1 Cold Graph 2 Cold Graph 3 Cold	r Filter.	Calc: MAX(") Calc: SUM(") Calc: SUM(")	tcp.time_delta	Style: Line Style: Line Style: Line	 Smooth Smooth Smooth Smooth 	X Axis Tick interval: 1 sec Pixels per tick: 5 View as time of day
iraphs Graph 1 Colo Graph 2 Colo Graph 3 Colo Graph 3 Colo Graph 4 Colo	or Filter:	Calc: MAX(") Calc: SUM(") Calc: SUM(") Calc: SUM(")	tcp.time_delta t	Style: Line Style: Line Style: Line Style: Line	V Smooth V Smooth V Smooth V Smooth V Smooth V Smooth	X Axis Tick interval: 1 sec Pixels per tick: 5 View as time of day Y Axis Unit: Advanced

Figure 126.1: Graph TCP Delays

From the graph of this trace file traffic, we can observe that there is a spike in the RTT values around 25 seconds. As you click on this high point in the graph, Wireshark will jump to that packet in the main window. Wireshark takes us to packet number 155. The TCP Delta column value is 15.757807 seconds. This is a TCP FIN. For such a delay, we don't care much. Let's add a filter to remove such normal delays.

Step 6: Enter the following expression in the Filter area of Graph1: tcp.time_delta >
1 && tcp.flags.fin==0 && tcp.flags.reset==0 && !http.request.method=="GET".
Click the Graph 1 button again to so that the filter can be applied.

			Rem accepta makes th	oving "normal" able delays fror is a more usefu	' or n view ıl graph				0	A	1 1 20s	M	
raphs	_								_	3	X Axis		_
Graph 1	Color	Filter:	tcp.time_delta > 1 && tcp.fla	Calc MAX(*)		tcp.time_delta	Style	Line	-	Smooth	Tick inter	val: 1 sec	
	Color	Filter:		Calc: SUM(*)			Style:	Line		Smooth	Pixels per	tick: 5	
oraph 2	A.L.	Filter:		Calc: SUM(*)			Style	Line	•	Smooth	V Avia	as time of day	1
Graph 2 Graph 3	CONDE			Calc: SUM(*)			Style:	Line	•	Smooth	Unit	Advanced	
Graph 2 Graph 3 Graph 4	Color	Filter:			Concession of the local division of the loca	(0.0			1337		- 1
Graph 2 Graph 3 Graph 4 Graph 5	Color	Filter:		Calc: SUM(*)	-		Style:	Line		Smooth	Scales	Auto	

Now when you click on the largest delay point in this graph, Wireshark jumps to Packet 86.

4		tr-chappellupcapng [Wireshark 1.12.3	(v1.12.3-0-gbb3e9a0 from master-	1.12)]	- 0
<u>File Edit View Go</u>	Gapture Analyze Statistics Telephony Looks	Internals Help			
0 0 1 1 1	🗎 🗎 🗶 🖉 🔍 🗧 🗢 🐳 🕇 🛓	। 📃 📑 । Q. Q. Q. 🖭 । 📓 🖬 🍕	\$ K 😫		
Filtes		✓ Expression Clear Apply	Seve TCP Dealy		
No. Time	TCP Delta Source	Destination Length Protocol	Stream index Info		
808.830	24.6.17	3.275.75.75.79 DNS	Standard	query 0xfd77 AAAA ww	w.nitrorea
818.838	75.75.7	5.724.6.173.102 DNS	Standard	query response 0xe4dd	aaaa 201
82 8.860	24.6.17	3.275.75.75.79 DNS	Standard	query Oxfd77 AAAA ww	w.nitrorea
838.869	75.75.7	5.724.6.173.400 dns	Standard	query response 0xfd77	Server fa
848.873	75.75.7	5.724.6.173.400 dns	Standard	query response 0xfd77	Server fa
85 8.913	0.00506200(184.73.	25024.6.173. 60 TCP	7 80 /35629	[ACK] 3eg-1 Ack-601 W	n-7168 L
869.004	5.99643000/24.6.17	3.2198.66.2: 62 TCP	0 [TCP Retr	ansmission] 35621+80	[SYN] Seq:
879.020	0.01670400(198.66.	23924.6.173. 62 TCP	0 80 -35621	[SYN, ACK] Seg=0 Ack=	1 Win-655.
889.021	0.00018000(24.6.17	3.2198.66.2: 54 TCP	0 35621→80	[ACK] Seq=1 Ack=1 Win	=64240 Lei
899.153	0.24018900(184.73.	25024.6.173.885 HTT	р 7 нттр/1.1	200 OK (application/	json)
					1

We can see that this is a TCP SYN retransmission packet. It took almost 6 seconds after the previous packet. Thus, indicating a connection request that is not receiving responses.

Topic 127: Wireshark Lab 34

This topic how to find DNS response times in Wireshark.

Domain Name Service (DNS) provides the network IP address of a machine for a network name such as (such as <u>www.wireshark.org</u>). Applications protocols such as SMTP, HTTP generate DNS queries when a user provides a network name to them. DNS, which is a request/response protocol, employs UDP.

Step 1: Open tr-dns-slow.pcapng.

Step 2: The **dns.time** field exists only in DNS response packets. In this trace file, Packet 3 is the first DNS response packet. Let's use this packet to create a **dns.time** column. Expand Packet 3's the **Domain Name System (response)** section.

Step 3: Right-click on the [Time: 0.107083000 seconds] line and click Apply as Column.



Step 4: The newly created column appears to the left of the Info column. You can click and drag your new column to the right of the existing **TCP Delta** column.

Step 5: Right-click on the new column heading and selecting **Edit Column Details**. Rename the new column to **DNS Delta**. Click **OK**.

Step 6: By clicking the new **DNS Delta** column heading twice, it gets sorted from high to low.

🖌 tr-	dns-sl	ow.pca	png	1													,
<u>F</u> ile	<u>E</u> dit	View	<u>G</u> o	Captur	e <u>A</u> na	lyze	<u>Statistics</u>	Telep	hony	Tools	Internals	Help					
0	•	(=	đ		X	2	9.0	\$	S	F 2		0	a a) 🔧 %	
Filt	ter:										• E	xpression	Cle	ear	Apply	Sav	e TCP De
No.	Time		TCP C	elta		DN	S Delta 👡		▼ Sou	rce		Des	tination			Protocol	Info
11	1.7	292				1.	29219	20	21	6.14	8.227.	68 24	.6.12	26.21	.8	DNS	Standa
107	2.	329				0.	20725	000	~	-						DNS	Standa
3	1.1	L07				0.	10708	3000	0 20	So	rt vour d	ins.ti	ime CO	lumn	to	DNS	Standa
200	3.3	381	0.0	0009	8000				24		lantify al		C room			ТСР	[TCP
199	3.3	381	0.0	0008	6000				24		ientity si	OWDN	Stesp	onse	5	ТСР	aairne
198	3.3	381	0.0	0130	6000				14	0.02	. 210.1	JU 24	.0.14	U. LI	.0	HTTP	ГТСР
197	3	880	0_0	0005	5000	ia ĝ	-	ala	24	.6.1	25,218	14	6.82.	218	136	ТСР	[ТСР

A significant delay preceding Packet 11 of this trace file can be observed —almost 1.3 seconds. We need to investigate further to determine why the DNS server is working slowly.

Topic 128: Wireshark Lab 35

This topic describes how to create a Button to detect High DNS response times in Wireshark.

Domain Name System is a distributed database implemented in a hierarchy of many DNS servers. It is an application-layer protocol that allows hosts to query the distributed database and to resolve names (address/name translation). Consider a browser (i.e. an HTTP client) running on some a host A, who requests www.wireshark.org. In order for host A to be able to send an HTTP request message, host A must first obtain IP address of server. The host A runs the client side of the DNS application. The browser extracts the hostname from the URL and passes it to the client side of the DNS application. The DNS client sends a query using UDP to the DNS server listening at port 53. The DNS client eventually receives a reply which includes the IP address for the host name. Once the browser receives the IP address from DNS, it can initiate a TCP connection to the HTTP server process located at port 80 at that IP address.

Let's create a button to detect DNS response times larger than 1 second.

Step 1: Open tr-dns-slow.pcapng.

Step 2: Type **dns.time > 1** in the display filter area and then click **Save**.

Step 3: Name your button DNS Delay and click OK.

Step 4: Click your new **DNS Delay** button. We observe that Packet 11 is the only packet that matched this filter i.e Packet 11 contains the largest DNS delay time.

												tr-dns-s	low.pc	png	[Wire	shark	.12.3	(v1.12	2.3-0	gbb30	:9a0 fr	om ma	ster-1.	12)]			
<u>F</u> ile	<u>E</u> dit	View	<u>G</u> o	<u>C</u> aptu	ire <u>A</u>	<u>i</u> nalyze	<u>S</u> tati	tistics	Teleph	nony]	Tools	Internals	<u>H</u> elp														
0	0	(1	ß	6		XÊ	0	(¢	\$	97	ł		⊕	Q (Q 🖻	2	8	*	Ħ								
4	iter:	dns.tin	ne > 1							>		۷	Expressio	n	Clear	Ap	ily	Save	Т	P Dealy							
No.		Time		S	ource				Dest	tination			Lengt	h Pr	rotocol	Info											
	11	1.29	92		216	.14	8.2	227.	624	4.6.	126	.218	- 4	99 C	NS	Sta	inda	ard	que	ery	resp	ons	e Ox	002	9 CN	AME	us.
<																											
• F	rar	ne 1	1:	499	b	yte	S 0	n w	ire	(39	992	bits	5), 4	199	byt	es (apt	ure	d (399	2 bi	ts)	on	int	erfac	e 0	
•	the	erne	t I	Ί,	Sr	c: (Cad	ant	_22	:89	:c2	(00:	01:	ic:	22:8	9:c	2),	Dst	: A	mbi	tMid	_aa	:af:	80	(00:d	0:5	9:aa
•	Inte	erne	t F	rot	toc	۱ 10	Ver	sio	n 4	, Sr	rc:	216.	148	22	7.68	(2	16.1	.48	227	.68),[st:	24.	6.1	26.21	8 (24.6
٥	lsei	' Da	tag	Iran	ηP	rot	000	1,	Src	Por	rt:	53 ((53)	D	st P	ort	36	517	(36	17)							
	So	urce	P	ort	: 5	3 (53))																			
	De	stir	at	ion	Ро	rt:	36	517	(36	17)																	
	Le	ngth	1: 4	465																							
8	Ch	ecks	um	: 0	xb2	35	[va	lid	lati	on	dis	able	d]														
	[S	trea	um :	ind	ex:	0]																					
•)oma	ain	Nan	ie s	sys	tem	(r	esp	ons	e)																	

Figure 128.1: Create a Button to Detect High DNS Response Times

Topic 129: Wireshark Lab 36

In this topic, we describe how to plot DNS response times in Wireshark.

To highlight DNS delays in a trace file, let's create a graph.

- Step 1: Open tr-dns-slow.pcapng.
- Step 2: Select Statistics | IO Graph.
- Step 3: In the Y Axis Unit area, select Advanced...

Step 4: Select the **MAX(*)** Graph 1 Calc option and enter **dns.time** in the Calc area.

Step 5: Click the Graph 1 button to plot results.

		This c respo	graph shows that one point where th onse time exceeds	t we have We DNS Is 1 second	
(н			0s
araphs				_	A AUS
Graph 1 Color Filter:	Calc: MAX(*)	▼ dns.time	Style: Line	🔻 📝 Smooth	Tick interval: 1 sec
Graph 1 Color Filter. Graph 2 Color Filter.	Calc: MAX(*) Calc: SUM(*)	• dns.time	Style: Line Style: Line	 Smooth Smooth 	Tick interval: 1 sec Pixels per tick: 5
Graph 1 Color Filter: Graph 2 Color Filter: Graph 3 Color Filter:	Calc: MAX(") Calc: SUM(") Calc: SUM(")	• dns.time	Style: Line Style: Line Style: Line	 ✓ Smooth ✓ Smooth ✓ Smooth ✓ Smooth 	Tick interval: 1 sec Pixels per tick: 5 View as time of day
Graph 1 Color Filter. Graph 2 Color Filter. Graph 3 Color Filter. Graph 4 Color Filter.	Calc: SUM(*) Calc: SUM(*) Calc: SUM(*) Calc: SUM(*) Calc: SUM(*)	dns.time d	Style: Line Style: Line Style: Line Style: Line	 Smooth Smooth Smooth Smooth Smooth 	Tick interval: 1 sec Pixels per tick: 5 View as time of day Y Axis Unit: Advanced

Figure 129.1: Graph DNS Response Times

Step 6: Click the highest point in the graph. Wireshark takes us to that packet.

1		tr-dns-slow.pcapng [Wireshark 1.12.3 (v1.12.3-0-gbb3e9a0 from master-1.12)]
Eik	e <u>E</u> dit <u>V</u> iew <u>G</u> o	<u>Capture Analyze Statistics Telephony</u> Iools Internals <u>H</u> elp
0	0 🖌 🔳 🔬	= ¥ 2
8	filter:	Expression Clear Apply Save TCP Dealy
No.	Time	Source Destination Length Protocol Info
	10.000	24.6.126.218 216.148.227.6 73 DNS Standard query 0x0029 A www.ncmec.org
	21 000	24.6.126.218.204.127.202.4 73.DNS Standard query 0x0020 A www.nemec.org
	31.107	204.127.202.424.6.126.218 499 DNS Standard query response 0x0029 CNAME us mi
	41 111	24 6 126 218 146 82 218 13 62 TCP 3618-80 [SVN] Seg-0 Win-64512 Lon-0 M99-146
	51.189	146.82.218.1324.6.126.218 62 TCP 80→3618 [SYN, ACK] Seq=0 Ack=1 Win=5840 Len
	61.189	24.6.126.218 146.82.218.13 54 TCP 3618→80 [ACK] Seq=1 Ack=1 Win=64512 Len=0
	71.191	24.6.126.218 146.82.218.13 459 HTTP GET / HTTP/1.1
	81.275	146.82.218.1324.6.126.218 60 TCP 80→3618 [ACK] Seq=1 Ack=406 Win=6432 Len=0
	91.291	146.82.218.1324.6.126.218 1514 HTTP HTTP/1.1 200 OK (text/html)
	101.292	24.6.126.218 146.82.218.13 54 TCP 3618-80 [ACK] Seq=406 Ack=1461 Win=64512 Le

We can see it is packet 3. It is a DNS query response packet has caused the largest delay. Sorting **dns.time** column can help us to detect delays more quickly.

Topic 130: Wireshark Lab 37

This topic discusses how to reassemble TCP streams by disabling the Allow Subdissector in Wireshark.

The delta time between an HTTP request (e.g. GET) and the response (e.g. 200 OK) is the HTTP response time. In Wireshark, HTTP response time can be estimated with **http.time** field. If a web server is overloaded with the connection, or requests, or needs to consult another server to answer client's requests. This can result in high HTTP response times. If enabled, Wireshark measures time from HTTP Request to the final data packet of the response. If you want to estimate the total time taken by downloading an object, then enable this setting. By disabling it, we can learn how quickly the server responded. There are two methods to change this TCP preference setting.

Step 1: Open tr-http-pcaprnet101.pcapng.

Step 2: Right-click on the **TCP header** in the Packet Details of Packet 5. Select **Protocol Preferences.** Then, uncheck *Allow subdissector to reassemble TCP streams*.



Figure 130.1: Disable the *Allow Subdissector to Reassemble TCP Streams* Preference Setting

Step 3: The 2nd method is: Select **Edit | Preferences | (+) Protocols | TCP**.



Topic 131: Wireshark Lab 38

This topic describes how to locate HTTP response times in Wireshark.

Step 1: Open tr-http-pcaprnet101.pcapng.

Step 2: Only the HTTP response packets contain the **http.time** field. In this trace file, the first HTTP response packet is Packet 10 (HTTP 303 See Other).

4	1								tr-h	ttp-pcap	rnet101.	pcap	ng [Wireshark	1.12.3	v1.12.3-0-	gbb3e9a0 fro	om master-1	.12)]			
File	Edit	t <u>V</u> iew	Go	Capture	Analyze	Statis	stics T	elephony	Tools	Internals	Help											
0	0		đ		X	9 Q		4 📦	72		0,6	0	. 🗉		5 %							
F	ilte:									¥ 8	pression.		Clear	Apply	Save	TCP De	sly					
No,		Time		Source				Destinal	ion		Length	Pro	tocol	Info								
	1	0.0	00	24.	6.1	73.	220	75.	75.7	5.75	7	3 DI	NS	Stand	lard	query	Oxc3bf	A www	.pcap	r.ne	t	
	2	0.0	21	75.	75.	75.	75	24.	5.17	3.220	8	9 DI	NS	Stand	dard	query	respon	se Oxc3	bf A	209	.133.3	32.6
	3	0.0	23	24.	6.1	73.	220	75.	75.7	5.75	7	3 DI	NS	Stand	dard	query	0x406e	AAAA	www.p	capr	.net	
	4	0.0	48	75.	75.	75.	75	24.1	5.17	3.220	140	6 DI	NS	Stand	lard	query	respon	se 0x40)6e			
	5	0.0	51	24.	6.1	73.	220	209	.133	. 32.6	9 60	6 T	CP	21213	3→80	[SYN]	Seq=0	Win=819	2 Len	=0 N	ISS=140	50 W
	6	0.0	70	209	.13	3.3	2.69	24.	5.17	3.220	66	6 T	СР	80→21	213	[SYN,	ACK] S	eq=0 Ac	:k=1 W	in=5	840 Le	en=0
	7	0.0	70	24.	6.1	73.	220	209	.133	. 32.6	9 54	4 T	CP	21213	3→80	[ACK]	Seq=1	Ack=1 V	in=65	700	Len=0	
	8	0.0	71	24.	6.1	73.	220	209	.133	. 32.6	34:	1 H	TTP	GET /	/ HTT	P/1.1						
	0	0.0	88	200	.13	3.3	2.65	24.	.17	3.220	61	O TI	CP	80 -21	213	[ACK]	Seg-1	Ack=288	Win=	6912	Len=	0
ſ	10	0.0	97	209	0.13	3.3	2.69	24.	5.17	3.220	35	7 H	TTP	HTTP/	1.1	303 S	ee Othe	r				
Т	11	0.0	98	209	.13	3.3	2.69	24.	5.17	3.220	60	0 T (СР	80→21	213	[FIN.	ACK] S	eg=804	Ack=2	88 h	in=691	12 L
5	12	0.0	00	24	6.1	72	220	200	122	22.6	2	1 1	CD	21212	0.9	EACK1	Seg-20	Ack=3	05 Wi	n=65	396 Le	en=0
	13	0.0	99	24.	6.1	73.	220	209	133	. 32.6	5	4 T(CP	21213	3→80	[FIN.	ACK1 S	eq=288	Ack=3	05 W	in=653	396
	14	0.1	05	24.	6.1	73.	220	209	133	. 32.6	9 66	6 T	CP	21214	++80	[SYN]	Seg=0	Win=819	2 Len	=0 N	ISS=146	50 W

Figure 131.1: Add/Sort an HTTP Response Time Column to Find HTTP Response Times

Let's use this packet to create our **http.time** column. In the Packet Details pane of Packet 10, right-click on the **Hypertext Transfer Protocol** section and select **Expand Subtrees.**

Step 3: Locate the [Time since request: 0.026416000 seconds] line and right-click on it and select Apply as Column.

Step 4: Your newly created column appears to the left of the Info column. Click and drag your new column to a place with better visibility.

Step 5: Right-click on the column heading and select **Edit Column Details**. Rename this new column to **HTTP Delta**. Click **OK**.

Step 6: Sort HTTP responses from high to low by clicking your new **HTTP Delta** column heading twice. Packets with the largest delays will appear at the top of the list.

<u>File Edit View Go</u>	apture <u>A</u> nalyze <u>S</u> tatistics Telephony <u>T</u> ools <u>I</u> nternals <u>H</u> elp	
0 0 💉 🔳 🔬	🖹 🕻 🗶 🔍 🗢 🗢 🖓 🛃 🗐 📴 🔍 Q. Q. 🖭) 🕷 🛙	1 🐻 🔆 🛄
Filter:	▼ Expression Clear Apply	Save TCP Dealy
No. Time	HTTP Delta Source Destination Length	Protocol Info
432 20.508	1.92250400209.133.32.6924.6.173.220 151	4 HTTP HTTP/1.1 200 OK (text/html)
201.924	1.79835200209.133.32.6924.6.173.220 151	4 HTTP HTTP/1.1 200 OK (text/html)
307 2.341	0.13315100173.194.79.8224.6.173.220 148	4 HTTP HTTP/1.1 200 OK (application/
427 19.186	0.12387400209.133.32.6924.6.173.220 117	3 HTTP HTTP/1.1 200 OK (text/html)
257 2.207	0.08842200173.194.79.8224.6.173.220 117	1 HTTP HTTP/1.1 200 OK (PNG)
259 2.208	0.08420000173.194.79.8224.6.173.220 148	4 HTTP HTTP/1.1 200 OK (PNG)[Unreass
412 13.291	0.07508700209.133.32.6924.6.173.220 117	3 HTTP HTTP/1.1 200 OK (text/html)
483 22.880	0.07350200209.133.32.6924.6.173.220 117	3 HTTP HTTP/1.1 200 OK (text/html)
152 2.107	0.07115400173.194.79.8224.6.173.220 148	4 HTTP HTTP/1.1 200 OK (text/plain)
161 2.109	0.06962200173.194.79.8224.6.173.220 148	4 HTTP HTTP/1.1 200 OK (text/plain)
1652.110	0.06942600173.194.79.8224.6.173.220 75	OHTTP HTTP/1.1 200 OK (text/css)
266 2.215	0.06040200173.194.79.8224.6.173.220 148	4 HTTP HTTP/1.1 200 OK (PNG)[Unreass
233 2.172	0.05996200173.194.79.8224.6.173.220 52	4 HTTP HTTP/1.1 200 OK (PNG)
228 2.171	0.05972900173.194.79.8224.6.173.220 148	4 HTTP HTTP/1.1 200 OK (application/

We can observe that HTTP response time from the HTTP server (209.133.32.69) is over 1.7 seconds twice in this trace file. Those packets are bearing numbers 432 and 20.

Topic 132: Wireshark Lab 39

In this topic, we create a button to quickly detect high HTTP response times in Wireshark.

Wireshark allows us to create buttons based on filter expressions. It is one of the most powerful features to improve your efficiency in locating network problems.

Step 1: Open tr-http-pcaprnet101.pcapng.

Step 2: Type **http.time > 1** in the display filter area and then, click **Save**.

							tr-http	-pcaprnet	:101.pcaj	png (V	Vireshark	1.12.3 (v1	l.12.3-0-gbb3e9a) from m	aster-1.1	12)]	
<u>F</u> ile	<u>E</u> dit	<u>V</u> iew <u>G</u> o	<u>Capture</u>	<u>A</u> nalyze	<u>Statistics</u>	Telephony	<u>T</u> ools <u>I</u> r	iternals <u>H</u> e	lp								
0	0		6	X 🕄	0, ¢	🔶 🥥 🖥	F 🕹		Ð, Q, (Q 🖭	¥ ¥	5 🕺	1				
E	ilter:	http.time >	>1					✓ Expres	ssion	Clear	Apply	Save	TCP Dealy				
Vo.		Time	HTTP	Delta	Sour	rce		Destinatio	n		Length	Protocol	Info				
	20	1.924	1.	79835	520(20	09.133	3.32.	624.6	.173	. 220)1514	HTTP	HTTP/1.1	200	OK	(text/html)	
4	132	20.50	8 1.	92250	040(20	09.133	3.32.	624.6	.173	. 220)1514	HTTP	HTTP/1.1	200	0K	(text/html)	

Step 3: When you click **save**, Wireshark prompts to name the button. Name it **HTTP Delay** and then click **OK**.



Always remember to clear your display filter when you are done reviewing the results.

Step 4: As you click your new **HTTP Delay** button, you will find two packets of this trace file matching the filter—Packet 20 and Packet 432.

Topic 133: Wireshark Lab 40

In this topic, we plot HTTP response times in Wireshark.

Wireshark allows a user to define up to five differently colored graphs. We can make the following configurations:

Graphs

- **Graph 1-5:** enable the specific graph 1-5 (only graph 1 is enabled by default).
- Color: (cannot be changed)
- Filter: a display filter for this graph
- **Style:**(Line/Impulse/FBar/Dot)

X Axis

- Tick interval: an interval in x direction lasts (10/1 mins or 10/1/0.1/0.01/0.001s)
- **Pixels per tick:** use 10/5/2/1 pixels per tick interval
- View as time of day: instead of sec or mins, view x as time of day

Y Axis

- Unit: the unit for y direction (Packets/Tick, Bytes/Tick, Bits/Tick, Advanced...)
- Scale: the scale for the y unit (Logarithmic, Auto, 10, 20, 50, 100, 200, 500, ...)

Let's Graph HTTP Response Times.

Step 1: Open tr-http-pcaprnet101.pcapng.

	[tr-	http	-pcaprn	et101.p	capng [\	Wireshark	1.12.3 (v	1.12.3-0-gb	b3e9a0 fro	m mas	ster-1.
F	ile <u>E</u> di	t <u>V</u> iew	Go	$\underline{C} apture$	Analyze	Statis	tics T	clephon	lools 1	Int	ernals H	Help							
Ę	0 0	/ =	ß	8	X 🔁	10	\$	۵	Ŧ 1			⊕, ⊝,	0, 🖭	¥ ¥	🍕 🐇 ।	11			
	Filten										✓ Exp	ression	Clear	Apply	Save	TCP Dealy	HTTP Delay		
ł	o.	Time		Sour	ce			Destina	tion			Length	Protocol	Info					
	1	.0.0	00	24	.6.1	L73.	220)75.	75.7	75.	75	73	DNS	Stan	dard	query	0xc3b	of	A w
	2	0.0	21	75	.75.	75.	75	24.	6.17	73.	220	89	DNS	Stan	dard	query	respo	nse	0x
	3	0.0	23	24	.6.1	173.	220)75.	75.7	75.	75	73	DNS	Stan	dard	query	0x406	ie	AAA.
	4	0.0	48	75	.75.	75.	75	24.	6.17	73.	220	146	DNS	Stan	dard	query	respo	nse	0x-
	5	0.0	51	24	.6.1	.73.	220	209	.133	3.3	32.6	66	TCP	2121	3→80	[SYN]	Seq=0) Wi	n=8
	6	0.0	70	20	9.13	33.3	2.6	24.	6.17	73.	220	66	TCP	80→2	1213	[SYN,	ACK]	Seq	=0.
	7	0.0	70	24	.6.1	.73.	220	209	.133	3.3	32.6	54	TCP	2121	3→80	[ACK]	Seq=1	L AC	k=1
	8	0.0	71	24	.6.1	.73.	220	209	.133	3.3	32.6	341	HTTP	GET	/ нтт	FP/1.1			
	9	0.0	88	20	9.13	33.3	2.6	24.	6.17	73.	220	60	TCP	80→2	1213	[ACK]	Seq=1	L AC	k=2
	10	0.0	97	20	9.13	33.3	2.6	24.	6.17	73.	220	357	HTTP	НТТР	/1.1	303 Se	ee oth	ier	
	11	0.0	98	20	9.13	33.3	2.6	24.	6.17	73.	220	60	TCP	80→2	1213	[FIN,	ACK]	Seq	=30
	12	0.0	99	24	.6.1	173.	220	209	.133	3.3	32.6	54	TCP	2121	3→80	[ACK]	Seq=2	88	Ack
	13	0.0	99	24	.6.1	.73.	220	209	.133	3.3	32.6	54	TCP	2121	3→80	[FIN,	ACK]	Seq	=28
	14	0.1	05	24	.6.1	173.	220	209	.133	3.3	32.6	66	TCP	2121	4→80	[SYN]	Seq=0) Wi	n=8

Step 2: Select Statistics | IO Graph.

Step 3: In the Y Axis Unit area, select Advanced...

Step 4: Select the **MAX(*)** Graph 1 Calc option and then, enter **http.time** in the Calc area.

Step 5: To plot your results, click the Graph 1 button.

			The http.time g high-response t	raph makes the imes easy to spo	two ot	
						0s 20s
			m			
raphs Graph 1 Color	Filter:	Calc: MAX(*)	Thttp.time	Style: Line	💌 👽 Smooth	X Axis Tick interval: 1 sec
raphs Graph 1 Color Graph 2 Color	Filter:	Calc: MAX(*) Calc: SUM(*)	m http.time	Style: Line Style: Line	Smooth Smooth Smooth	X Axis Tick interval: 1 sec Pixels per tick: 5
raphs Graph 1 Color Graph 2 Color Graph 3 Color	Filter: Filter: Filter:	Calc: MAX(*) Calc: SUM(*) Calc: SUM(*)	" http.time	Style: Line Style: Line Style: Line	Smooth Smooth Smooth Smooth Smooth	X Axis Tick interval: 1 sec Pixels per tick: 5 View as time of day
raphs Graph 1 Color Graph 2 Color Graph 3 Color Graph 4 Color	Fjiter; Fjiter; Fjiter; Fjiter;	Calc: MAX(*) Calc: SUM(*) Calc: SUM(*) Calc: SUM(*)	III	Style: Line Style: Line Style: Line Style: Line	V Smooth V Smooth V Smooth V Smooth V Smooth V Smooth V Smooth	X Axis Tick interval: 1 sec Pixels per tick: 5 View as time of day Y Axis Unit: Advanced

If you click the highest points in the graph, Wireshark will take you to those packets which have encountered highest delays. Packet 20 and Packet 432 show that the HTTP response time from the HTTP server (209.133.32.69) was over 1.7 seconds

Topic 134: Wireshark Lab 41

In this topic, we add Server Message Block (SMB) response time column in Wireshark.

What is a Server Message Block (SMB)?

On Windows-based networks, Microsoft employs a file sharing protocol called Server Message Block (SMB). SMB and SMB version 2 are the two versions that are commonly employed on Windows-based networks. SMB is a request/response protocol. Its response codes are available on Microsoft's Open Specification site <u>www.microsoft.com/openspecifications/</u>. SMB delays can be estimated by looking at the **smb.time** field in the SMB response packets.

Step 1: Open tr-smb-slow.pcapng.

Step 2: The first SMB response packet is Packet 5 (Negotiate Protocol Response).

	4										tr-smi	b-slow	.pcapr	ng [Win	eshark 1.12	2.3 (v1.1	2.3-0-gbb	3e9a0 fron	n master-1	1.12)]				
	File	Edit 1	liew (<u>io C</u> ap	ture /	<u>I</u> nalyze	Stat	istics	Telephor	ny <u>I</u> ools	Intern	als <u>H</u> e	elp											
	0	0 🖌		116		X 🔒	0	\ #	🦃	7 Ż			ର୍ପ୍	0 🖻	i	5 %	I							
	Filt	er:									Y	Expre	ssion	Clear	Apply	Save	TCP Dea	aly HTTP De	lay					
1	lo.	Tir	ne		Source				Destin	ation		le	ngth	Protocol	Info									
		10	.00	0	192	2.16	58.	0.8	8 192	2.168	8.0.	99	62	TCP	1178	→445	[SYN] Seq	=O wir	1=655	35 Le	n=0	MSS=	4
		20	.00	0	192	.16	58.	0.9	9 1 97	2.168	3.0.	88	62	TCP	445→	1178	[SYN	, ACK] Seq=	=0 Ac	k=1 V	/in=8	192 I	.e
		30	. 00	1	192	16	8	08	8 1 9 1	2 168	0	99	54	тср	1178	<u>→445</u>	Гаск] Seg	=1 Ac	<u>(=1</u>)	/in=65	535	Len=()
	٢	40	.00	2	192	2.16	68.	0.8	8 192	2.168	8.0.	99	191	SMB	Nego	tiat	e Pro	tocol	Reque	est				
	L	50	.00	3	192	2.16	58.	0.9	9 192	2.168	8.0.	88	251	SMB	Nego	tiat	e Pro	tocol	Respo	onse				
		60	.00	5	192	. 10	ð.	υ.δ	8 I.A'	2.160	.0.	99	359	KKB3	A2-K	EQ			_					
		70	.00	4	192	1.16	58.	0.9	9 1 92	2.168	8.0.	88	148	KRB5	KRB	Erro	r: KR	B5KDC	_ERR_C	C_PRI	NCIPA	L_UN	KNOW	
		80	.00	5	192	2.16	68.	0.8	8 192	2.168	8.0.	99	359	KRB5	AS-R	EQ								
		90	.00	5	192	2.16	68.	0.9	9 192	2.168	8.0.	88	148	KRB5	KRB	Erro	r: KR	B 5KDC	_ERR_C	C_PRI	NCIPA	L_UN	KNOW	1
		10 0	.00	6	192	2.16	58.	0.8	8 192	2.168	8.0.	99	294	SMB	Sess	ion	Setup	AndX	Reque	est,	NTLMS	SP_N	EGOT	[]
		110	.00	7	192	2.16	58.	0.9	9 1 9	2.168	8.0.	88	574	SMB	Sess	ion	Setup	AndX	Respo	onse,	NTLM	ISSP_	CHALI	E
		12 0	.00	8	192	2.16	58.	0.8	8 192	2.168	8.0.	99	434	SMB	Sess	ion	Setup	AndX	Reque	est,	NTLMS	SP_A	UTH,	u
		13 0	.00	8	192	2.16	68.	0.9	9 192	2.168	3.0.	88	93	SMB	Sess	ion	Setup	AndX	Respo	onse,	Erro	r: S	TATUS	5_
		14.0	00	^	100	1 10	0	0 0	0 1 0	1.00	0	00	E 4	-	1170	445	Correct.	1.01/	1	750	A also 7		in C	1

The **smb.time** field exists only in SMB response packets. Let's use packet 5 to create a **smb.time** column. Expand the **SMB** section and the **SMB Header** section of Packet 5 in the Packet Details pane.

Step 3: Right-click on the [Time from request: 0.000766000 seconds] line and click Apply as Column.

trsmb File Edit Yew So Supture Analyze Statistics Telephony Icols Internal ◎ ◎ ▲ ■ ▲ ⊨ □ ¥ & □ • ◆ ◆ ● 7 ± □	Expand Subtrees Collapse Subtrees Expand All Collapse All)-gbb3e9a0 from master-1.12)]
Filter	Apply as Column	CP Dealy HTTP Delay
No. Time Source Destination 10.000 192.168.0.88 192.168.0.5 2 20.000 192.168.0.88 192.168.0.5 3 30.001 192.168.0.88 192.168.0.5 4 40.002 192.168.0.88 192.168.0.5 5 60.003 192.168.0.99 192.168.0.5 5 70.004 192.168.0.99 192.168.0.5 5 90.005 192.168.0.99 192.168.0.5 9 90.005 192.168.0.88 192.168.0.5 9 SMB (Server Message Block Protocol) SMB Header Server Component: SMB Festorise to: 41 10 10 10	Apply in Hele Proper a Filter Colorise with Filter Follow ICP Stream Follow ICP Stream Copy Expond Stelected Packet Bytes Eith Protect @ Wile Protocol Page @ Filter Field Reference Protocol Help Protocol Peter Decode Ac @ Daskle Protocol Brosher Name @ to Companyonal Packet	YN] Seq=0 Win=65535 Le YN, ACK] Seq=0 Ack=1 W ACK] Seq=1 Ack=1 Win=65 Protocol Request Protocol Response KRB5KDC_ERR_C_PRINCIPA KRB5KDC_ERR_C_PRINCIPA
[Time from request: 0.000766000	Show Packet Reference in New Window	
NT Status: STATUS_SUCCESS (0x000	00000)	

4									tr-smb	slow.	pcapn	g [Wire	eshark 1	1.12.3	(v1.12.3	8-0-gbb3e9	9a0 from	n master-1.	.12)]	
Eil	t <u>E</u> dit	⊻iew	<u>G</u> o (Capture	Analyze	Statistics	Telephony	Tools	Internal	s Hel	p									
0	0	(•	1	8	X 🔁	0, 4	e 👳 🤹	Ŧ 2		(a Q	0 🖻	1	8	8 % 1	11				
F	filter:								۷	Expres	sion	Clear	App	dv	Save	TCP Dealy	HTTP De	lay		
No.		Time		Sourc	æ		Destinat	ion		Ler	igth	Protocol	Timef	from re	equest	le lo				
	1	0.0	00	19	2.16	8.0.	88 192	. 168	.0.9	9	62	TCP				1178-	445	[SYN]	Seq	=0 W
	2	0.0	00	19	2.16	8.0.	99 192	. 168	.0.8	8	62	тср				445→1	178	[SYN,	ACK] Se
	3	0.0	01	19	2.16	8.0.	88 192	. 168	.0.9	9	54	тср				1178→	445	[ACK]	Seq	=1 A
	4	0.0	02	19	2.16	8.0.	88 192	. 168	.0.9	9 :	191	SMB				egot	iate	Prote	locol	Req
	- 5	0.0	03	19	2.16	8.0.	99 192	. 168	.0.8	8 2	251	SMB	0.0	000	7660	legot	iate	Prot	ocol.	Res
	6	0.0	03	19	2.16	8.0.	88 192	. 168	.0.9	9	359	KRB.				S-RE	Q			
	7	0.0	04	19	2.16	8.0.	99 192	. 168	.0.8	8	148	KRB.				RB E	rror	: KRB	5KDC.	ERR
	8	0.0	05	19	2.16	8.0.	88 192	. 168	.0.9	9	359	KRB.				S-RE	Q			
	9	0.0	05	19	2.16	8.0.	99 192	. 168	.0.8	8	148	KRB.				RB E	rror	: KRB	5KDC.	ERR
	10	0 0	00	10	1 10	0 0	00 10 1	100	0 0	0	104	CMD	_	-	_				A number of	Dee

Step 4: The newly created column appears to the left of the Info column. Click and drag your new column to a suitable location where visibility is much better.

Step 5: Wireshark assigned the new **smb.time** column the label *Time from request*. On column heading click right and then, select **Edit Column Details**. Rename it's title to **SMB Delta**. Click **OK**.

Step 6: To examine the SMB response packets with the largest delays, click the newly created **SMB Delta** column heading twice. This will sort packets from high to low.

4			tr-s	mb-slow.pcapng	[Wireshark 1.12	13 (v1.12.3-0-gbb	o3e9a0 from master-1.12)]
<u>File</u> <u>I</u> dd	t <u>V</u> iew <u>Go</u>	apture Analyze Statistics	lelephony <u>look</u> inte	emals <u>H</u> elp			
00		B B X Ø 9, 4	* * 7 2 🛽	0.0.0		5 🗱 🕷	
Filter				v Expression	Clear Apply	Seve TCP De	wly HTTP Delay
No.	Time	SMB Delta	Source	Destination	n	Longth Protocol	Info
132	28,763	13.902527000	192.168.	0.99192.	168.0.88	174 SAMR	LookupDomain response
142	28.922	0.088521000	192.168.	0.99192.	168.0.88	142 SAMR	SetUserInfo2 response
136	28.830	0.064965000	192.168.	0.99192.	168.0.88	170 SAMR	CreateUser2 response
582	90.443	0.003131000	192.168.	0.99192.	168.0.88	451 SMB	Session Setup AndX Re
320	37.193	0.003085000	192.168.	0.99192.	168.0.88	451 SMB	Session Setup AndX Re
75	14.623	0.002264000	192.168.	0.99192.	168.0.88	451 SMB	Session Setup AndX Re.
349	48.562	0.001814000	192.168.	0.99192.	168.0.88	97 SMB	Logoff AndX Response
134	28.764	0.000924000	192.168.	0.99192.	168.0.88	162 SAMR	OpenDomain response
586	90.447	0.000911000	192.168.	0.99192.	168.0.88	254 SMB	Trans2 Response, GET_I
351	48.563	0.000888000	192.168.	0.99192.	168.0.88	93 SMB	Tree Disconnect Respon
148	28.940	0.000869000	192.168.	0.99192.	168.0.88	162 SAMR	Close response
100	14.860	0.000860000	192.168.	0.99192.	168.0.88	238 SAMR	EnumDomains response
577	90.440	0.000847000	192.168.	0.99192.	168.0.88	251 SMB	Negotiate Protocol Re

Topic 135: Wireshark Lab 42

In this topic, we examine Server Message Block (SMB) statistics in Wireshark.

Service response times, including SMB and SMB2 response times are created and tracked by Wireshark.

Step 1: Open tr-smb-slow.pcapng.

			tr-smb	-slow.pcapng [Wir	eshark 1.12.3	(v1.12.3-0-gbb	b3e9a0 from master-1.12)]
Ēle	<u>E</u> dit <u>V</u> iew <u>G</u>	o <u>C</u> apture <u>A</u> nalyze <u>S</u> tatistics	Telephony <u>T</u> ools Internal	s <u>H</u> elp			
0	0 🖌 🔳 🕼	(R B X 2 Q +	♦ 🌒 🖥 🕹 🗐 🗐	🛯 🛛 🔍 🖗 🖻	i 🖉 📓	*	
F	ilter:		v	Expression Clear	Apply	Save TCP De	lealy HTTP Delay
No.	Time	SMB Delta	Source	Destination	Leng	gth Protocol	Info
	10.00	D	192.168.0.	.88 192.168	.0.99	62 TCP	1178→445 [SYN] Seq=0 Win=65535 L
	20.00	0	192.168.0.	99 192.168	.0.88	62 TCP	445→1178 [SYN, ACK] Seq=0 Ack=1
	30.00	1	192.168.0.	88 192 . 168	.0.99	54 TCP	1178-445 [ACK] Seq=1 Ack=1 Win=6
	40.00	2	192.168.0.	88 192.168	.0.99 1	L91 SMB	Negotiate Protocol Request
	50.00	3 0.000766000	192.168.0.	99 192.168	.0.88 2	251 SMB	Negotiate Protocol Response
	60.00	3	192.168.0.	88 192.168	.0.99 3	359 KRB5	AS-REQ
	70.004	4	192.168.0.	99192.168	.0.88 1	L48 krb5	KRB Error: KRB5KDC_ERR_C_PRINCIP
	80.00	5	192.168.0.	88 192.168	.0.99 3	359 KRB5	AS-REQ
	90.00	5	192.168.0.	99 192.168	.0.88 1	L48 KRB5	KRB Error: KRB5KDC_ERR_C_PRINCIP
	100.00	6	192.168.0	88 192.168	.0.99 2	294 SMB	Session Setup AndX Request, NTLM
	110.00	7 0.000749000	192.168.0.	99 192.168	.0.88 5	574 SMB	Session Setup AndX Response, NTL
	120.00	8	192.168.0.	88 192.168	.0.99 4	434 SMB	Session Setup AndX Request, NTLM
	130.00	8 0.000689000	192.168.0.	99 192.168	.0.88	93 SMB	Session Setup AndX Response, Err

Step 2: Select **Statistics | Service Response Time | SMB.** As Wireshark prompts you, Click **Create Stat**. For a trace file, the SMB Service Response Time statistics window shows the minimum, maximum and average Service Response Time (SRT).



This SMB Service Response Time statistics window provides a list of all the request procedures.

Topic 136: Wireshark Lab 43

In this topic, we create a button to detect Server Message Block (SMB) Response Times in Wireshark.

In Wireshark, SMB and SMB2 response times can be estimated by using **smb.time** and **smb2.time**. Let's learn how to create a button to detect SMB and SMB2 response times larger than 1 second.

Step 1: Open tr-smb-slow.pcapng.

4		tr-smb-s	slow.pcapng [W	ireshark 1.12	.3 (v1.12.3-0-gbi	b3e9a0 from maste	r-1.12)]	
File Edit View Go	Capture Analyze Statistics Te	lephony <u>T</u> ools Internals	Help					
0 0 🖌 🔳 🔬	🖹 🗎 🗶 🛃 🔍 🔶 🗟	🔿 🐺 🛓 🗏 🖪		1 🖉 🖬	S 🗱 🔡			
Filter: smb.time >	1 smb2.time > 1	v (Expression Clea	r Apply	Save TC De	taly HITTP Delay		
No. Time	SNID DEIT8	Source	Destination		tengin violocol	Info		
10.000		192.168.0.	88192.16	8.0.99	62 TCP	1178→445	[SYN]	Seq=C
20.000		192.168.0.	99192.16	8.0.88	62 TCP	445→1178	[SYN,	ACK]
30.001		192.168.0.	88192.16	8.0.99	54 TCP	1178→445	[ACK]	Seq=1
40.002		192.168.0.	88 192.16	8.0.99	191 SMB	Negotiate	Proto	ocol R
50.003	0.000766000	192.168.0.	99192.16	8.0.88	251 SMB	Negotiate	Proto	ocol R
60.003		192.168.0.	88192.16	8.0.99	359 KRB5	AS-REQ		
70.004		192.168.0.	99192.16	8.0.88	148 KRB5	KRB Error	: KRB	KDC_E
80.005		192.168.0.	88192.16	8.0.99	359 KRB5	AS-REQ		
90.005		192.168.0.	99192.16	8.0.88	148 KRB5	KRB Error	: KRB	KDC_E
100.006		192.168.0.	88192.16	8.0.99	294 SMB	Session S	etup A	AndX R
110.007	0.000749000	192.168.0.	99192.16	8.0.88	574 SMB	Session S	etup A	AndX R
120.008		192.168.0.	88192.16	8.0.99	434 SMB	Session S	etup A	NdX R
130.008	0.000689000	192.168.0.	99192.16	8.0.88	93 SMB	Session S	etub A	ndx R

Step 2: Type **smb.time > 1 || smb2.time > 1** in the display filter area. Click **Save**.

Step 3: You can name your button SMB/SMB2 Delay and then, click OK.

Step 4: Click your newly created **SMB Delay** button. We find that Packet 132 is the only packet that matched your filter.



Topic 137: Wireshark Lab 44

This topic plots Server Message Block (SMB) Response Times in Wireshark.

Let's learn how to plot SMB response times in Wireshark.

- Step 1: Open tr-smb-slow.pcapng.
- Step 2: Select Statistics | IO Graph.
- Step 3: In the Y Axis Unit area, select Advanced...

Step 4: Select the **MAX(*)** Graph 1 Calc option and enter **smb.time** in the Calc area.

Step 5: Click the Graph 1 button to plot the SMB response times.

		Wireshari	K IO Graphs: tr-smb-slow.	pcapng			-
							-
		1 0s		40s 60s	 	i 100s	-
aphs						X Axis	
aphs iraph 1 Color	Filter	Calc: MAX(*)	▼ smb.time	Style: Line	▼ 🗹 Smooth	X Axis Tick interval: 1 sec	5
iaphs iraph 1 Color iraph 2 Color	Filter:	Calc: MAX(*) Calc: SUM(*)	✓ smb.time✓	Style Line Style Line	v v Smooth	X Axis Tick interval: 1 sec Pixels per tick:	5 f day
iaphs iraph 1 Color iraph 2 Color iraph 3 Color	Filter: Filter: Filter	Calc: MAX(*) Calc: SUM(*) Calc: SUM(*)	v smb.time	Style: Line Style: Line Style: Line	v v Smooth v v Smooth v v Smooth v	X Axis Tick interval: 1 sec Pixels per tick: <u>View as time o</u> Y Ame	5 of day
raphs Graph 1 Colur Graph 2 Color Graph 3 Color Graph 4 Color	Fjlter: Fjlter: Filter:	Calc: MAX(") Calc: SUM(") Calc: SUM(") Calc: SUM(")	v smb.time	Style: Line Style: Line Style: Line Style: Line	✓ ✓ Smooth ✓ ✓ Smooth ✓ ✓ Smooth ✓ ✓ Smooth ✓ ✓ Smooth	X Axis Tick interval: 1 sec Pixels per tick: <u>Vicw as time o</u> Y Axis Unit: Advanc	5 of day ed

Step 6: Since SMB and SMB2 use different time fields, we have to create two graphs; one for each delta time.





Step 7: As you click on the highest points in the graph, Wireshark will jump to those packets in the main window for further analysis.

Topic 138: Wireshark's Expert Infos System

This topic provides an overview of Wireshark's Expert Infos System.

There are two buttons and three columns in Wireshark's Status Bar. The first button is named the **Expert Infos** button. Wireshark generates alerts for you to numerous network concerns seen in the trace file as well as packet comments in the Expert Infos window. Expert infos are only a hint and help both novice and expert users to find probable network problems a lot faster, compared to scanning the packet list "manually". The Expert classifies information into 6 categories:

- Errors: red
- Warnings: yellow
- Notes: cyan
- Chats: blue
- Details: grey
- Packet Comments: green

When you click the **Expert Infos** button in the bottom left corner of the Status Bar, the Expert Infos window will open.

Errors: 0 (0)	Warnings: 2 (473)	Notes: 591 (3819)	Chats: 15 (197)	Details: 4489	Packet Comments: 0			
Group	Protocol	Summary				4 Count	•	
E Sequence	TCP	Previous segme	nt not captured (common at ca	pture start)		456	1
🗄 Sequence	ТСР	Out-Of-Order se	gment				17	_
Packet	: (52					1	1
Packet	: (3					1	U
Packet	3781	.0					1	
Packet	3781	1	Exnand a	ection an	d click on a nack	ot	1	
Packet	3841	5	to jump to	that locat	ion in the trace fi		1	
Packet	: 3841		to jump to	linal iocal	ion in the trace in	le	1	
Packet	3841	9					1	

If we understand the causes of the errors, warnings, and notes, then this will help us to figure out what is affecting the network performance. We next look at examples of various commonly occurring Expert warnings, notes, and chats.

Chats: information about usual workflow: e.g. a TCP packet with the SYN flag set

Notes: notable things: e.g. an application returned an "usual" error code like HTTP 404

Warnings: e.g. application returned an "unusual" error code like a connection problem.

Topic 139: Wireshark's Packet Loss Detection

This topic provides an overview of Wireshark's Packet loss detection process.

Most of the time packets are dropped by interconnecting devices such as a switch, a router, a firewall i.e. anything that makes forwarding decisions.

What Causes Packet Loss: The common reasons are a switch or router is overloaded. It is unable to keep up with the required packet forwarding rate. The device is simply faulty.

How Wireshark's detects Packet Loss: The TCP sequencing process is used by Wireshark to detect lost packets. You can also turn off this feature by disabling the *Analyze TCP Sequence Numbers* TCP Preference setting. Click the Preferences button, then expand **Protocols** and choose **TCP**.

	Wireshark: Preferences - Profile: Default
TCP ^	a
TCPENCAP	Show TCP summary in protocol tree
TDMoE	Validate the TCP checksum if possible:
TDS	Menu nuk derarterte te esserandela 1/12 etraseen
Teredo	Hanni sanassetta to tensentale i CP streams.
tetra	Analyze TCP sequence numbers 🗹
TFTP	Relative sequence numbers:
TIME	
TIPC	Scaling factor to use when not available from capture. Not known
TNS	Track number of bytes in flight: 🗹
Token-Ring	Colculate conversation firmed annes 🛛 🐱
TPKT	Concentre contre services and the services (*)
TPNCP	Try heuristic sub-dissectors first:
π	Ignore TCP Timestamps in summary:
UA	
UASIP	Do not call subdissectors for error packets
UAUDP	TCP Experimental Options with a Magic Number:
UBERTOOTH	
UCP	

Wireshark creates a value called *nextseq* (displayed in the Next Sequence Number field). Its value is determined by adding together the Sequence Number field value of each packet to the number of data bytes in it. Therefore, the next expected sequence number from the sender is available in *nextseq*. Wireshark assumes that one or more packets have been lost when it sees sequence number jump beyond the *nextseq* value.

Topic 140: Packet Loss Recovery Methods

In this topic, we understand Fast Recovery and Retransmission Timeout (RTO) methods for Packet loss. An Internet application developer has two choices for the transport layer protocol: Transmission Control Protocol (TCP), or User Datagram Protocol (UDP). TCP provides a packet loss recovery mechanism while UDP does not. Based on this feature, there are applications which can and which cannot tolerate data loss. File transfer, e-mail, and Web documents, instant messaging are examples of those applications which cannot tolerate loss. Loss-tolerant apps include real-time audio/video, stored audio/video, and interactive games.

Packet Loss Recovery Method #1—Fast Recovery

This method can be used only if the receiver supports Fast Recovery. In this method, when the receiver observes a jump in the sequence number value, it will immediately begin sending Duplicate Acknowledgments. Upon receipt of four identical ACKs (the original and three Duplicate ACKs), the sender will retransmit the packet that has been lost.

Packet Loss Recovery Method #2—Sender Retransmission Timeout (RTO)

In this method, a sender maintains a timer for a packet. If the packet has not been acknowledged within Retransmission Timeout (RTO) value, the sender retransmits the packet.

Topic 141: Wireshark Lab 45

In this topic, we apply a Filter to count previous segments in Wireshark.

Step 1: Open tr-general101d.pcapng.

	_						tr-	genera	101d.pcap	ong [Wi	ireshark 1.1	2.3 (v1.12.	3-0-gbb3e	a0 from n	naster-1.12)]
<u>F</u> ile	<u>E</u> dit <u>V</u>	iew <u>G</u> o	<u>C</u> apture	<u>A</u> nalyze	<u>S</u> tatistics	Telephony	<u>T</u> ools	Internals	<u>H</u> elp						
0	•		🖪 🗋	X 🔁	0, 🖗	ې 🔷 🍬	F 🕹		Đ, Q	0 🖭	¥ 🗹	🖪 💥 🕻	1		
F	ilter:							¥	Expression	Clear	Apply	Save	TCP Dealy	HTTP Delay	SMB Delay
No.	Tim	e	Sour	ce		Destinatio	on		Length	Protocol	Info				
	10	. 000	10	.9.9	.9	10.1	10.10).10	1374	ТСР	30000)→1479	ACK] Seq=	=1 Ack=
	2 0	. 000	10	.10.	10.10) 10.9).9.9)	54	тср	1479-	→30000	ACK] Seq=	=1 Ack=
	3 0	. 000	10	.9.9	.9	10.1	0.10).10	1374	TCP	30000)→1479	[ACK] Seq=	=1321 /
	4 0	. 000	10	.9.9	.9	10.1	0.10).10	1374	TCP	30000)→1479	[ACK] Seq=	=2641 /
	5 0	. 000	10	.10.	10.10) 10.9	.9.9)	54	тср	1479-	→30000	[ACK	Seq=	=1 Ack=
	6 0	. 000	10	.9.9	.9	10.1	0.10).10	1374	тср	30000)→1479	[ACK	Seq=	=3961 /
	70	. 000	10	.9.9	.9	10.1	0.10).10	1374	тср	30000)→1479	[ACK	Seq=	=5281 /
	8 0	. 000	10	.10.	10.10	10.9	.9.9)	54	тср	1479-	→30000	[ACK	Seq=	=1 Ack=
	90	. 004	10	.9.9	.9	10.1	0.10).10	1374	тср	30000)→1479	[ACK	Seq=	=6601 /
	100	. 004	10	.9.9	.9	10.1	0.10).10	1374	тср	30000)→1479	[ACK] Seq=	=7921 /
	110	. 004	10	.10.	10.10	10.9	.9.9)	54	тср	1479-	→30000	[ACK	Seq=	=1 Ack=
	12 0	. 004	10	.9.9	.9	10.1	0.10).10	1374	ТСР	30000)→1479	[АСК	Seq=	=9241 /
	130	. 004	10	.9.9	.9	10.1	0.10).10	1374	тср	30000)→1479	[ACK] Seq=	=10561
<													_		

Step 2: In the display filter area, enter the filter **tcp.analysis.lost_segment**. Click **Apply**.

					tr-gene	eral101d.pca	png (Wi	reshark 1.1	12.3 (v1.12.	.3-0-gbl	b3e9a0 from ma	ister-1.1	2)]		-	ā
<u>File</u> Edit y	<u>V</u> iew <u>G</u> o	<u>C</u> apture <u>A</u> nalyze	<u>Statistics</u>	Telephony	<u>T</u> ools <u>I</u> ntern	als <u>H</u> elp										
0 0 🖌	∎ ₫	B D X 2	0, 6	🔶 🌖 🍹	1	🖪 Q, Q	0, 🖻	X	5 % 5							
Filter: t	tcp.analysis.l	ost_segment				Epression	Clear	Apply	Save	TCP De	aly HTTP Delay	5MB Delay				
No. Tin	me	Source		Destination	1	Length	Protocol	Info							_	
4204 1	053	10.9.9	.9	10.1	0.10.1	0 1374	TCP	[TCP	Previ	ous	segment	not	captured]	30000→1479	[ACK]	2
0417 3	3.014	10.9.9	.9	10.1	0.10.1	0 1374	TCP	[TCP	Previ	ous	segment	not	captured]	30000→1479	[ACK]	2
1499 3	3.375	10.9.9	.9	10.1	0.10.1	0 1374	TCP	[TCP	Previ	ous	segment	not	captured]	30000→1479	[ACK]	2
5699 5	.715	10.9.9	.9	10.1	0.10.1	0 1374	TCP	[TCP	Previ	ous	segment	not	captured]	30000→1479	[ACK]	5
2016 9	5.274	10.9.9	.9	10.1	0.10.1	0 1374	TCP	[TCP	Previ	ous	segment	not	captured]	30000→1479	[ACK]	2

From the Status Bar, we can observe that Wireshark has detected packet loss five times in this trace file. In the Info column, we can see "Previous Segment Not Captured". This warning means that Wireshark was not able to see the previous packet(s) in a TCP communication. Wireshark keeps a track of the packet ordering using TCP Sequence Numbers. From this, it can easily detect when packets are missing.

Step 3: As you finish analyzing the trace file, click **Clear** to remove your filter. Applying the filter **tcp.analysis.lost_segment** on the traffic of a trace file, we easily found that packet loss is occurring. To determine how many packets were actually lost, we need to examine the TCP sequence numbers.

Topic 142: Wireshark Lab 46

In this topic, we add TCP sequencing columns in Wireshark.

In this lab exercise, we will create three columns to help understand how TCP sequencing works in a trace file. Using these columns, we will be able to determine how many packets are actually lost when Wireshark displays *Previous Segment Not Captured*.

Step 1: Open tr-general101d.pcapng.

Step 2: Expand the TCP header in Packet 1.

Step 3: Right-click on the **Sequence Number** field and select **Apply as Column**. Rename this column **SEQ#** by right-clicking on your new **Sequence Number** column and selecting **Edit Column Details**.

Filter:				v Expre	ssion	Clear	Apply	Save	TCP De	aly HTTP Del
No. 1 2 3 4 5 6 7 8 9 10 11 12 13 14	Time 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.000 0.004 0.004 0.004 0.004 0.004 0.004 0.004	Source 10.9.9.9 10.10.10.10 10.9.9.9 10.10.10.10 10.9.9.9 10.10.10.10 10.9.9.9 10.10.10.10 10.9.9.9 10.10.10.10 10.9.9.9 10.10.9.9.9 10.9.9.9 10.10.10.10	Destination 10.10.10.10 10.9.9.9 10.10.10.10 10.10.10.10 10.9.9.9 10.10.10.10 10.10.10.10 10.9.9.9 10.10.10.10 10.9.9.9 10.10.10.10 10.9.9.9 10.10.10.10 10.10.10.10 10.9.9.9		Expai Colla Expai Colla Appl Prep Colo Follo Follo Follo Follo Copy Expo Edit I	nd Subtrees uppe Subtrees nd All uppe All y as Column y as Filter rize with Filter w TCP Stream w UDP Stream w UDP Stream / t t Selected Pao Packet	Cket Bytes			ieq=1 Act ieq=1 Act ieq=2641 ieq=2641 ieq=2641 ieq=5281 ieq=5281 ieq=6601 ieq=7921 ieq=1 Act ieq=9241 ieq=1 Act
<pre>Frame 1: 1374 bytes on wire (10992 bits), 137 Ethernet II, Src: Cisco_00:00:01 (00:0c:ce:00 Internet Protocol Version 4, Src: 10.9.9.9 (1 Transmission Control Protocol, Src Port: 3000 Source Port: 30000 (30000) Destination Port: 1479 (1479) [Stream index: 0] ITCP. Segment Lon: 1320] Sequence number: 1 (relative sequence num </pre>					 Wiki Filter Proto Proto Proto Q Q Disat <u>Goto</u> Show 	Protocol Page Field Reference ocol Help ocol Preference ode As ole Protocol Ive Name o Correspondi v Packet Refere	es ng Packet ence in Nev	r Window	or 00 , . 1 9)	interf):0c:ce: 0.10.10), Seq: 1

4			tr-gene	ral101d.pcapng [Wi	reshark 1.12.3	v1.12.3-0-gbb3e	9a0 from master-1.12)]
Ele	Edit View Go	<u>Capture</u> <u>Analyze</u> <u>Statistics</u>	Telephony <u>T</u> ools Interna	sk <u>H</u> elp			
0	0 🖌 🖬 🔬		• • • • 7 2	🔋 ର୍ ର୍ 🛛	🏽 🕅 🥵 🛊	K 🖬	
F	ilter			Expression Clear	Apply	Save TCP Dealy	HTTP Delay SMB Delay
Vo.	Time	Source	Destination	Length Protocol	SEQ#	Info	
	10.000	10.9.9.9	10.10.10.10	1374 TCP		30000+1479	[ACK] Seq=1 Ack=1
	2 0.000	10.10.10.10	10.9.9.9	54 TCP	1	1479+30000	[ACK] Seq=1 Ack=13
	3 0.000	10.9.9.9	10.10.10.10	1374 TCP	1321	30000→1479	[ACK] Seq=1321 Act
	4 0.000	10.9.9.9	10.10.10.10	1374 TCP	2641	30000+1479	[ACK] Seq=2641 Act
	5 0.000	10.10.10.10	10.9.9.9	54 TCP	1	1479+30000	[ACK] Seq=1 Ack=39
	60.000	10.9.9.9	10.10.10.10	1374 TCP	3961	30000-1479	[ACK] Seq-3961 Ack
	7 0.000	10.9.9.9	10.10.10.10	1374 TCP	5281	30000-1479	[ACK] Seq=5281 Act
	8 0.000	10.10.10.10	10.9.9.9	54 TCP	1	1479+30000	[ACK] Seg=1 Ack=60
	90.004	10.9.9.9	10.10.10.10	1374 TCP	6601	30000-1479	[ACK] Seg-6601 Act
	10 0.004	10.9.9.9	10.10.10.10	1374 TCP	7921	30000-1479	[ACK] Seg=7921 Act
	11 0.004	10.10.10.10	10.9.9.9	54 TCP	1	1479-30000	[ACK] Seg_1 Ack_9:
	12 0.004	10.9.9.9	10.10.10.10	1374 TCP	9241	30000-1479	[ACK] Seg-9241 Act
	13 0.004	10.9.9.9	10.10.10.10	1374 TCP	10561	30000-1479	[ACK] Seg=10561 A
	14 0.004	10.10.10.10	10.9.9.9	54 TCP	1	1479+30000	[ACK] Seq=1 Ack=1

Step 4: Select **Apply as Column** after right-clicking on the **Next Sequence Number** field. Rename this column **NEXTSEQ#** by right-clicking on this newly created column and selecting **Edit Column Details**.

Step 5: Create the 3rd new column by right-clicking on the **Acknowledgment Number** field and select **Apply as Column**. Rename this column **ACK#** as we did in the previous step.

Step 6: Click the **Go To Packet** button on the Main Toolbar. As packet 10417 is one of the packets tagged with the *Previous Segment Not Captured* Expert indication. Enter 10417 and click **Jump To**. From the NEXTSEQ# column, we find that the next packet (after Packet 10,416) from 10.9.9.9 should use sequence number 9,164,761. But SEQ# is equal to 9,175,321 for the next packet from 10.9.9.9.

Eile Edit	View <u>Go</u> Capture	Analyze Statistics Tele	phony <u>1</u>	ools Internals	⊕ SEQ#9	164761 is exp t from 10.9.9.9	ected
Filter: No.	Source	Destination	Protocol	 Express SEQ# 	ion NEXTSEQ#	ACK#	Info
10415	10.10.10.10	10.9.9.9	TCP	1	V	9163441	dberegister
10416	10.9.9.9	10.10.10.10	TCP	9163441	9164761	1	ndmps > db
10417	10.9.9.9	10.10.10.10	TCP	9175321	9176641	1	[TCP Previ
10418	10.10.10.10	10.9.9.9	1	1		9164761	dberegister
10419	10.9.9.9			9176641	9177961	1	ndmps > dbe
10420	10.10.10	but SEQ# 917532	21	1		9164761	[TCP Dup A
10421	10.9.9.9	arrives next	_	9177961	9179281	1	ndmps > db
10422	10.10.10.10	TO'A'A'A	ICF	1		9164761	[TCP Dup A
10423	10.9.9.9	10.10.10.10	TCP	9179281	9180601	1	ndmps > db
10424	10.10.10.10	10.9.9.9	TCP	1		9164761	[TCP Dup A
10425	10.9.9.9	10.10.10.10	TCP	9180601	9181921	1	_ndmps > db

Because of the mismatch, Wireshark tags Packet 10,417 with *Previous Segment Not Captured*. 9,175,321 - 9,164,761 = 10,560 bytes have been lost. The TCP header of Packet 10,417 indicates that it contains 1,320 bytes of data. Assuming all the lost packets to be of 1,320 bytes long, 10,560/1,320 = 8 packets were lost prior packet 10,417.

Topic 143: Wireshark Lab 47

In this topic, we build a bad TCP Filter expression in Wireshark.

We would like to create a button, which when clicked once will view many of the key TCP problems in a trace file.

Step 1: Open tr-general101d.pcapng.

Step 2: Type the following expression in the display filter area: tcp.analysis.flags && !tcp.analysis.window_update. Click the Coloring Rules button and then, look for the **Bad TCP coloring rule**. The filter expression we typed in Step 2 is the same as the **Bad TCP coloring rule** string.

4		Wireshark: Coloring Rules - Profile: Default	- 🗆 🗙
Edit	Filter		Order
		List is processed in order until match is found	
<u>IN</u> ew	Name	String	
Edit	DNS Errors	dos flaos reode > 0	Up
-	Bad TCP	tcp.analysis.flags && !tcp.analysis.window_update	
Enable	HSRP State Change	nsrp.state := 8 a.a. nsrp.state := 10	
Disable	Spanning Tree Topology Cl	nange_stp.type == 0x80	
Disable	OSPF State Change	ospf.msg != 1	Move
Delete	ICMP errors	icmp.type eq 3 icmp.type eq 4 icmp.type eq 5 icmp.type eq 11	selected filter
	ARP	arp	up or down
Manage	ICMP	істр іструб	
Import	TCP RST	tcp.flags.reset eq 1	
F	SCTP ABORT	sctp.chunk_type eq ABORT	
Export	TTL low or unexpected	(!ip.dst == 224.0.0.0/4 && ip.ttl < 5 && !pim) (ip.dst == 224.0.0.0/2	Down
<u>C</u> lear	Chockrum Errore	oth for had1 II in chackrum had1 II ton chackrum had1 II ur	
<u>H</u> elp		<u>Q</u> K <u>A</u> pply	<u>C</u> ancel

Step 3: Click **Save** on the display filter toolbar to create a new button. Name this new button **Bad TCP**.

				tr-general101d.pca	png [Wireshark 1.	.12.3 (v1.12.3-0-gbb	3e9a0 from master-
<u>F</u> ile	<u>E</u> dit <u>V</u> iew	<u>G</u> o <u>C</u> apture <u>A</u> nalyze	Statistics Telephony To	ols <u>I</u> nternals <u>H</u> elp			
0	•	(E 🗋 🗙 🛃	् 🧔 🔹 🕹 🐴	🕹 🔳 📑 🕀 e	Q 🖭 🌉 🖾	🌆 💥 🔀	
Fil	ter: tcp.anal	ysis.flags && !tcp.analysi	s.window_update	✓ Expression.	Clear Apply	Save CP Dea	aly HTTP Delay SMB D
No.	Time	Source	Destination	Length	Protocol Info		
	10.00	0 10.9.9.	9 10.10.	10.10 1374	4 ТСР 3000	0→1479 [ACK] Seq=1 Ack
	20.00	0 10.10.1	0.10 10.9.9).9 54	4 TCP 1479	→30000 [ACK] Seq=1 Ack
	30.00	0 10.9.9.	9 10.10.	10.10 1374	4 TCP 3000	0→1479 [ACK] Seq=1321
	40.00	0 10.9.9.	9 10.10.	10.10 1374	4 TCP 3000	0→1479 [ACK] Seq=2641
	50.00	0 10.10.1	0.10 10.9.9).9 54	4 TCP 1479	→30000 [ACK] Seq=1 Ack
	60.00	0 10.9.9.	9 10.10.	10.10 1374	4 TCP 3000	0→1479 [ACK] Seq=3961
	70.00	0 10.9.9.	9 10.10.	10.10 1374	4 TCP 3000	0→1479 [ACK] Seq=5281
	80.00	0 10.10.1	0.10 10.9.9).9 54	4 TCP 1479	→30000 [ACK] Seq=1 Ack
	90.00	4 10.9.9.	9 10.10.	10.10 1374	4 TCP 3000	0→1479 [ACK] Seq=6601
	100.00	4 10.9.9.	9 10.10.	10.10 1374	4 TCP 3000	0→1479 [ACK] Seq=7921
	110.00	4 10.10.1	0.10 10.9.9).9 54	4 TCP 1479	→30000 [ACK] Seq=1 Ack
	120 00	1 10 0 0	0 10 10	10 10 137/	1 TCD 3000	0_1/70 FACK	1 500-02/1

Step 4: Click the newly created **Bad TCP** button.
File Fai	nt <u>v</u> iew <u>u</u>	<u>io C</u> abi	ture <u>A</u> n	aiyze <u>ə</u> t	atistics	i elepnon	<u>y 1</u> 008	s <u>i</u> nterna	is <u>H</u> eip												
0 0		(B			۵ 🖗	۵	7 ⊉		.	ર્ ૦	**	¥ ¥	8 %								
Filter	tcp.analy	/sis.flags	&& !tcp.	analysis.w	indow_u	pdate		Y	xpression	CI	ear	Apply	Sav	e TC	P Dealy F	HTP Dep		Ba	ad TCP		
о.	Time		Source			Destina	ation		Length	Proto	col Ir	nfo							_		
4204	1.053	3	10.9	.9.9		10.	10.1	.0.10	137	4 TC	P	[TCP	Prev	/ious	seg	ment	not	capt	ture	d] 3	
4206	1.053	3	10.9	.9.9		10.	10.1	.0.10	137	4 TC	P	[TCP	Out-	∙0f-0	rder] 30	000 → 1	L479	[AC	K] S	
0417	3.014	1	10.9	.9.9		10.	10.1	0.10	137	4 TCI	P	[TCP	Prev	/ious	s seg	ment	not	capt	ture	d] 3	
0420	3.014	1	10.1	.0.10	.10	10.	9.9.	9	6	6 TC	P	[TCP	Dup	ACK	1041	8#1]	1479	9→300	000	[ACK	
0422	3.014	1	10.1	.0.10	.10	10.	9.9.	9	6	6 TC	P	TCP	Dup	ACK	1041	8#2]	1479	9→300	000	[ACK	
0424	3.01	5	10.1	0.10	.10	10.	9.9.	9	6	6 TC	P	TCP	Dup	ACK	1041	8#3]	1479	9→300	000	- ACK	
0426	3.01	5	10.1	0.10	.10	10.	9.9.	9	6	6 TC	Р	TCP	Dup	ACK	1041	8#4]	1479	9→300	000	- [ACK	
0428	3.01	5	10.1	0.10	.10	10.	9.9.	9	6	6 TC	P	TCP	Dup	ACK	1041	8#5]	1479	9→300	000	- [ACK	
0430	3.01	5	10.1	0.10	.10	10.	9.9.	9	6	6 TC	P	ТСР	Dup	ACK	1041	8#6]	1479	9→300	000	- [ACK	
0432	3.01	5	10.1	.0.10	.10	10.	9.9.	9	6	6 TC	P	ГТСР	Dup	ACK	1041	<u>-</u> 8#71	1479	9→300	000	- Fack	
0434	3.01	5	10.1	.0.10	.10	10.	9.9.	9	6	6 TC	P	ГТСР	Dup	ACK	1041	8#8]	1479	9→300	000	- Fack	
												-								-	
0000	υυ	υc	ce	υυ	υυ	υυ	υυ	UC	ce	υυ	υυ	UΤ	00	υυ	40	υυ		• • • ;		• •	
0010	05	50	1c	22	40	00	4e	06	e4	60	0a	09	09	09	0a	0a		.P.'	"@.	Ν.	
020	0a	0a	75	30	05	с7	d8	0e	с8	30	6b	4c	e9	f2	50	10		u(0	•••	
	00	20	22	<u> </u>				20	22	22	20	20	46	20	20	20		#		F	2
🥖 🖉 🛛 File	: "C:\Us	. Pac	ckets: 3	1422 • 1	Display	ed: 161	2 (4.3%	6) · Loa	time: 0	.00.812											

There are 1,612 packets (4.3% of the traffic) matching our Bad TCP button. These packets can badly affect e.g. a file transfer process to a noticeable extent. Click **Clear** to remove your filter when you are finished.

Topic 144: Wireshark Lab 48

In this topic, we find packet loss counts with Expert Infos in Wireshark.

Step 1: Open tr-general101d.pcapng.

Step 2: Click the Expert Infos button on the Status Bar.



Step 3: There are 1,616 Expert Infos items in the title bar. There are no entries under the Errors tab as Wireshark could not detect any serious problem in the trace file. When you click the **Warnings** tab, Wireshark has observed 5 packet losses. This is because of the sudden jump in the sequence numbers. Wireshark does not take into

account the size of the increase. From the **Warnings** tab, we can see that there are 8 instances of Out-of-Order segments. If you want to analyze a problem, expand the Expert Infos sections and click on that packet listed. Wireshark will jump to that problem in the trace file. In the end, close the Expert Infos window.

Topic 145: Wireshark Lab 49

In this topic, we find out where packets are being dropped using Wireshark.

Step 1: Open tr-general101d.pcapng.

Step 2: To view the Expert Infos window, click Expert Infos button on the Status Bar.

Step 3: Click the **Warnings** tab and expand the **Previous segment not captured** (common at capture start) section.

Wireshark: 1616 Expert Infos	-	×
Errors: 0 (0) Warnings: 2 (13) Notes: 810 (1601) Chats: 1 (2) Details: 1616 Packet Comments: 0		
Group	 Count 	•
Sequence TCP Previous segment not captured (common at capture start)		5
Packet: 4204		1
Packet: 10417		1
Packet: 11499		1
Packet: 15699		1
Packet: 32016		1
Bequence TCP This frame is a (suspected) out-of-order segment		8
Limit to display filter		
Help	1	<u>C</u> lose

In the listing, let's examine **Packet 10,417.** Click on it and Wireshark will take you to that packet in Main window.

Step 4: After the point of packet loss, we can see numerous Duplicate ACKs after the missing packet indication.

4			tr-genera	101d.pcaprg (Wi	reshark 1.12.3 (v1.12.3-0-gbb3e9a0 from master-1.12)] -	. 8 💽
<u>File</u> Edit	<u>View Go C</u> a	pture Analyze Statistics 1	lelephony ∐ools Internals	Help		
0 0	(8 B X Ø 9, 4	* 🛛 7 2 🗉 🖬	ା ପ୍ ପ୍ ପ୍ 🖻	麗 老 🥦 茶 盟	
Filter			v	Expression Clear	Apply Save TCP Dealy HTTP Delay SMB Delay Bad TCP	
No.	Time	Source	Destination	Length Protocol	Info	
104	17 3.014	10.9.9.9	10.10.10.10	1374 TCP	[TCP Previous segment not captured] 30000→1479 [ACK]	Seq
104	18 3.014	10.10.10.10	10.9.9.9	66 TCP	1479→30000 [ACK] Seq=1 Ack=9164761 Win=32768 Len=0 S	SLE=9
104	193.014	10.9.9.9	10.10.10.10	1374 TCP	30000-1479 [ACK] Seq=9176641 Ack=1 Win=46 Len=1320	
104	203.014	10.10.10.10	10.9.9.9	66 TCP	[TCP Dup ACK 10418#1] 1479→30000 [ACK] Seq=1 Ack=916	64761
104	21 3.014	10.9.9.9	10.10.10.10	1374 TCP	30000→1479 [ACK] Seq=9177961 Ack=1 Win=46 Len=1320	
104	22 3.014	10.10.10.10	10.9.9.9	66 TCP	[TCP Dup ACK 10418#2] 1479-30000 [ACK] Seq=1 Ack=916	64761
104	23 3.015	10.9.9.9	10.10.10.10	1374 TCP	30000→1479 [ACK] Seq=9179281 Ack=1 Win=46 Len=1320	
104	24 3.015	10.10.10.10	10.9.9.9	66 TCP	[TCP Dup ACK 10418#3] 1479→30000 [ACK] Seq=1 Ack=916	64761
104	25 3.015	10.9.9.9	10.10.10.10	1374 TCP	30000-1479 [ACK] Seq=9180601 Ack=1 Win=46 Len=1320	
104	26 3.015	10.10.10.10	10.9.9.9	66 TCP	[TCP Dup ACK 10418#4] 1479→30000 [ACK] Seq=1 Ack=916	64761
104	27 3.015	10.9.9.9	10.10.10.10	1374 TCP	30000-1479 [ACK] Seq=9181921 Ack=1 Win=46 Len=1320	
104	28 3.015	10.10.10.10	10.9.9.9	66 TCP	[TCP Dup ACK 10418#5] 1479-30000 [ACK] Seq=1 ACk=916	64761
104	29 3.015	10.9.9.9	10.10.10.10	1374 TCP	30000-1479 [ACK] Seq=9183241 Ack=1 Win=46 Len=1320	
104	30 3.015	10.10.10.10	10.9.9.9	66 TCP	[TCP Dup ACK 10418#6] 1479→30000 [ACK] Seg=1 Ack=916	64761

10.10.10.10 (the receiver) supports Fast Recovery as it is using Duplicate ACKs. Each Duplicate ACK is a request for sequence number 9,164,761. See Acknowledgment Number field in the Duplicate ACKs. Close the Expert Infos window.



Step 5: Packets were captured upstream from the point of packet loss (closer to the data sender than the point of packet loss) or downstream from the point of packet loss (closer to the receiver than the point of packet loss)?

Step 6: We can use the sequence number of the packet that is missing to determine if we see the original and the Retransmission or just Retransmission. Enter **tcp.seq==9164761** in display filter area.

d t-gr	eneral101d.pcapng (Wireshark 1.12.3 (v1.12.3-0-gbb3e#a) from master-1.12))	- 6
[ie [dit View Go Capture Analyze Statistics Telephony Iools Int	ternals Llelp	
001111B0X804++++	3 🖬 Q, Q, Q, [[] 🕷 🛛 🧏 🕷 📓	
Filter: top.seq==916/761	Pression Clear Apply Save TCP Dealy HTTP Delay SM8 Delay Bad TCP	
ee Time Source Destination	Length Pretocol Into	
120353.480 10.9.9.9 10.10.10.	.10 1374 TCP [TCP Fast Retransmission] 30000-1479 [ACK] S	eq=9164761

We can see that traffic has been captured downstream from point of packet loss. Packet loss always affects network performance. Locate it and then figure out why it is happening.

Topic 146: Duplicate ACKs and their Causes

This topic describes duplicate ACKs and their causes.

Duplicate ACKs are used to inform the sender about packet loss. They can also be an indication of out-of-order packets. A host generates Duplicate ACKs if it supports Fast Recovery and notices that a packet has arrived with a sequence number beyond the calculated next sequence number. Details of Fast Recovery can be found in RFC 5681, "TCP Congestion Control." To determine whether an ACK is a duplicate ACK,

Wireshark uses Data bytes, Window Size, Sequence Number, and ACK Number fields. When two or more packets with same Data bytes, Window Size, Sequence Number, and ACK Number fields arrive at a host from a source, we mark the second and onwards as duplicates of the first ACK. In Duplicate ACKs, the ACK Number field value indicates the requested sequence. A TCP host keeps on sending Duplicate ACKs until it receives missing packet. When a receiver receives a packet with the incoming sequence number value jumping higher than the expected, it does not know if the packet is lost or it is just out of order and will be arriving soon. Wireshark marks a packet as Out-of-Order if the missing sequence number packet arrives within 3 ms. In case, it arrives later than 3 ms, Wireshark marks it a Retransmission or Fast Retransmission. In Wireshark, we can use **tcp.analysis.duplicate_ack** to determine the number of duplicate ACKs, while **tcp.analysis.out_of_order** is used to keep a record of Out-of-Order packets.

Topic 147: Wireshark Lab 50

In this topic, we use a Filter to count Duplicate ACKs in Wireshark.

Step 1: Open tr-general101d.pcapng.

Step 2: Enter the filter **tcp.analysis.duplicate_ack** in the display filter area and then Click **Apply**. From the Status Bar, we see that Wireshark has detected 1,019 Duplicate ACKs.

Step 3: Expand **TCP header** of a Packet. Click on the **Acknowledgment Number** field and select **Apply as Column**. Name it **ACK#** by right-clicking on this column and selecting **Edit Column Details**. On scrolling through the trace file, you will observe that these Duplicate ACKs are requests for a single missing packet whose sequence number is 9,164,761.

4				_	tr-general	101d.pca	ong (Wir	eshark 1.12	3 (v1.123-0) gbb3e3	a0 from m	vaster-1.	12)]				- 0
file fd	it yev go j	Capture Analyse	Subities	Telephony Io	sis įntemais	Цеф											
0 0		BIXC	0,4	* 4 4 3		0.0	Q, 🗇	2 🛛 💈	8 8								
Filter	top.aralysis.d	auplicate_ack			¥ I	apression	Citar	Apply	Save T	CP Dealy	HTTP Delay	SMB Dela	y Bad TCP				
No.	Time	Source		Destination		Length	Protocol	Acknowledge	ment number	río							
104	20 3.014	4 10.10.	10.10	10.9.9	9.9	66	TCP		916476	1 [TC	P Dup	ACK	10418#1] 1479-	-30000	[ACK]	Seq=
104	22 3.01	4 10.10.	10.10	10.9.9	9.9	66	TCP		916476	1 [TC	P Dup	ACK	10418#2] 1479-	-30000	[ACK]	Seq=.
104	24 3.01	5 10.10.	10.10	10.9.9	9.9	66	TCP	9	916476	1 [TC	P Dup	ACK	10418#3] 1479-	-30000	[ACK]	Seq=
104	26 3.01	5 10.10.	10.10	10.9.9	9.9	66	TCP		916476	1 [TC	P Dup	ACK	10418#4] 1479-	+30000	[ACK]	Seq=
104	28 3.01	5 10.10.	10.10	10.9.9	1.9	66	TCP		916476	1 [TC	P Dup	ACK	10418#5] 1479-	-30000	[ACK]	Seq=
104	30 3.01	5 10.10.	10.10	10.9.9	1.9	66	TCP		916476	1 [TC	P Dup	ACK	10418#6] 1479-	-30000	[ACK]	Seq=
104	32 3.01	5 10.10.	10.10	10.9.9	9.9	66	TCP		916476	1 [TC	P Dup	ACK	10418#7] 1479-	+30000	[ACK]	Seq=
.104	34 3.01	5 10.10.	10.10	10.9.9	9.9	66	TCP		916476	1 [TC	P Dup	ACK	10418#8] 1479-	+30000	[ACK]	Seq=
104	36 3.01	5 10.10.	10.10	10.9.9	9.9	66	TCP	9	916476	1 [TC	P Dup	ACK	10418#9] 1479-	-30000	[ACK]	Seq=
104	38 3.01	5 10.10.	10.10	10.9.9	1.9	66	TCP		916476	1 [TC	P Dup	ACK	10418#1	0] 1479	9-30000	[ACK]	Seq
104	40 3.01	5 10.10.	10.10	10.9.9	9.9	66	TCP		916476	1 [TC	P Dup	ACK	10418#1	1] 1479	9+30000	[ACK]	Seq
104	42 3.01	5 10.10.	10.10	10.9.9	9.9	66	TCP		916476	1 [TC	P Dup	ACK	10418#1	2] 1479	9+30000	[ACK]	Seq
104	44 3.01	5 10.10.	10.10	10.9.9	9.9	66	TCP		916476	1 [TC	P Dup	ACK	10418#1	3] 1479	9-30000	[ACK]	Seq
104	46 3.01	5 10.10.	10.10	10.9.9	9.9	66	TCP		916476	1 [TC	P Dup	ACK	10418#1	4] 1479	-30000	[ACK]	Seq
104	48 3.01	5 10.10.	10.10	10.9.9	9.9	66	TCP		916476	1 [TC	P Dup	ACK	10418#1	5] 1479	9+30000	[ACK]	Seq
104	50 3.010	6 10.10.	10.10	10.9.9	9.9	66	TCP		916476	1 [TC	P Dup	ACK	10418#1	6] 1479	9-30000	[ACK]	Seq

You can observe that there a many Duplicate ACKs that are requests for a single missing packet whose sequence number is 9,164,761. The host continues sending Duplicate ACKs until missing sequence number is resolved.

Step 4: To hide the **ACK#** column from view, right-click the **ACK#** column and select **Hide Column**. Click **Clear** to remove the display filter.

Topic 148: Wireshark Lab 51

This topic finds Duplicate ACKs using Expert Infos in Wireshark.

Step 1: Open tr-general101d.pcapng.

Step 2: Click the Expert Infos button on the Status Bar.

[Wi	reshark: 16	16 Expert In	fos	-	
Errors: 0 (0)	Warnings: 2 (13)	Notes: 810 (1601)	Chats: 1 (2)	Details: 1616	Packet Comments: 0		
Group	Protocol	 Summary 				 Count 	4
Limit to d	lisnlav filter						
Help							Close
Цер							Close

Step 3: Let's examine problems in this trace file by clicking the **Notes** tab. Count the Duplicate ACKs. Wireshark does not count all Duplicate ACKs together, it groups Duplicate ACKs based on their number. In a trace file, assume packet loss occurs twice. Both times the original ACK, Duplicate ACK#1, Duplicate ACK#2, and Duplicate ACK#3 were sent. Wireshark will list 3 Duplicate ACKs with an indication that each occurred twice.

(Wi	reshark: 16	16 Expert li	nfos	-	
Errors: 0 (0)	Warnings: 2 (13)	Notes: 810 (1601)	Chats: 1 (2)	Details: 1616	Packet Comments: 0		
Group	Protocol	 Summary 				 Count 	• /
Sequence	тср	Duplicate ACK	(#1)				5
E Sequence	тср	Duplicate ACK	(#2)				4
E Sequence	TCP	Duplicate ACK	(#3)				4
E Sequence	ТСР	Duplicate ACK	(#4)				4
🗄 Sequence	тср	Duplicate ACK	(#5)				4
🗄 Sequence	тср	Duplicate ACK	(#6)				4
🗄 Sequence	тср	Duplicate ACK	(#7)				4
🗄 Sequence	тср	Duplicate ACK	(#8)				4
🗄 Sequence	тср	Duplicate ACK	(#9)				4
🗄 Sequence	тср	Duplicate ACK	(#10)				4
E Sequence	тср	Duplicate ACK	(#11)				4
E Sequence	тср	Duplicate ACK	(#12)				4
E Sequence	ТСР	Duplicate ACK	(#13)				4 、
Limit to d	isplay filter						
<u>H</u> elp							<u>C</u> lose

Duplicate ACK (#1) occurs 5 times i.e., the Fast Recovery process was launched 5 times. The 4 Duplicate ACK (#2) indications in one case receiver only had to send the first Duplicate ACK and then it recovered.

4		Wi	reshark: 16	16 Expert li	nfos	-	
Errors: 0 (0)	Warnings: 2 (13)	Notes: 810 (1601)	Chats: 1 (2)	Details: 1616	Packet Comments: 0		
Group	Protocol	 Summary 				 Count 	4 ^
Sequence	тср	Duplicate ACK	(#1)				0
Packet	: 104	20					1
Packet	: 120	139					1
Packet	: 157	700					1
Packet	158	45					1
Packet	: 320	117					1
🗉 Sequence	тср	Duplicate ACK	(#2)				4
Packet	: 104	22					1
Packet	: 120	41					1
Packet	: 157	702					1
Packet	: 158	147					1
Sequence	тср	Duplicate ACK	(#3)				4
E Sequence	тср	Duplicate ACK	(#4)				4 🗸
Limit to d	isplay filter						
<u>H</u> elp							<u>C</u> lose

Probably, the one out-of-order packet situation in the trace file occurs near Packet 3,217.

Step 4: Scroll to the end of the Notes section.

Errors: 0 (0)	Warnings: 2 (13)	Notes: 810 (1601)	Chats: 1 (2)	Details: 1616	Packet Comments: 0			
Group 🖣	Protocol	 Summary 				 Cour 	nt ·	•
E Sequence	TCP	Duplicate ACK	(#798)					1
E Sequence	TCP	Duplicate ACK ((#799)					1
E Sequence	TCP	Duplicate ACK ((#800)					1
E Sequence	TCP	Duplicate ACK ((#801)					1
E Sequence	TCP	Duplicate ACK ((#802)					1
E Sequence	TCP	Duplicate ACK ((#803)					1
E Sequence	TCP	Duplicate ACK ((#804)					1
E Sequence	TCP	Duplicate ACK ((#805)					1
E Sequence	TCP	Duplicate ACK ((#806)					1
E Sequence	ТСР	Duplicate ACK	#807)					1
Sequence	ТСР	Duplicate ACK ((#808)					1
Packet:	1203	34						1
E Sequence	TCP	This frame is a (suspected) f	ast retransmiss	ion			2 🗸

The TCP receiver kept on requesting for a missing packet 809 times. This means a packet got lost and the recovery took a significant amount of time. This can be due to path latency or a network disconnect. Finally, close the Expert Infos window.

Topic 149: Wireshark Lab 52

In this topic, we determine if Selective ACK (SACK) is in use by examining Duplicate ACKs in Wireshark.

When two TCP hosts are communicating and are using SACK, only the missing packets are retransmitted. If SACK is not in use many unnecessary retransmissions will be generated. The sender retransmits every data packet starting from the missing sequence number. If the TCP handshake packets are not captured, we can examine duplicate ACKs to determine if SACK is enabled on the connection.

Step 1: Open tr-general101d.pcapng.

Step 2: Click the Expert Infos button on the Status Bar.

Step 3: Duplicate ACKs can be located by clicking on the Notes tab.

4	Wireshark: 1616 Expert Infos	-		ĸ
Errors: 0 (0) Warnings: 2 (13) Notes: 810 (1601) Chats: 1 (2) Details: 1616 Packet Comments: 0			
Group	Summary	 Count 	•	
Sequence TCP	Duplicate ACK (#1)		5	
	Duplicate ACK (#2)		4	
	Duplicate ACK (#3)		4	
	Duplicate ACK (#4)		4	
	Duplicate ACK (#5)		4	
	Duplicate ACK (#6)		4	
	Duplicate ACK (#7)		4	
	Duplicate ACK (#8)		4 🗸	
Limit to display filter				
<u>H</u> elp			<u>C</u> lose	

Step 4: Expand the **Duplicate ACK (#1)** line and click on the first packet listed, whose number is 10,420. Go to Wireshark to examine that packet.

4		Wir	eshark: 16	16 Expert In	ifos	-		x
Errors: 0 (0)	Warnings: 2 (13)	Notes: 810 (1601)	Chats: 1 (2)	Details: 1616	Packet Comments: 0			
Group 4	Protocol	 Summary 				 Count 	•	\wedge
😑 Sequence	тср	Duplicate ACK	(#1)				5	
Packet:	104	120						
Packet:	120	139					1	
Packet:	157	700					1	
Packet:	158	145					1	
Packet:	320)17					1	
	тср	Duplicate ACK	(#2)				4	
	ТСР	Duplicate ACK	(#3)				4	~
Limit to di	splay filter							_
Help							<u>C</u> lose	

Step 5: SACK Left Edge (SLE) and SACK Right Edge (SRE) information is listed in the **Info** column. You can also expand packet 10420 and look for the SLE and RLE values in the Options area of the TCP header.



Topic 150: Out-of-Order Packets and their Causes

This topic describes out-of-order Packets and their causes.

On a receiving host, the transmission control protocol (TCP) layer cannot pass the received data up to the application layer until all the bytes are in the correct order.

Let's assume a scenario where two packets arrive at a host in reverse order. The packets both arrive within 1 ms. This situation is unlikely to cause a problem. Therefore, we may expect that performance will not be affected if there is a little time between the expected arrival and the actual arrival of out-of-order packets. Wireshark labels a packet out-of-order if it

- (a) Contains data,
- (b) Does not advance the sequence number value, and
- (c) Arrives within 3 ms of the highest sequence number seen.

Causes of Out-of-Order Packets

- a stream using multiple different speed paths to reach the target,
- a poorly configured queuing along a path or,
- Asymmetric routing.

A queuing device that does not forward packets in a first-in/first-out (FIFO) order can cause packets to arrive a host in an out-of-order fashion.

Topic 151: Wireshark Lab 53

This topic uses a filter to count out-of-order packets in Wireshark.

Step 1: Open tr-general101d.pcapng.

Step 2: Enter the expression **tcp.analysis.out_of_order** in the display filter area. Click **Apply**.

4		tr-general101d.pc	apng (Wireshark 1.1.	2.3 (v1.12.3-0-gbb3e9a0 fro	m master-1.12)]		
Ele Edit Yew Go Cepture	Analyze Statistics Telepho	ny Iook Internels Help					
0041481	X 🛛 🔍 + + 4	7 1 E E Q (9.9.0	8 % 11			
Filter topanelysisout_of_or	rder	Y Expression	Clear Apply	Save TCP Dealy HITTP D	elay SMB Delay Dod TCF		
No. Time Sou	rce Desti	ation Length	Protocol Info				
4206 1.053 10	0.9.9.9 10	.10.10.10 137	4 TCP [TCP	Out-Of-Order]	30000-1479	[ACK] Seq	=3698641
12249 3.543 10	0.9.9.9 10	.10.10.10 137	4 TCP [TCP	Out-Of-Order]	30000-1479	[ACK] Seq	=9167401
12251 3.543 10	0.9.9.9 10	.10.10.10 137	4 TCP [TCP	Out-Of-Order]	30000-1479	[ACK] Seq	=9168721
12252 3.545 10	0.9.9.9 10	.10.10.10 137	4 TCP [TCP	Out-Of-Order]	30000→1479	[ACK] Seq	=9170041
12254 3.545 10	0.9.9.9 10	10.10.10 137	4 TCP [TCP	Out-Of-Order]	30000-1479	[ACK] Seq	=9171361
12256 3.545 10	0.9.9.9 10	10.10.10 137	4 TCP [TCP	Out-Of-Order]	30000-1479	[ACK] Seq	=9172681
12257 3.545 10	0.9.9.9 10	10.10.10 137	4 TCP [TCP	Out-Of-Order]	30000→1479	[ACK] Seq	=9174001
32018 95.274 10	0.9.9.9 10	10.10.10 137	4 TCP [TCP	Out-Of-Order]	30000-1479	[ACK] Seq	=2750088
0070	ua ua	(5 5	0 05		Ue	C 3 1	
0020			0.05				
			- 00	100 50		л)	
💛 🜌 File: "	C:\Users	Packets: 3742	2 · Displaye	ed: 8 (0.0%) · L	.oad time: 0	:00.812	

We find that Wireshark has detected 8 Out-of-Order packets as indicated by the Status Bar. On examining the packet numbers, we can have an idea that packets

4206 and 32018 are out-of-order packets, while remaining 6 packets belong to a group of out-of-order packets in close proximity.

```
THEFHER FLOROCOL AGIZION 4' 21C' TO'2'2'2 (TO'2'2'2')' DZC' TO'TO'TO'TO (TO'TO'TO'TO'
Transmission Control Protocol, Src Port: 30000 (30000), Dst Port: 1479 (1479), Seq: 3698641, A
  Source Port: 30000 (30000)
  Destination Port: 1479 (1479)
  [Stream index: 0]
  [TCP Segment Len: 1320]
                             (relative sequence number)
  Sequence number: 3698641
  [Next sequence number: 3699961
                                  (relative sequence number)]
  Acknowledgment number: 1
                            (relative ack number)
  Header Length: 20 bytes
 B.... 0000 0001 0000 = Flags: 0x010 (ACK)
  Window size value: 46
  [Calculated window size: 46]
  [Window size scaling factor: -1 (unknown)]
 Checksum: OxdOfa [validation disabled]
  Urgent pointer: 0
 [SEQ/ACK analysis]
   [Bytes in flight: 2640]
   [TCP Analysis Flags]
   [Expert Info (Warn/Sequence)
                                       frame is
                                                a (suspected) out-of-order segment
                                  This
 Timestampsi
```

If you find multiple Out-of-Order packets in close proximity, then it is likely that a set of packets have been lost. These packets are Retransmissions that arrived within 3 ms.

Step 3: When you are done, click the Clear button to remove the filter.

Topic 152: Wireshark Lab 54

In this topic, we find out-of-order packets with Expert Infos in Wireshark.

Step 1: Open tr-general101d.pcapng.

Step 2: On the Status Bar, click the Expert Infos button.

4		Wi	fos	-		×		
Errors: 0 (0)	Warnings: 2 (13)	Notes: 810 (1601)	Chats: 1 (2)	Details: 1616	Packet Comments: 0			
Group	Protocol	 Summary 	1			 Count 		•
Limit to a	display filter							
<u>H</u> elp							<u>C</u> lose	

Step 3: Click the Warnings tab. Expand the Out-of-Order segment section.

Errors: 0 (0) Warnings:	2 (13) Notes: 810 (16)	01) Chats: 1 (2)	Details: 1616	Packet Comments: 0		
iroup • Protocol	 Summary 				 Count 	4
Sequence TCP	Previous se	gment not capture	ed (common	at capture start)		5
Sequence TCP	This frame i	s a (suspected) ou	ut-of-order se	gment		8
Packet:	4206					1
Packet:	12249					1
Packet:	12251					1
Packet:	12252					1
Packet:	12254					1
Packet:	12256					1
Packet:	12257					1
Packet:	32018					1

Click on the first entry i.e. Packet 4,206 and then click the **Close** button and then go to the main Wireshark window.

									tr-gen	neral101d	pcap	ng (Wi	reshark 1.	123 (v1.1)	2.3-0-gł	bb3e9a0 fi	om master-	-1.12)]		
	Eile Edit	liew	<u>Go</u> <u>C</u> ap	ture 🛓	inalyze	Statistics	s Telepho	ny Iool	ls Inter	nak <u>H</u> elp										
	0 0	(<u> </u>	6 M -	X 🕄	0, (6 4 4	74		B 0	Q	0, 🖻	i 🖉 🕅	5 %	H.					
	Filter:									 Expres 	sion	Clear	Apply	Save	TCPE	Dealy HTTP	Delay SMB 0	elay Bad TCP		
N	b.	Tim	N.J.L	Source			Destin	sation		Len	gth .	Protocol	Info	W . + 11		0501			05074	
	420)11	052	10	9.9	.9	10	.10.	10.	1013	74	TCP	3000	00→14	79 ľ	ACK]	Sea=3	696001	Ack=1	Win=4
	420)2 1	052	10	.10.	10.	10 10	.9.9	9.9		54	TCP	1479	9→3000	00 1	ACK]	Seg=1	Ack=3	697321	Win=3
	420	031	053	10	9.9	.9	10	.10.	10.	1013	74	TCP	3000	00→14	79	ACK]	Seq=3	697321	Ack=1	Win=4
ľ	420)41	053	10	9.9	.9	10	.10.	.10.	1013	74	TCP	TCF	Prev	viou	s sec	ment	not ca	ptured	1 3000
1	420)51	053	10	10	10	10 10	9.0	9 9	2020	66	TCP	1470	1→3000	1 00	ACK]	Sen=1	Ack=3	698641	Win=3
ľ	420	061.	.053	10	.9.9	.9	10	.10.	.10.	1013	74	тср	TCF	out.	-0f-	order	1 300	00→147	9 [ACK	Seq=
	420	17 T.	055	10	. 10.	10.	10 10		1.9		54	TCP	1479	/→3000	JU Î	ACK	Seg=1	ACK=5	701281	Win=3
	420	18 1	054	10	9 9	9	10	10	10	1013	74	TCP	3000	0-14	79 1	ACK	Sen-3	701281	Ack=1	win-4
	420	10 1	054	10	10	10	10 10	0 0	0 0	1015	54	TCP	1470	3000		ACK	Seg-1	Ack-3	702601	win-3
	42	10 1	054	10	0 0	10.	10 10	10	10	1012	74	TCP	2000	0.14	70 L	ACK	Seq-1	702601	Ack=1	win=3
	74.	10 1	054	10.	. 9. 9	. 9	10	. 10.	10.	1015	1	TCP	5000	10-14	19 1	ACK	Seq=5	702001	ACK=1	win=4
	42.	111	054	10	.9.9	.9	10	.10.	10.	1013	74	тср	3000	00→14	/9 L	ACK	Seq=3	/03921	ACK=1	Win=4
	42.	121	054	10	10	10	10 10	a c	a a		54	TCD	1470	773000	nn r	vcv]	Con-1	Ack-3	705741	win-2
ŝ	Ena	me	4206	• 1	374	hvt	es o	n wi	ire	(109	92	hit	s) 1	374	hvte	s car	otured	(1099	2 hits) on i
	E+k	arn	а+ т	÷	Sec		600	00.0		1 (0	0.0	00.00	a.00.	00.0	1)	Det	Cisco	00.00	.00 (0	0.00.00
	Test	ier II		± ,	510.		300_	- 4		1	Å.	00.0	0 (10	00.0	-/, -//	Dot.	10 1	0 10 1	0 (10	10 10
-	Int	ern.	et P	rot	000	i ve	1510	11 4,	, 51	C: 1	.0.	9.9.	3 (10	1.9.9	. 9),	DST:	10.1	0.10.1	0 (10.	10.10
		and the second	1001	0.0	C			TOCC		E B C	00		2 C		11 11 1 1	0.00				

Step 4: If you have previously created, **SEQ#**, **NEXTSEQ#** and **ACK#** columns, then right-click on any column heading and display them. Otherwise, create them. Select **Sequence number** field in the TCP header of a packet, and then **Apply as a Column**. Rename the column as **SEQ#** by right-clicking on the column and selecting **Edit Column Details**.

4												t	r-ger	neral 1	01d.p	capr	ng (Wires	shark	1.12.	3 (v	1.12	.3-0-g	bb3	le9a0 fr	om n	naster	-1.12)]		
File	Edit	<u>v</u>	icw	Go	Capt	ture	Analyz	e 9	tatistics	Te	lephony	<u>I</u> cols	Inter	mals	Hdp																
0	٥	£		ß			8		0, () 📦 i	F±			Ð	e,	0,		a 1	8	*		Ū.								
F	ilter:													v Ex	pressio	on	Cle	ar	Арр	ly	Sa	we	TCP	Deal	у НТТР	Delay	SMB	Delay	Bad T	СР	
No.			TCP	Delta	5		SEC	ŧ		N	ISEQ#		AC	K#			Info														
	420	00	0.	. 0(000)81	.03	69	468	1	369	600	1			1	30	00	0→1	147	9	[A	ACK]		seq=	=36	94(581	AC	k=:	1
	42()1	0.	.0(000)15	03	69	600	1	369	732	1			1	30	00	0→1	147	9	[A	CK]	1	Seq=	=36	96(001	AC	k=:	1
	42()2	0	. 0(000)12	0			1				369	973	21	14	79	→3(000	0	[A	(CK	1	seq=	-1	Ac	(=3)	697	32	1
	42()3	0.	. 0()13	371	03	69	732	1	369	864	1			1	30	00	0→1	147	9	[A	(CK		Seq=	=36	973	321	AC	k=:	1
	420)4	0.	. 0(000)24	03	69	996	1	370	128	1			1	[Τ	СР	Pi	rev	io	us	; se	egi	ment	t n	ot	ca	ptu	ire	d]
	42()5	0.	. 0(000)24	0			1				369	986	41	14	79	→3(000	0	[/	\CK]		Seq=	=1	Ac	(=3)	598	64.	1
	42()6	Q.	. 0(000)61	<u>J</u> B	69	864	1	369	996	1			1	[1	СР	0ι	ut-	0f	-0)rde	er.] 30	000	0→:	L47	9 [AC	К]
	42()7	0.	. 00)0()17	0			1				370)12	81	14	79	→3(000	0	[A	\CK]	1	Seq=	:1	Acl	(=3)	701	.28	1
	420)8	0	. 00	002	229	03	70	128	1	370	260	1			1	30	00	0→1	147	9	[4	CK]	1	Seq=	-37	012	281	Ac	k=:	1
	42()9	0.	. 0()00)18	0			1				370)26	01	14	79	→3(000	0	[A	CK]		Seq=	-1	Ac	(=3)	702	60	1
	4.7.4		0	0.	0.00	1 1 1	0.0	70	200		370	202					20	00	0 1		0	г.	7		-	27	0.04	101		1 .	

We can see that **SEQ#** of Packet 4,206 is lower than the **NSEQ#** in Packet 4,204. Also, the out-of-order packet arrived 85 microseconds, which is less than 3 ms after the previous packet from 10.9.9.9.

Topic 153: Causes of Fast Retransmissions

In this topic, we describe fast retransmissions and their causes. Fast retransmissions are triggered, when three identical ACKs (the original ACK and two Duplicate ACKs) arrive at a receiving host. Fast Retransmissions are a sign that packet loss has occurred and are part of the Fast Recovery process.

Characteristics of Fast Retransmissions

The segment contains data or has the SYN or FIN bits set to 1. The segment does not advance the sequence number. At least 2 Duplicate ACKs were coming from the reverse direction. The Sequence Number field value matches Acknowledgment Number field value in the preceding Duplicate ACKs. The packet arrived within 20 ms of the last Duplicate ACK.

Causes of Fast Retransmissions

When a host believes there is a packet loss and supports Fast Recovery, then it generates at least four identical ACKs. In addition to the original ACK, at least three Duplicate ACKs are sent towards the sender requesting the missing segment. Packets are typically lost at infrastructure devices.

Topic 154: Wireshark Lab 55

This topic uses a filter to count fast retransmission packets in Wireshark.

Step 1: Open tr-general101d.pcapng.

Step 2: Enter the filter **tcp.analysis.fast_retransmission** in the display filter area. Click **Apply**.

									tr-gene	eral101	d.pcap	ng (V	/iresha	rk 1.12.	.3 (v1.12.3·	-0-gt
<u>F</u> ile	<u>E</u> dit <u>V</u> iew	<u>G</u> o <u>C</u>	apture 🧕	<u>A</u> nalyze	<u>S</u> tatistic	s Tel	lephony	Too	ls <u>I</u> ntern	als <u>H</u> el	р					
0	0 煮 📕		b	X 🔁	Q	() ()	> 🏟	7 1			QQ	0] 🏹	i 🗹 🍢] 💥 💢	
Fil	ter: tcp.ar	nalysis.fas	t_retransn	mission	٦					 Expres 	sion	Clea		Apply	Save	тср р
No.	500	nee			estinati	on			Length	Protoco	l Info	,				
	11	0.9.	9.9		10.	10.	10.	10	1374	TCP	3	000	0→1	479	[ACK]] S
	21	0.10).10	.10	10.	9.9	.9		54	тср	1	479	→30	000	[ACK]	S
	31	0.9.	9.9		10.	10.	10.	10	1374	тср	3	000	0→1	479	[ACK]	S
	41	0.9.	9.9		10.	10.	10.	10	1374	тср	3	000	0→1	479	[ACK]	S
	51	0.10).10	.10	10.	9.9	.9		54	тср	1	479	→30	000	[ACK]	S
	61	0.9.	9.9		10.	10.	10.	10	1374	TCP	3	000	0→1	479	[ACK]	S
	71	0.9.	9.9		10.	10.	10.	10	1374	тср	3	000	0→1	479	[ACK]	S
	81	0.10).10	.10	10.	9.9	.9		54	тср	1	479	→30	000	[ACK]	S
	91	0.9.	9.9		10.	10.	10.	10	1374	тср	3	000	0→1	479	[ACK]	S
	10 1	0.9.	9.9		10.	10.	10.	10	1374	тср	3	000	0→1	479	[ACK]	S
	11 1	0.10	0.10	.10	10.	9.9	.9		54	тср	1	479	→30	000	[ACK]	S
	10.1	0 0	0 0		10	10	10	10	1 7 7 4		2	000	0 1	170	E A OV	

In this trace file, Wireshark has detected two Fast Retransmissions as indicated by the Status Bar.



Step 3: Expand the **[SEQ/ACK analysis]** section on one of the Fast Retransmissions. Wireshark colors this area cyan. Color of the Expert Infos Notes is Cyan.

No. Source	Destination	Length Protocol	Info			
12035 10.9.9.9	10.10.10.1	01374 TCP	[TCP Fast	Retransmission	30000→1479	[ACK] Sec
12248 10.9.9.9	10.10.10.1	01374 тср	[TCP Fast	Retransmission	30000→1479	[ACK] Sec
			-			
<						
Acknowledgment	number: 1	(relat	ive ack nu	mber)		
Header Length:	20 bytes					
0000 0001	0000 = Fla	gs: 0x010	(ACK)			
Window size va	lue: 46	-				
[Calculated wi	ndow size:	461				
[Window size s	caling fact	or: -1 (u	nknown)]			
Checksum: Oxh9	58 [validat	ion disah	ledl			
Urgent nointer	· 0	ion disub	icaj			
- [cco/ack analy						
ESEQ/ACK analy	515]	~ 1				
[Bytes in fli	ght: 109032	0]				
[TCP Analysis]	Flags]					
Expert Info	(Note/Sequ	ience): тh	is frame i	s a (suspected)	fast retrans	smission]
Expert Info	(Note/Sequ	ence): Th	is frame i	s a (suspected)	retransmiss	ion]
Timestamps]						-

There are two Expert Infos indications marked by Wireshark on this one packet. Frame is marked to be a Fast Retransmission and a Retransmission. Fast Retransmission is a Retransmission.

Step 4: Click the **Clear** button to remove the filter as you finish your analysis.

Topic 155: Wireshark Lab 56

In this topic, we find fast retransmission packets with Expert Infos in Wireshark.

Step 1: Open tr-general101d.pcapng.

Step 2: Click the Expert Infos button on the Status Bar.

Step 3: Click the **Notes** tab. Scroll through the list. As you find the Fast Retransmissions section, expand it and click on the first entry, **Packet 12,035**.

Errors: 0 (0)	Warnings: 2 (13)	Notes: 810 (1601)	Chats: 1 (2)	Details: 1616	Packet Comments: 0			
Group	Protocol	 Summary 				 Count 	1	,
E Sequence	TCP	Duplicate ACK	(#801)				1	
E Sequence	тср	Duplicate ACK	(#802)				1	
E Sequence	тср	Duplicate ACK	(#803)				1	
E Sequence	ТСР	Duplicate ACK	(#804)				1	
Sequence	тср	Duplicate ACK	(#805)				1	
E Sequence	тср	Duplicate ACK	(#806)				1	
E Sequence	тср	Duplicate ACK	(#807)				1	
E Sequence	тср	Duplicate ACK	(#808)				1	
Sequence	тср	This frame is a	(suspected) f	fast retransmiss	ion		2	١
Packet	: 120	35					1	I
Packet	122	48					1	I
E Sequence	ТСР	This frame is a	(suspected) i	retransmission			580	J
Limit to d	isplay filter							

We can observe that there are 808 Duplicate ACKs before the Fast Retransmission packet. Return to the main Wireshark window after clicking the **Close** button.

Step 4: If you have created **SEQ#, NEXTSEQ# and ACK#** columns, then you can display them by right-clicking on a column and selecting them from Displayed Columns.

<u>F</u> ile J	<u>E</u> dit <u>V</u> iew	<u>G</u> o <u>C</u> apture <u>A</u> r	nalyze <u>S</u> tati	istics Teleph	non <u>y T</u> ools <u>I</u> nterna	ls <u>H</u> elp								
0 0) 🖌 🔳	(B)	(2 9	(🏟 🛊 i	a 🕈 🕹 🔳 🛙	🛯 🔍 ପ୍ ପ୍	1	5 🕺 🛛	a de la compañía de					
Filte	n				v	Expression C	lear Apply	Save	TCP Dealy HTTP	Delay SMB Delay	Bad TCP			
No.	2029	CP Delta	SEQ#)1171	NSEQ# LUZ JZ 441	ACK#	Source	ษ.ษ	Destination	Length .V. IVIJ/4	Protocol + ICP	Info SUUUU→1479	LACK	seq=
17	2030	0.000010	0	1		9164761	l 10.10	.10.1	010.9.9.	9 74	1 TCP	[TCP Dup A	CK 104	18#80
17	2031	0.000029	0 102	52441	10253761	. 1	10.9.	9.9	10.10.1	0.101374	4 тср	30000→1479	[ACK]	Seq=
17	2032 (0.00009	0	1		9164761	10.10	.10.1	010.9.9.	9 74	4 TCP	[TCP Dup A	CK 104	18#80
1	2033	0.002698	0 102	53761	10255081	1	10.9.	9.9	10.10.1	0.101374	1 ТСР	30000→1479	[ACK]	Seq=
17	2034	0.000015	0	1		9164761	10.10	.10.1	010.9.9.	9 74	4 TCP	[TCP Dup A	CK 104	18#80
1	2035	0.000033	0 916	64761	9166081	1	10.9.	9.9	10.10.1	0.101374	4 TCP	[TCP Fast	Retran	smiss
17	2036	0.000204	0 102	55081	10256401	. 1	10.9.	9.9	10.10.1	.0.101374	1 ТСР	30000→1479	[ACK]	Seq=
17	2037	0.000011	.0	1		9166081	L 10.10	.10.1	010.9.9.	9 74	4 тср	1479→30000	[ACK]	Seq=
17	2038	0.000019	0 102	56401	10257721	. 1	10.9.	9.9	10.10.1	0.101374	4 тср	30000→1479	[ACK]	Seq=
17	2039 (0.00009	0	1		9166081	L 10.10	.10.1	010.9.9.	9 74	4 TCP	[TCP Dup A	CK 120	37#1]

From the TCP Delta column, we can see that the Fast Retransmission occurred within 20 ms of the last Duplicate ACK. If you consider time, you will find that only about 465 ms have passed between when packet with sequence number 9,164,761 went missing and when Fast Retransmission was initiated. Although there were 808 Duplicate ACKs, but this ½–second delay would be unnoticeable by the user.

Topic 156: Causes of Retransmissions

This topic describes retransmissions and their causes.

Wireshark considers a packet to be a Retransmission if the segment contains data or has the SYN or FIN bits set to 1. Sequence number is not advanced by the segment.

Duplicate ACKs have not triggered Retransmission. The segment arrives > 3 ms later than the previous packet with a higher sequence number. A Retransmission Time Out (RTO) at a sender triggers Standard Retransmissions. The RTO timer is used to ensure data delivery continues even if the TCP peer stops communicating (with ACKs). Each TCP host calculates and maintains a RTO timer. This timer value is based on the round trip time learned through previous data transmissions and related acknowledgments. The RTO value consistently changes through the conversation.

What Causes Retransmissions?

A TCP host begins counting down the RTO, when it sends a data packet. The sender retransmits the unacknowledged data packet, if the RTO timer expires without receiving an ACK for the data packet. The sender has no idea whether the original packet was lost or the acknowledgment was lost. The sender just knows that there some problem as it did not receive an ACK within the RTO.

Topic 157: Wireshark Lab 57

This topic uses a filter to count retransmission packets in Wireshark.

Applying **tcp.analysis.retransmission** in Wireshark as a filter for Retransmissions will also display Fast Retransmissions. You need to exclude Fast Retransmissions in the filter, if you are interested in seeing packets that are only labeled as standard Retransmissions.

Step 1: Open tr-general101d.pcapng.

Step 2: Enter the filter **tcp.analysis.retransmission** in the display filter area. Click **Apply**.

File Edit View Go Capture Analyze Statistics Telephony Tools Internal	als Help
◎ ◎ ∡ ■ ∡ ⊟ ≥ ≈ ≈ = 0, + + + + ∓ ± ■	🖬 Q, Q, Q, 🔟 🐺 🔟 🥵 🛠 🔛
Filter: tcp.analysik.retransmission v	Fxpression Clear Apply Save TCP Dealy HTTP Delay SMB Delay Bad TCP
12035 0.0000330 9166081 10.9.9.9	10.10.10.101374 TCP [TCP Fast Retransmission] 30
12248 0.0000190 9167401 10.9.9.9	10.10.10101374 TCP [TCP Fast Retransmission] 3
12259 0.0012310 9890761 10.9.9.9	10.10.101374 TCP [TCP Retransmission] 30000-1
12260 0.0002120 9892081 10.9.9.9	10.10.10101374 TCP [TCP Retransmission] 30000-1
12262 0.0000210 9893401 10.9.9.9	10.10.1010374 TCP [TCP Retransmission] 30000-1
12264 0.0017130 9894721 10.9.9.9	10.10.101374 TCP [TCP Retransmission] 30000-:
12265 0.0000250 9896041 10.9.9.9	10.10.101374 TCP [TCP Retransmission] 30000-:
12267 0.0002250 9897361 10.9.9.9	10.10.101374 TCP [TCP Retransmission] 30000-1
12269 0.0000180 9898681 10.9.9.9	10.10.10101374 TCP [TCP Retransmission] 30000-1
12270 0.0016360 9900001 10.9.9.9	10.10.10101374 TCP [TCP Retransmission] 30000+:
12272 0 0000180 9901321 10 9 9 9	10 10 10 101374 TCP [TCP Retransmission] 30000+
ESEQ/ACK analysis]	
[Bytes in flight: 1090320]	
□[TCP_Analysis_Flags]	
[Expert Info (Note/Sequence) [Expert Info (Note/Sequence)	: This frame is a (suspected) fast retransmission] : This frame is a (suspected) retransmission]

From the Status Bar, we can see that Wireshark has detected 580 Retransmissions in the trace file. This also includes two Fast Retransmissions.

Step 3: Update the filter to the following: **tcp.analysis.retransmission && !tcp.analysis.fast_retransmission.** Click **Apply**.

Filter	tcp.analysis.retransmissi	on && Itop.analysis.fast	t retransmission	Expression Clear	Apply Save TCP	Dealy HTTP Delay SMB De	lay Bad TCP	
No.	TCP Delta	NSEQ#	Source	Destination	Length Protocol	Info		
1225	9 0.00123	10 9890761	10.9.9.9	10.10.10).101374 тср	[TCP Retra	nsmission]	30000→1479
1226	0 0.00021	20 9892081	L 10.9.9.9	10.10.10).101374 тср	[TCP Retra	nsmission]	30000→1479
1226	2 0.00002	10 9893401	L 10.9.9.9	10.10.10).101374 TCP	[TCP Retra	nsmission]	30000→1479
1226	4 0.00171	30 9894721	L 10.9.9.9	10.10.10).101374 тср	[TCP Retra	nsmission]	30000→1479
1226	5 0.00002	50 9896041	L 10.9.9.9	10.10.10).101374 тср	[TCP Retra	nsmission]	30000→1479
1226	7 0.00022	50 9897361	L 10.9.9.9	10.10.10).101374 тср	[TCP Retra	nsmission]	30000→1479
1226	9 0.00001	80 9898681	L 10.9.9.9	10.10.10).101374 тср	[TCP Retra	nsmission]	30000→1479
1227	0 0.00163	50 9900001	L 10.9.9.9	10.10.10).101374 тср	[TCP Retra	nsmission]	30000→1479
1227	2 0.00001	80 9901321	L 10.9.9.9	10.10.10).101374 тср	[TCP Retra	nsmission]	30000→1479
1584	1 0.03541	301340988	110.9.9.9	10.10.10).101374 тср	[TCP Retra	nsmission]	30000→1479
1591	2 0 02689	701341120	110 9 9 9	10 10 10) 101374 тср	TCP Retra	nsmissionl	30000→1479
<								
U L	jenc porno	er. v						
■ [SE	EQ/ACK ana	lysis]						
[1	Bytes in f	light: 50	05560]					
⊜ [1	TCP Analys	sis Flags						
	[Expert In	nfo (Note,	/Sequence):	This fram	ne is a (sus	pected) ret	ransmissio	nj
	[The RTO 1	for this :	segment was	: 0.004044	1000 seconds]		
(RTO based	d on delta	a from fram	ne: 12246]		_		J

There are 578 packets that match this filter. So the two Fast Retransmissions have been removed from view.

Step 4: Click the Clear button to remove the filter.

Topic 158: Wireshark Lab 58

This topic finds retransmission packets with Expert infos in Wireshark.

Step 1: Open tr-general101d.pcapng.

Step 2: On the Status Bar, Click the Expert Infos button.

Step 3: Click the **Notes** tab. Scroll through list. Expand the **Retransmissions** section and click on **Packet 12,259**. This will take you to the Wireshark Main window. Click the **Close**.

Wireshark: 1616 Expert Infos	-		×
Errors: 0 (0) Warnings: 2 (13) Notes: 810 (1601) Chats: 1 (2) Details: 1616 Packet Comments: 0			
Group Protocol Summary	 Count 	•	^
Gequence TCP This frame is a (suspected) fast retransmission		2	
Sequence TCP This frame is a (suspected) retransmission		580	
Packet: 12035		1	
12210		- 1	
Packet: 12259		1	
Packet: 12200		1	
Packet: 12262		1	
Parket: 12264		1	¥
Limit to display filter			
Help	1	<u>C</u> lose	

No.		Source	Destination	Length Protocol	Info			
12	2244	10.9.9.9	10.10.10.	101374 тср	30000→1479	[ACK] Seq=	.0392361 Ac	k=1 w
12	2245	10.10.10.1	010.9.9.9	74 тср	[TCP Dup AC	ск 12037#104	·] 1479→300	00 [A
12	2246	10.9.9.9	10.10.10.	101374 тср	30000→1479	[PSH, ACK]	Seq=103936	81 Ac
12	2247	10.10.10.1	010.9.9.9	74 тср	[TCP Dup AC	ск 12037#10] 1479→300	00 [A
12	2248	10.9.9.9	10.10.10.	101374 тср	[TCP Fast F	Retransmiss [.]	on] 30000→	1479
12	2249	10.9.9.9	10.10.10.	101374 тср	[TCP Out-Of	-Order] 300	00→1479 [A	CK] S
12	2250	10.10.10.1	010.9.9.9	74 тср	1479→30000	[ACK] Seq=1	. Ack=91687	21 Wi
12	2251	10.9.9.9	10.10.10.	101374 тср	[TCP Out-Of	-Order] 300	00→1479 [A	CK] S
12	2252	10.9.9.9	10.10.10.	101374 тср	[TCP Out-Of	-Order] 300	00→1479 [A	CK] S
12	2253	10.10.10.1	010.9.9.9	74 тср	1479→30000	[ACK] Seq=	. Ack=91713	61 Wi
12	2254	10.9.9.9	10.10.10.	101374 тср	[TCP Out-Of	-Order] 300	00→1479 [A	CK] S
12	2255	10.10.10.1	010.9.9.9	74 тср	1479→30000	[ACK] Seq=1	. Ack=91726	81 Wi
12	2256	10.9.9.9	10.10.10.	101374 тср	[TCP Out-Of	-Order] 300	00→1479 [A	CK] S
12	2257	10.9.9.9	10.10.10.	101374 тср	[TCP Out-Of	-Order] 300	00→1479 [A	CK] S
12	2258	10.10.10.1	010.9.9.9	66 TCP	1479→30000	[ACK] Seq=	. Ack=98894	41 Wi
12	2259	10.9.9.9	10.10.10.	101374 тср	[TCP Retrar	nsmission]	0000→1479	[ACK]

Step 4: If you have previously created, **SEQ#**, **NEXTSEQ#** and **ACK#** columns, then right-click on any column heading and display them. Otherwise, create them. Select **Sequence number** field in the TCP header of a packet, and then **Apply as a Column**. Rename the column as **SEQ#** by right-clicking on the column and selecting **Edit Column Details.** Scroll up and you will find Packet 12,246 is a data packet from 10.9.9.9, which has the highest Sequence No. field value at this point of the trace file. Right-click on **Packet 12,246** and select **Set Time Reference (toggle)**.

📕 tr-general101d.pcapng	120		
Eile Edit View Go Capture Analyze Statistics Telephony	<u>Iools</u> Internals <u>H</u>	lelp	
0 0 ▲ ■ <u>▲</u> 🖨 🖄 🗶 🕄 🔍 ↔ ↔ ↔	PKT12246 from	: Highest SEQ# n 10.9.9.9	PKT12248: Fast Retransmission within 20 ms of Duplicate ACK
No. Time ICO Data Course	Destina	Protocol SEO#	NEXTSEO# K# Info
12241 3.541 12242 3.541 12243 3.541 12243 3.541 Within 3 ms of highest SEQ#	10.9.9 10.10.10. 10.9.9.9	TCP 1 TCP 10391041	66081 ГСР Dup ACK 12037 1 10392361 1 081 ГСР Dup ACK 12037
12244 3.541 0 0098000 10.9.9.9 12245 3.541 0 00010000 10.10.10.10	10.10.10.10	P 10392361	1 10393681 1 9161 81 ГСР Вир АСК 12037
12247 0.000 12248 0.000 0.000011000 10.10.10.10 12248 0.000 0.000019000 10.9.9.9	10.9.9.9 10.10.10.10	TCP 1 TCP 9166081	916608 [TCP Dup ACK 12037 9167401 1 [TCP Fast Retransm
12249 0.000 PKT12259: Retransmission	10.10.10.10 10.9.9.9	тср 9167401 тср 1	9168721 1 [TCP Out-Of-Order] 9168721 dberegister > ndmp
did not advance SEQ#; no 12252 0.002 12253 0.002 related Duplicate ACKs;	10.10.10.10	TCP 9168/21 TCP 9170041 TCP 1	9171361 1 [TCP Out-Of-Order] 9171361 1 [TCP Out-Of-Order] 9171361 dberegister > ndmp
12254 0.002 not within 3ms 12255 0.002 07 0024000 10.10.10.10	10.10.10.10 10.9.9.9	TCP 9171361 TCP 1	9172681 1 [TCP Out-Of-Order] 9172681 dberegister > ndmp
12256 0.002 0 00181000 10.9.9.9 12257 0.002 1 000026000 10.9.9.9	10.10.10.10 10.10.10.10	TCP 9172681 TCP 9174001	9174001 1 [TCP Out-Of-Order] 9175321 1 [TCP Out-Of-Order]
12258 0.002 0.00512000 10.10.10.10 12259 0.004 0.001231000 10.9.9.9 12260 0.004 0.000212000 10.9.9.9	10.9.9.9	TCP 1 TCP 9889441 TCP 9890761	9890761 1 [TCP Retransmissio 9892081 1 [TCP Retransmissio
12261 0.004 0.000016000 10.10.10.10 12262 0.004 0.000021000 10.9.9.9	10.9.9.9 10.10.10.10	TCP 1 TCP 9892081	9892081 dberegister > ndmp 9893401 1 [TCP Retransmissio
12263 0.004 0.000018000 10.10.10.10	10.9.9.9	Janual	9893401 dheregister admo

Step 5: Time Reference, can be removed by right-clicking on **Packet 12,246**, and selecting **Set Time Reference (toggle)**.

Topic 159: Causes of ACKed Unseen Segments

In this topic, we explain ACKed Unseen segments and their causes.

What is ACKed Unseen Segment?

When Wireshark sees an ACK, but it did not see the data packet that is being acknowledged, it calls it an ACK Unseen Segment. For each host, Wireshark keeps a track of sequence number, next sequence number, and acknowledgment number values. Wireshark also keeps a track of a "Maximum Sequence Number to be ACKed" (*maxseqtobeacked*) value. This value gets updated as data is received. If Wireshark sees an ACK to acknowledge a data packet that has a higher Sequence Number field value than *maxseqtobeacked*, then it shows that it has missed the data packet that is being ACKed. Let's assume that Wireshark is running on a client which is communicating with a server. An ACK being sent to acknowledge all sequence numbers up to 9,380 has been captured.

Causes of ACKed Unseen Segment:

A) Problems occurring during the capture process can cause ACKed Unseen Segments. If the switch is oversubscribed when you are using switch port spanning, then it is dropping packets that should have been forwarded to Wireshark.

B) Asymmetric routing can cause ACKed Unseen Segments. With asymmetric routing, data may be flowing along one path on the network while ACKs flow along another path. Wireshark was able to capture ACKs but not the data.

Topic 160: Wireshark Lab 59

In this topic, we use a filter to count ACKed Unseen Segment warnings in Wireshark.

Step 1: Open tr-badcapture.pcapng.

Step 2: Enter the filter expression: **tcp.analysis.ack_lost_segment** in the display filter area. Click **Apply**.

					<u>.</u>	-				
Filter	tcp.analysis.ack_lo	ost_segment		~ Express	ion Clear Ap	oply Sav	e TCP De	aly HTTP Delay	SMB Delay Bad TCP	
No.	Time	Source	Destination	-	Length Protocol	Info				
15	0.087	24.6.17	73.22/23.62.22	8.65	54 TCP	[TCP	ACKed	unseen	segment]	35318→443
18	0.089	24.6.17	73.22 23.62.22	8.65	54 TCP	[TCP	ACKed	unseen	segment]	35318→443
23	0.106	24.6.17	73.22 23.62.22	8.65	54 TCP	[TCP	ACKed	unseen	segment]	35318→443
26	0.107	24.6.17	73.22 23.62.22	8.65	54 TCP	[TCP	ACKed	unseen	segment]	35318→443
31	0.123	24.6.17	73.22 23.62.22	8.65	54 TCP	[TCP	ACKed	unseen	segment]	35318→443
34	0.124	24.6.17	73.22 23.62.22	8.65	54 TCP	[TCP	ACKed	unseen	segment]	35318-443
36	0.138	24.6.17	73.22 23.62.22	8.65	54 TCP	[TCP	ACKed	unseen	segment]	35318→443
40	0.142	24.6.17	73.22 23.62.22	8.65	54 TCP	[TCP	ACKed	unseen	segment]	35318-443
43	0.143	24.6.17	73.22 23.62.22	8.65	54 TCP	[TCP	ACKed	unseen	segment]	35318→443
45	0.144	24.6.17	73.22 23.62.22	8.65	54 TCP	[TCP	ACKed	unseen	segment]	35318→443
49	0.155	24.6.17	73.22 23.62.22	8.65	54 TCP	[TCP	ACKed	unseen	segment]	35318→443
51	0.156	24.6.17	73.22123.62.22	8.65	54 TCP	[TCP	ACKed	unseen	segment]	35318→443
53	0.159	24.6.17	73.22 23.62.22	8.65	54 TCP	[TCP	ACKed	unseen	segment]	35318→443
57	0.161	24.6.17	73.22123.62.22	8.65	54 TCP	[TCP	ACKed	unseen	segment]	35318→443
61	0.163	24.6.17	73.22 23.62.22	8.65	54 TCP	[TCP	аскед	unseen	segment]	35318→443
62	0 165	24 6 17	72 22 22 22 22	0 65	54 TCD	TCD	Acrod	Uncoon	commont]	25210 442

Look at the Status Bar. You will see that Wireshark has detected 24 ACKed Unseen Segments.

Filter	tcp.analysis.ack_lo	ist_segment		✓ Expression		Apply Sa	ive TCP Dealy	HTTP Delay	SMB Delay	Bad TCP
No.	Time	Source	Destination	b	ength Protocol	Info				
15	0.087	24.6.173	. 22 23 . 62 .	228.65	54 TCP	[TCP	ACKed	unseen	segr	ent]
18	0.089	24.6.173	.22 23.62.	228.65	54 TCP	[ТСР	ACKed	unseen	segn	ent]
23	0.106	24.6.173	.22 23.62.	228.65	54 TCP	[TCP	АСКеd	unseen	segn	ent]
26	0.107	24.6.173	.22 23.62.	228.65	54 TCP	[TCP	ACKed	unseen	segn	ent]
31	0.123	24.6.173	.22 23.62.	228.65	54 TCP	[TCP	ACKed	unseen	segr	ent]
34	0.124	24.6.173	.22 23.62.	228.65	54 TCP	[TCP	ACKed	unseen	segn	ent]
36	0.138	24.6.173	. 22 23 . 62 .	228.65	54 TCP	[TCP	ACKed	unseen	segm	ent]
40	0.142	24.6.173	.22 23.62.	228.65	54 TCP	[TCP	ACKed	unseen	segn	ent]
	0 140	- 14 C 173	22,22 62	220 65	E4 TCD	TCD	C K a d			
U UI	ескъчш.	NXCT/C [/	alluation	i utsant	euj					
Ur	gent por	inter: 0								
□ [S	EQ/ACK a	analysis]								
ſ	This is	an ACK to	o the seam	nent in	frame:	14]				
ī	The RTT	to ACK th	le segment	was 0	000188	000 SE	conds]			
ĥ	int o	015571000) cocondol				condoj			
L	IKII. U	.0133/1000	, seconds						_	
	TCP Ana	lysis Flag	js j							
۲	[Expert	Info (Wa	rn/Sequenc	ce): ACK	ed segn	ient tl	nat was	n't ca	pture	d (ca
ΒT	mestam	ne i							-	

Step 3: Click the **Clear** button so that the filter expression is removed.

Topic 161: Wireshark Lab 60

In this topic, we find ACKed Unseen Segment indications using Expert Infos in Wireshark.

Step 1: Open tr-badcapture.pcapng.

Step 2: Click the Expert Infos button on the Status Bar.

Step 3: Click the **Warnings** tab. Expand the section with title **ACKed segment that** wasn't captured (common at capture start).



Step 4: Click on the first entry, **Packet 15** and then click the **Close** button. Let's go to the main Wireshark window. Display the **SEQ#**, **NEXTSEQ#** and **ACK#** columns.

	_															
1	lo.	Time St	Q#	NSEQ#	ACK#	Source		Destination		Length	Protocol	Info				
	11	0.0/1	2010	3970	009	23.02	.220.0	24.0	1/3.220	1014	ILSV	. 191101 EU		OWIT	RECULU	
	12	0.0/1	39/6	5436	809	23.62	.228.6	24.6.	1/3.220	1514	TLSV	Ignored	i Unkn	own	Record	
	13	0.071	809		5436	24.6.	173.22	23.62	.228.65	54	тср	35318→4	143 [A	CK]	Seq=809	
	14	0.087	6896	8356	809	23.62	.228.6	24.6.1	173.220	1514	TLSV	[TCP Pr	reviou	s se	gment n	0
	15	0.087	809		8356	24.6.	173.22	23.62	.228.65	54	ТСР	[TCP AC	CKed u	nsee	n segme	n
	16	0.088	8356	9816	809	23.62	.228.6	24.6.1	173.220	1514	TLSV	Ignored	l Unkn	own	Record	
	17	0.089	11276	12736	809	23.62	.228.6	24.6.1	173.220	1514	TLSV	[TCP Pr	eviou	s se	gment n	0
ĺ	18	0.089	809		12736	24.6.	173.22	23.62	.228.65	54	TCP	[TCP AC	CKed u	nsee	n segme	n
	19	0.104	12736	14196	809	23.62	.228.6	24.6.1	173.220	1514	TLSV	Ignored	l unkn	own	Record	
	20	0.105	14196	15656	809	23.62	.228.6	24.6.1	173.220	1514	TLSV	Ignored	l Unkn	own	Record	
	21	0.105	15656	17116	809	23.62	.228.6	24.6.1	173.220	1514	TLSV	Ignored	l Unkn	own	Record	
	22	0.105	809		17116	24.6.	173.22	23.62	.228.65	54	тср	35318→4	143 [A	CK]	Seq=809	
	23	0.106	809		20036	24.6.	173.22	23.62	.228.65	54	ТСР	[TCP AC	CKed u	nsee	n segme	n
	24	0.106	20036	21496	809	23.62	.228.6	24.6.1	173.220	1514	TLSV	:[TCP Pr	eviou	s se	gment n	0
	25	0.106	21496	22956	809	23.62	.228.6	24.6.1	173.220	1514	TLSV	Ignored	l Unkn	own	Record	
I	26	0.107	809		22956	24.6.	173.22	23.62	.228.65	54	ТСР	[TCP AC	CKed u	nsee	n segme	n
I	27	0.121	22956	24416	809	23.62	.228.6	24.6.1	173.220	1514	TLSV	Ignored	l Unkn	own	Record	
	20	0 400	21110	25070	000	22.02	220.0	24.0		1044		-	L I		- 1	

As you examine the SEQ# values, there are jumps in values. But no Retransmissions, Duplicate ACKs, or Fast Retransmissions can be observed. As far as the TCP peers are concerned, no packets have been lost. The spanned switch is not sending all the traffic down the port to which Wireshark is connected.

Topic 162: Causes of Keep Alives

In this topic, we explain Keep Alives and their causes.

Keep Alives are used refer to TCP Keep Alive Probe packets. These packets are used to detect dead connections, detect dead TCP peers, and prevent a connection from terminating when idle. Keep Alives are implemented as ACK packets that are either empty or contain 1 byte of data. The sequence number value in a Keep-Alive packet is 1 less than the next expected sequence number. TCP SYN, FIN and RST packets cannot be considered to be Keep Alives.

How Wireshark detects Keep-Alives?

Wireshark can easily detect Keep Alives because it keeps a track of the Sequence Number field values in all TCP streams. The Notes column of Wireshark's Expert Info window provides information about Keep Alives/Keep Alive ACKs.

What Causes Keep Alives?

Applications written to use Keep Alives generate Keep Alives. Three parameters are defined for Keep Alives:

1) Keep Alive Time: The amount of time between the last data packet and the first Keep Alive probe.

2) Keep Alive Interval: Interval between Keep Alive Probes.

3) Keep Alive Probes: Number of unacknowledged Keep Alive Probes that should be sent before considering the connection to be dead.

Topic 163: Wireshark Lab 61

This topic uses a filter to count Keep Alive/Keep Alive ACK Packets in Wireshark.

Step 1: Open tr-keepalives.pcapng.

Step 2: Enter the filter **tcp.analysis.keep_alive || tcp.analysis.keep_alive_ack** in the display filter area. Click **Apply**.



We can see that only two packets match the filter.

[Calculated window size: 65273] [Window size scaling factor: 1] @ Checksum: 0x139a [validation disabled] Urgent pointer: 0
<pre>[SEQ/ACK analysis]</pre>
[iRTT: 0.097910000 seconds] [Bytes in flight: 1]
[TCP Analysis Flags]
Expert Info (Note/Sequence): TCP keep-alive segment]
<pre> [T1mescamps] </pre>

Step 3: Display **TCP Delta** column if it is hidden. Otherwise, create it. If you look at the TCP Delta before Packet 61, then you can see that 10.2.122.80 has a 300 second Keep Alive Time value. Packet 62 is the Keep Alive ACK packet.

 U	лесфанксэргарий (инсэнангичгээ (ингэээлдарлэсэал нон назентичгэ)) — — — —
Ele Edit View Go Capture Analyze Statistics Telephony Iools	nternals Help
0 0 <u>/ 1 / 1 X (4 + 4 7 1 1</u> 0 0	🗏 📴 Q, Q, Q, 🕅 👪 🛛 🧏 💥 🙀
Filter: tcp analysis keep_alive tcp analysis keep_alive_ack	▼ Expression Clear Apply Save TCP Dealy HTTP Delay SM8 Delay Bad TCP
No. Time TCD Dalka	Source Destination Length Protocol Info
61 302.094 300.059512000	10.2.122.8010.1.119.172 60 TCP [TCP Keep-A]ive] 5200→64862 [ACH
62 302.094 0.000297000	10.1.119.17.10.2.122.80 78 TCP [TCP Keep-Alive ACK] 64862+5200

Step 4: Remove the filter by clicking the **Clear** button.

Topic 164: Wireshark Lab 62

In this topic, we find Keep Alive/Keep Alive ACK Packets with Expert Infos in Wireshark.

Keep Alives and Keep Alive ACKs are typically used to check for dead TCP peers and avoid time out of idle connections. They are not indications of problems.

Step 1: Open tr-keepalives.pcapng.

Step 2: Click the Expert Infos button on the Status Bar.

Step 3: Click the **Notes** tab. You can see **Keep Alives** and **Keep Alive ACKs**. Expand the **Keep Alive** and **Keep Alive ACK** lines. You can see Packets 61 and 62 listed.

4	Wireshark: 7 Expert Infos	-	□ X
Errors: 0 (0) Warnings: 2 (2)	Notes: 2 (2) Chats: 3 (3) Details: 7 Packet Comments: 0		
Group	Summary	 Count 	•
E Sequence TCP	TCP keep-alive segment		1
Packet:	61		1
🗆 pequence TCP	ACK to a TCP keep-alive segment		1
Packet:	62		1
Limit to display filter			
<u>H</u> elp			<u>C</u> lose

Step 4: Click on the **Keep Alive** entry (Packet 61). This will take you to the main Wireshark window. Click the **Close** button to close the Expert Infos window. Display the **SEQ#**, **NEXTSEQ#** and **ACK#** columns. For Packet 61, you can find that Sequence Number field value has been decremented by 1 from 10,943 to 10,942.

🖌 tr-	keepaliv	es.pcapi	ng			16						<
File	Edit \	liew G	o <u>C</u> apture	Analyze	Statistics Telepho	ny Iools I	nternals <u>H</u>	elp				
0	۰ 🚺			* 2	0 0 0 0	72		Q Q Q		2 🔧 %		
Filt	ter:				In Keep Alive	packets t	he Expr	ession (Clear App	ily Sav	ve TCP	Delay HTTR
No.	TCP De	lta S	ource		SEQ# decre	ments by	1 gth	SEQ#	NEXTSEQ#	ACK#	Info	
58	0.00	01031	0.1.119	.172			8	3538	3660	10472	TDS5	query[Pa
59	0.09	3601	0.2.122	.80	10.1.119.17	2 TDS	11	047Z	10943	3660	Respon	nse[Pack
60	0.03	32661	0.1.119	.172	10.2.122.80	ТСР	66	560	-	10943	64862	> targy
61	300.	0591	10.2.122	.80	10.1.119.17	2 TCP	60	10942	10943	3660	[TCP	Keep-A1
62	0.00	0291	10.1.119	.172	10.2.122.80	TCP	78	3660		10943	[TCP	Keep-Al
63	161.	4221	0.1.119	.172	10.2.122.80	TDS	82	3660	3676	10943	TDS5	query 🎙
64	0.09	1881	10.2.122	.80	10.1.119.17	2 TDS	103	10943	10980	3676	Respon	nse 🖌
65	0.00	0361	0.1.119	.172	10.2.122.80	ТСР	66	3676		10980	64862	> targ
66/		de.		12	20-3- 5 2	T	75	Peres	and the second second	10		man

Topic 165: Wireshark Lab 63

In this topic, we identify Keep Alive Packets used in Zero Window Conditions in Wireshark.

The two sides of a TCP conversation advertise their receive buffer space in the Window Size Value field. It is a 2-byte field whose maximum value can be 65,535 bytes. Over today's high-speed links, if a receiving application is slow in pulling data out of the receive buffer, it's not unusual that the advertised Window Size value can drop to zero – Zero Window Condition. Hosts can send TCP Keep Alives in place of Window Zero Probes. When a TCP host sends a Keep Alive to a peer that is advertising a Zero Window condition. You will not see Keep Alive ACK responses.

Step 1: Open tr-youtubebad.pcapng.

							t	r-youtub	ebad.pca	apng (V	/ireshark 1.1	12.3 (1	v1.12.3-	0-gbb3e9	a0 from m	naster-1.1	2)]		
<u>F</u> ile	Edit	<u>V</u> iew <u>G</u>	o <u>C</u> apture	<u>A</u> nalyze	<u>Statistics</u>	Telephony	Tools	Internals	<u>H</u> elp										
0	0	(• 4		X 🖁	0, ¢	\$ 🖗	F ±		⊕, 6	e, e, e] 🏭 🕅	5	6 🖁						
Fi	lter:							۷	Expression	Clea	r Apply	9	Save	TCP Dealy	HTTP Delay	SMB Delaj	Bad TCP		
No.		Tim	2	TCP D)elta			Source			Destination			Length	Protocol	Info			
	1	0.	.000	0.	00000	00000		24.	4.7.	217	208.1	17.	232	.1 6	6 ТСР	567	70→80	[SYN]	Seq=0
	2	0	.029	0.	02903	3000		208	.117	.232	24.4.	7.2	217	6	6 ТСР	80→	56770	[SYN,	ACK]
	3	0	.029	0.	00010	06000		24.	4.7.	217	208.1	17.	232	.1 54	4 TCP	567	70→80	[ACK]	Seq=1
	4	0	.029	0.	00058	35000		24.	4.7.	217	208.1	17.	232	. 1132	3 HTT	P GET	/vide	eoplayl	back?s
	5	0	.054	0.	02458	30000		208	.117	.232	24.4.	7.2	217	6	O TCP	80→	56770	[ACK]	Seq=1
	6	0.	331	0.	27670	04000		208	.117	.232	24.4.	7.2	217	37	9 htti	P HTT	P/1.1	200 0	K
	7	0	331	0.	00095	57000		208	.117	.232	24.4.	7.2	217	34	6 ТСР	80→	56770	[PSH,	ACK]
	8	0	331	0.	00000)3000		208	.117	.232	24.4.	7.2	217	7	7 тср	80→	56770	[PSH,	ACK]
	9	0	331	0.	00000)3000		208	.117	.232	24.4.	7.2	217	6	3 тср	80→	56770	[PSH,	ACK]
	10	0	331	0.	00000	02000		208	.117	.232	24.4.	7.2	217	18	2 TCP	80→	56770	[PSH,	ACK]
	11	0	331	0.	00000	02000		208	.117	.232	24.4.	7.2	217	6	3 тср	80→	56770	[PSH,	ACK]
	12	0	331	0.	00000	02000		208	.117	.232	24.4.	7.2	217	18	2 ТСР	80→	56770	[PSH,	ACK]

Step 2: Click the Expert Infos button on the Status Bar and then click the Notes tab.



During a successful keep alive process, hosts exchange Keep Alives and Keep Alive ACKs. In this case, either the peer is dead or this is not a standard keep alive process.

Step 3: If you expand the **Keep-Alive** section, you will find **Packet 2,721** the first Keep Alive listed. Click on it and then click the **Close** button to return to the main Wireshark window.

4		Wireshark	: 162 Exper	t Infos	-	
Errors: 0 (0) Warning	s: 2 (50) Notes: 1 (3	2) Chats: 6 (80)	Details: 162	Packet Comments: 0		
Group	ol Summa	ry			 Count 	4 ^
Packet:	2721	ep-alive segment				32 1
Packet:	2723					1
Packet:	4024					1
Packet:	4026					1
Packet:	4028					1
Packet:	4030					1
Packet:	4032					1 🗸
Limit to display filt	er					
<u>H</u> elp						<u>C</u> lose

<u>File Edit 1</u>	(iew <u>G</u> o <u>C</u> apture	e <u>Analyze</u> <u>Statistics</u> Telephony <u>T</u> ools	ls Internals Help
001		1 X 🔁 9, 4 4 4 🕯 🕇 🗄	: 📃 🖪 e. q. q. 🗉 📓 🛛 📕 % 🔛
Filter:			Expression Clear Apply Save TCP Dealy HTTP Delay SMB Delay Bad TCP
No.	Time	TCP Delta	Source Destination Length Protocol Info ∠V0.11/.22/24.4./.21/ 1214 127 00→30//0 1ACK 1.2
2716	2.485	0.000004000	208.117.232 24.4.7.217 1514 TCP 80→56770 [ACK] S
2717	2.485	0.00003000	208.117.232 24.4.7.217 1514 TCP 80-56770 [ACK] S
2718	2.485	0.000097000	24.4.7.217 208.117.232.1 54 TCP 56770+80 [ACK] 5
2719	2.486	0.000769000	208.117.232 24.4.7.217 1514 TCP [TCP Window Fu]]
2720	2.688	0.202321000	24.4.7.217 208.117.232.1 54 TCP [TCP ZeroWindow]
2721	3.095	0.407261000	208.117.232 24.4.7.217 60 TCP [TCP Keep-Alive]
2722	3.095	0.000049000	24.4.7.217 208.117.232.1 54 TCP [TCP ZeroWindow]
2723	3.882	0.786594000	208.117.232 24.4.7.217 60 TCP [TCP Keep-Alive]
2724	3.882	0.000048000	24.4.7.217 208.117.232.1 54 TCP TCP ZeroWindow
2725	4.094	0.211823000	24.4.7.217 208.117.232.1 54 TCP [TCP Window Upda
2/26	4.114	0.020629000	208.117.232.24.4.7.217 1514 TCP 80+56/70 [ACK] 5
101	4 115	0 000x39000	/IX II/ /3//4 4 / /I/ ISI4 TCP XII+S6//I LACKI S

You can see that the first Keep Alive occurs almost 400 ms after a packet from the peer. It is not used to check for a dead peer. It is used to determine if Window Update (Window Size value has increased) has taken place.

Topic 166: Causes of Reused Ports

In this topic, we explain reused ports and their causes.

Sometimes Reused Ports are not a problem. Other times Reused Ports can cause significant delays in communications. Mostly, we can reuse port numbers without any problem as long as the previous TCP connection is terminated. Let's assume a client establishes a TCP connection with a server using source port number 1026 and destination port number 80. After a while, without terminating the previous connection, it tries to establish another new connection with same port number. In this case, if the first connection had been terminated, then reused ports would have not been a problem. Reused Ports become issue, when previous connection has not terminated (through TCP FINs/RSTs).

When Wireshark detects a previous SYN packet using the same IP address/port number combination in a trace file, then it marks SYN packets with the Reused Ports Expert Analysis definition.

What Causes Reused Ports?

Reused ports can be seen if

- A) An application defines a static source port number, or
- B) An application uses a very small range of source port numbers.
- C) A host is going through a lot of source port numbers very quickly.

Topic 167: Wireshark Lab 74

This topic uses a filter to count reused port Packets in Wireshark.

Step 1: Open tr-reusedports.pcapng.

Step 2: Type the filter **tcp.analysis.reused_ports** in the display filter area. Click **Apply**. In this trace file, Wireshark has detected one Reused Port as indicated by Status Bar.



Step 3: Expand the **[SEQ/ACK analysis]** section of the resued port packet whose number is 317. This area is colored cyan. Wireshark associates cyan with Expert Infos **Notes**.

Filter	tcp.analysis.reused_ports			 Expression 	Clear	Apply	Save	TCP Dealy	HTTP Delay SMB (Delay Bad TCP			
No.	Time	Source	Destination		Length	Protocol	Info			17	50010 0100	[augu]	
31	/ 43.060	192.168.	1.4.192.1	68.1.10	1 66	тср	LTCP	Port	numbers	reused	59319+9100	[SYN]	Seq=
<													
5	Acream mac	· · · · · · · · · · · · · · · · · · ·											
L	ICP Segment	Len: Uj	(. 1										
S	equence num	ber: 0	(relativ	e seque	ence	numb	er)						
A	cknow ledgme	nt number	•: 0										
H	eader Lengt	h: 32 byt	es										
. ⊛	0000 00	00 0010 =	: Flags: ()x002 (s	SYN)								
W	indow size	value: 61	.44										
[Calculated	window si	ze: 6144]										
B C	necksum: Ox	4d8c [va]	idation o	lisableo	1]								
U	rgent point	er: 0											
⊞ 0	otions: (12	bytes),	Maximum s	egment	size	, NO	-Oper	ation	(NOP),	Window s	cale, No-Op	eration	1 (NO
8	SEQ/ACK ana	lysis]		, in the second se									
	[iRTT: 0.00	1683000 s	seconds]										
Ξ	[TCP Analys	is Flags											
	Expert Ir	fo (Note,	/Sequence): A nei	w tcp	ses	sion	is st	tarted w	ith the s	ame ports a	s an e	arlie
- [I mor tamor I										_	_	

Step 4: Let's remove filter by clicking the **Clear** button.

Topic 168: Wireshark Lab 75

This topic finds Reused Ports with Expert Infos in Wireshark.

Step 1: Open tr-reusedports.pcapng.

Step 2: Click the Expert Infos button on the Status Bar.

Step 3: Click the **Notes** tab. Expand the **A new tcp session is started with the same ports as an earlier session in this trace** section. Click on Packet **317**. Close the Expert Infos window by clicking **Close**. Wireshark will take you to main Wireshark window.

Wireshark: 54 Expert Infos	-	
Errors: 0 (0) Warnings: 3 (17) Notes: 3 (17) Chats: 4 (20) Details: 54 Packet Comments: 0		
Group	 Count 	•
Sequence TCP TCP Zero Window Probe		8
Sequence TCP ACK to a TCP Zero Window Probe		8
Sequence TCP A new tcp session is started with the same ports as an earlier session in the same ports as as an earlier sess	this	
Packet: 317		1
Limit to display filter		
Help		<u>C</u> lose

Step 4: As expected, the Reused Ports packet shown is a SYN packet. This packet uses IP address 192.168.1.44 and port 59319.

_																
<u>File</u> <u>E</u> dit	View	<u>G</u> o <u>C</u> apture	<u>A</u> nalyze	Statistics	Telephony	<u>I</u> ools <u>I</u> nter	nals <u>H</u> elp									
00/		<u> B </u>) X 🛿	🔍 🝦	🕈 🌒 🖥	2 I 🗉	🖬 Q, (ର୍ଷ୍] 🏭 🕅	8 %	X					
Filter:							 Expression 	Clear	Apply	Save	TCP Dea	y HTTP Del	ay SMB Delay	Bad TCP		
No.	Tin J	ne 9.005	Source	. 100.	1.77	Destination	100.1	. 101	Length P	rotocol	Info	-9100		JCy-		ACK-1
314	3	9.684	192	168.	1.44	192.	168.1	.101	1514	ГСР	59319	→9100	[ACK]	Seq=	36291	7 Ack=1
315	3	9.685	192	.168.	1.101	192.	168.1	.44	60 -	ГСР	9100-	59319	[ACK]	Seq=	1 Ack	=364377
_316	4	3 058	192	168	1 101	192	168 1	44	60 1	ГСР	<u>9100-</u>	59319	ETN,	PSH	ACK]	Seq=1 /
317	4	3.060	192	.168.	1.44	192.	168.1	.101	66 1	ГСР	[TCP	Port	number	s reu	sed]	59319→91
318	4	3.062	192	.168.	1.101	192.	168.1	.44	66 1	ГСР	9100→	59319	[RST,	ACK]	Seq=	1 Ack=0
<																
Tot		. II.,	510	. mi 1 vo		_JC.a	10.07	2 160	1 44	(10	1.07	1 44		. 100	160	1 101 (1
• TUC	erne			i ve	rs ton	4, 51	C: 19	2.100	50210	(19	2.100	.1.44), DSC	192.	100	L.IUI (]
	ISI	SSTO	Con		Prote	, 100	SEC P	ort:	28218	(39	519),	DST	PORT:	aton (, ATON), Seq:
50	urce	e Port	:: 59	319	(5931	<i>i</i>)										
De	stil	nation	1 Por	t: 9	100 (9	9100)										
[S	trea	am ind	lex:	1]												
[т	CP S	Segmer	nt Le	n: 0]											
Se	quei	nce nu	umber	: 0	(re	elativ	/e seq	uence	e numb	er)						
AC	know	vledgr	nent	numb	er: 0						-					
Не	adeı	r Lend	th:	32 b	vtes											
		_	,		1.20											

Step 5: The sequence number field can be used to determine whether the SYN belongs to a unique connection request, a retransmission or a reused port. Wireshark applies a relative sequence number to TCP. In the Packet Details pane of any packet, right-click on the **TCP header.** Select **Protocol Preferences** and toggle *Relative sequence numbers* off.

Step 6: To view the sequence number value of SYN packets. Enter **tcp.flags==0x0002** in the display filter area and click **Apply**. You will find only 2 packets.

Step 7: Display your **SEQ#** column to view the sequence numbers quickly. The sequence numbers are different. Packet 317 would be a retransmission if they were the same.



Step 8: To turn the *Relative sequence numbers* preference setting on, click on a **TCP header** in the Packet Details pane, select **Protocol Preferences**.

Topic 169: Causes of Checksum Errors

This topic explains checksum errors and their causes.

A checksum is used to detect errors (e.g., flipped bits) in a transmitted segment. In TCP/IP model, UDP, TCP, IPv4 and link-layer protocols check against errors using a checksum. By default, Wireshark does not perform TCP, UDP or IPv4 checksum validation. Wireshark's Checksum validation processes can be enabled/disabled easily using the Preference settings. To enable checksum validation for IPv4, TCP and UDP, select the **Preferences** button on the Main Toolbar. Expand the Protocols section and select **IPv4**. Check the **Validate the IPv4 checksum if possible** preference setting.

	Wireshark: Preferences - Profile: Default
ATTR2 ^	N-L BATTERIA - NEL FAL
x	Celate Pik (Careo & Miserinec
œ	Ressemble fragmented Pv4 datagrams 🗹
CMP	Show Pull summary in protocol tree 🛛
EEE 802.11	
RF 802.154	Validate the PA4 checksium if possible
2EE 802.14H	Support packet-cepture from PTSG-enabled hardwares 🛛 🕄
ACP	
U I	Enable Geoli I bokups 💜
MF	Interpret Reserved Flag as Security Flag (RFC 3514):
NRP	Technistic schrösseten fint
nfinBand	diama management []
nfiniband SDP	
PDC .	
214	
h4	
246	
sune	

Causes of Checksum Errors

A faulty Network Interface Card (NIC) or any device that alters the content of the packets along the path. On a capturing device, task Offloading. By task offload, we mean that numerous processes are offloaded to a network interface card to free up a host's CPU for other tasks. Assume you are capturing your own traffic to and from your machine on which Wireshark is running. You would see Checksum Errors on all outbound traffic if

- (a) Task offloading is enabled on that host, and
- (b) Wireshark checksum validation processes are enabled.

Topic 170: Wireshark Lab 76

This topic detects checksum errors with Expert infos in Wireshark.

Step 1: Open tr-checksums.pcapng.

Step 2: To enable checksum validation for IPv4, TCP and UDP, select the **Preferences** button on the Main Toolbar.

Step 3: Expand the Protocols section and select **IPv4**. Check the **Validate the IPv4** *checksum if possible* preference setting.

Step 4: Repeat step 3 for TCP.

Step 5: Repeat step 3 for **UDP**. Click **OK** to save these preference settings. You will see that the coloring in the Packet List pane changes. Packets from 192.168.1.72 have a black background and red foreground.

Step 6: Click the Expert Infos button on the Status Bar.

Step 7: Go to **Errors** tab. You can see 68 bad IPv4 checksums, 6 bad UDP checksums and 62 bad TCP checksums. Expand the **TCP Bad Checksum** section.

4			Wireshark:	163 Expert	Infos	-	×
Errors: 3 (136)	Warnings: 1 (1)	Notes: 2 (3)	Chats: 13 (23)	Details: 163	Packet Comments: 0		
Group 4	Protocol	 Summary 				 Count 	4
E Checksum	IPv4	Bad checks	um				68
E Checksum	UDP	Bad checks	um				6
⊟ Checksum	ТСР	Bad checks	um				62
Packet:		5					1
Packet:		7					1
Packet:		8					1
Packet:	1	10					1
Packet:	1	15					1 🗸
Limit to dis	play filter						<u>C</u> lose

Step 8: Packet 5 is the first TCP Bad Checksum entry. Click on it. This will take you to the main Wireshark window. You can see that the packets from 192.168.1.72 when arrived at the target do not contain error. The target acknowledged receipt of the packets, processed them, and replied to the requests. Host with 192.168.1.72 is configured to use task offloading.

Step 9: Disable, IPv4, UDP and TCP checksum validation settings.

Topic 171: Wireshark Lab 77

This topic uses DNS errors Filter in Wireshark.

Two most common DNS errors are 1) a server failure (Reply Code 2), and 2) a name error, listed as Non-Existent Domain, (Reply Code 3).

To quickly identify DNS errors in a trace file, let's create a button.

Step 1: Open tr-dnserrors.pcapng.

Step 2: Packet 5 is the first DNS response. Click on it. From the Info column, we can see that it is a Server Failure reply.

Step 3: In **Packet 5**, right-click on the **Domain Name System (response)** line and select **Expand Subtrees**. You will find the DNS Reply Code field inside the Flags section.

Step 4: Right-click on the Reply code field and select Prepare a Filter | Selected.

Filter Expression Clear Expand Subtrees Expand Subtrees No. Time Source Destination Length Expand Subtrees Collapse Subtrees Collapse Subtrees 1 0.000 192.168.0.113 192.168.0.1 7 Apply as Column Oxa 570 A 3 1.999 192.168.0.113 192.168.0.1 7 Apply as Column Oxa 570 A 4 3.999 192.168.0.113 192.168.0.1 7 Apply as Column Oxa 570 A 6 5.095 192.168.0.11 192.168.0.1 7 Follow TCP Stream	00	(🔳 🔬 🖻 🛙) 🗶 🍠 🔍 🔶 🏟 🥥 👍	👱 🗏 📑 Đ, Q, Q,	T 🖌	M 🔜 💥 🔀	_		
No. Time Source Destination Length 1 0.000 192.168.0.113 192.168.0.1 7 2 0.999 192.168.0.113 192.168.0.1 7 3 1.999 192.168.0.113 192.168.0.1 7 4 3.999 192.168.0.113 192.168.0.1 7 4 3.999 192.168.0.1 192.168.0.1 7 6 5.095 192.168.0.1 192.168.0.113 7 6 5.095 68.87.76.183 192.168.0.113 7 7 6.047 192.168.0.1 192.168.0.113 7 8 6.047 68.87.78.136 192.168.0.113 7 9 21.825 192.168.0.113 192.168.0.1 7 9 21.825 192.168.0.113 192.168.0.1 7 9 21.825 192.168.0.113 192.168.0.1 7 9 21.825 192.168.0.113 192.168.0.1 7 9 1.02.168.0.1 102.168.0.1 7 Protocol Page 10.11.10 = Recursion available: </th <th>Filter:</th> <th></th> <th></th> <th>✓ Expression Cle</th> <th>ar Ap</th> <th>Expand Subtrees Collapse Subtrees</th> <th>ela</th> <th>y Bad TCP</th> <th>HTTP Error</th>	Filter:			✓ Expression Cle	ar Ap	Expand Subtrees Collapse Subtrees	ela	y Bad TCP	HTTP Error
1 0.000 192.168.0.113 192.168.0.1 7 Collapse All 0xa570 A 2 0.999 192.168.0.113 192.168.0.1 7 Apply as Column 0xa570 A 3 1.999 192.168.0.113 192.168.0.1 7 Apply as Column 0xa570 A 4 3.999 192.168.0.113 192.168.0.1 7 Prepare a Filter Selected Not Selected 5 5.095 192.168.0.1 192.168.0.113 7 Follow UP Stream and Selected a	No.	Time	Source	Destination	Length	Expand All			
2 0.999 192.168.0.113 192.168.0.1 7 3 1.999 192.168.0.113 192.168.0.1 7 4 3.999 192.168.0.113 192.168.0.1 7 5 5.095 192.168.0.1 192.168.0.113 7 6 5.095 68.87.76.183 192.168.0.113 7 7 6.047 192.168.0.1 192.168.0.113 7 8 6.047 68.87.78.136 192.168.0.113 7 9 21.825 192.168.0.113 192.168.0.1 7 9 21.825 192.168.0.113 192.168.0.1 7 9 21.825 192.168.0.113 192.168.0.1 7 9 21.825 192.168.0.113 192.168.0.1 7 9 21.825 192.168.0.113 192.168.0.1 7 9 1.02.168.0.1 102.168.0.113 192.168.0.1 7 9 1.02.168.0.1 102.168.0.1 7 Epot Selected Protocol Page 1 = Recursion available: Protocol Preferences Protoco	1	0.000	192.168.0.113	192.168.0.1	7	Collapse All	1	0xa570	Α
3 1.999 192.168.0.113 192.168.0.1 7 4 3.999 192.168.0.113 192.168.0.1 7 5 5.095 192.168.0.1 192.168.0.113 7 6 5.095 68.87.76.183 192.168.0.113 7 7 6.047 192.168.0.1 192.168.0.113 7 8 6.047 68.87.78.136 192.168.0.113 7 9 21.825 192.168.0.113 192.168.0.1 7 9 21.825 192.168.0.113 192.168.0.1 7 9 21.825 192.168.0.113 192.168.0.1 7 9 21.825 192.168.0.113 192.168.0.1 7 10 21.827 102.168.0.113 192.168.0.1 7 1 = Recursion desired: D Protocol Page Filter Field Reference 1 = Recursion available: Disable Protocol. Beolow Name = Recursion available: Disable Protocol. Beolow Name = Answer aut	2	0.999	192.168.0.113	192.168.0.1	7	Apply as Column	1	0xa570	Α
4 3.999 192.168.0.113 192.168.0.1 7 5 5.095 192.168.0.1 192.168.0.113 7 6 5.095 68.87.76.183 192.168.0.113 7 7 6.047 192.168.0.1 192.168.0.113 7 8 6.047 68.87.78.136 192.168.0.113 7 9 21.825 192.168.0.113 192.168.0.1 7 9 21.825 192.168.0.113 192.168.0.1 7 10 21.827 192.168.0.113 192.168.0.1 7 9 21.825 192.168.0.113 192.168.0.1 7 10 21.827 192.168.0.1 102.168.0.1 7 1 = Recursion desired: D Protocol Page Filter Field Reference 1 = Recursion available: Protocol Page Filter Field Reference 0 = Answer authenticated Bolow Non Bolow Non 0 Non-authenticated Sorver Sorver 0 Non-authenticated <td>3</td> <td>1.999</td> <td>192.168.0.113</td> <td>192.168.0.1</td> <td>7</td> <td>Annly as Filter</td> <td>•</td> <td>0xa570</td> <td>Δ</td>	3	1.999	192.168.0.113	192.168.0.1	7	Annly as Filter	•	0xa570	Δ
5 5.095 192.168.0.1 192.168.0.113 7 6 5.095 68.87.76.183 192.168.0.113 7 7 6.047 192.168.0.1 192.168.0.113 7 8 6.047 68.87.78.136 192.168.0.113 7 9 21.825 192.168.0.113 192.168.0.1 7 10 21.825 192.168.0.113 192.168.0.1 7 10 21.825 192.168.0.113 192.168.0.1 7 1 = Recursion desired: D Potocol Page Filter Field Reference 1 = Recursion available: Potocol Help Potocol Help 0 = Xnop-authenticated Bisble Protocol Bisble Protocol Bisble Protocol 0 = Nop-authenticated Got Corresponing Packet Show Packet Reference in New Window	4	3,999	192.168.0.113	192.168.0.1	7	Prepare a Filter	<u>۲</u>	Selected	
6 5.095 68.87.76.183 192.168.0.113 7 7 6.047 192.168.0.1 192.168.0.113 7 8 6.047 68.87.78.136 192.168.0.113 7 9 21.825 192.168.0.113 192.168.0.1 7 10 21.825 192.168.0.113 192.168.0.1 7 10 21.825 192.168.0.1 102.168.0.1 7 10 21.825 192.168.0.1 7 6.047 10 21.827 102.168.0.1 7 6.047 1 = Recursion desired: D Pitter Field Reference Protocol Page 0 = Recursion available: Pitter Field Reference Protocol Page 0 = Recursion available: Disable Protocol. Bisole Name 0 = Answer authenticated Bisole Name Ortion Was 0 = Non-authenticated Sorver for the Window Ortion Was	5	5 095	192 168 0 1	192 168 0 113	7	Follow TCP Stream	`	and Selected	н
7 6.047 192.168.0.1 192.168.0.113 7 8 6.047 68.87.78.136 192.168.0.113 7 9 21.825 192.168.0.113 192.168.0.1 7 10 21.827 192.168.0.1 192.168.0.1 7 1 = Recursion desired: D Protocol Page Filter Field Reference 1 = Recursion available: Protocol Page Filter Field Reference = Recursion available: Bible Protocol Page Filter Field Reference = Recursion available: Bible Protocol Page Filter Field Reference = Recursion available: Bible Protocol Page Sive querie = Recursion available: Bible Protocol Page Sive querie	6	5 095	68 87 76 183	192 168 0 113	7	Follow UDP Stream		<u>o</u> r Selected	
7 0.047 192.100.011 192.100.0113	7	6 047	102 168 0 1	102 168 0 113	7	Follow SSL Stream		a <u>n</u> d not Sele	ected
8 6.047 08.87.78.130 192.108.0.113 7 9 21.825 192.168.0.113 192.168.0.1 7 10 21.827 102.168.0.1 102.168.0.1 7 1	/	0.047				Сору	•	or not select	ted
9 21.825 192.168.0.113 192.168.0.1 7 Edit Packet 0x8b73 A 10 21.825 192.168.0.1 102.168.0.113 0 Wiki Protocol Page 0x8b73 A	ð	0.047	08.8/./8.130	192.108.0.113) /	Export Selected Packet Bytes		respon	se u
10 21 837 102 168 0 113 0 Wiki Protocol Page	9	21.825	192.168.0.113	192.168.0.1	7	<u>E</u> dit Packet		0x8b73	A
<pre> Control Con</pre>	10	21 827	102 168 0 1	102 168 0 113	<u> </u>	Wiki Protocol Page	1	rachan	<u>م</u> ۸
	<					Filter Field Reference			
1 = Recursion available: 0 = Z: reserved (0) 0 = Answer authenticated 0 = Non-authenticated da 0 = Non-authenticated da 0 = Non-authenticated da 0 = Non-authenticated da 0 = Non-authenticated da		1	= Re	ecursion desire	ed: D	Protocol Help			
			1 – Pe	cursion avail	ahle.	Protocol Preferences	' .c	ive du	erie
	•		· · · · · · · · · · · · · · · · · · ·		ibic.	දී Decode As	3	ive qu	crite
	•	• • • • • • • •	.0 = 2:	reserved (0)		Disable Protocol			
			0 = An	nswer authenti	cated	<u>R</u> esolve Name)0	rtion	was
0010 - Reply code: Server f Show Packet Reference in New Window			0 = No	n-authenticate	ed da	Go to Corresponding Packet			
			0010 - Pe	anly code: ter	ver f	Show Packet Reference in New Window			

Using this filter will locate Server Failures. As we want to create a button to display DNS errors. All such packets will have values larger than 0 in the **dns.flags.rcode** field.

Step 5: Type **dns.flags.rcode > 0** and click the **Save** button. Rename the button's name to **DNS Errors** and click **OK**.

Step 6: Click the new **DNS Errors** button. You will find that there are eight DNS error responses in this trace file.

Step 7: Expand the **Queries** section in Packet Details pane of a response. Right click on the **Name** field and **Apply as Column**. This will determine what name(s) generated these responses

	5.0 4	Minus C	- Carta	Analas	Charlie		Expand Subtrees		[Wire	shark 1.1	2.3 (v1.
©	©						Expand All Collapse All		2 🖭	¥	10 %
F	ilter:	dns.flags	.rcode > 0				Apply as Column		Clear	Apply	Sa
No.	5	Tim 5	.095	Source 192	. 16		Apply as Filter Prepare a Filter	+ +	13	Length 72	Protocol DNS
	6	5.	.095	68.8	87.		Colorize with Filter	+	13	72	DNS
	7	6	.047	192	.16		Follow TCP Stream		13	72	DNS
	8	6	047	68	87.		Follow SSL Stream		13	72	DNS
	17	40	026	192	16		Сору	•	13	76	DNS
	1.8		026	68	27		Export Selected Packet Bytes		13	76	DNIS
	21			102	16		<u>E</u> dit Packet		12	70	DNC
	21	50	1.061	. 192	. 10	٩	Wiki Protocol Page		13	76	DNS
	22	50	0.061	. 68.8	37.	۲	Filter Field Reference		L3	/6	DNS
							Protocol Help		L		
4							Protocol Preferences	•	L		
<	au		viiai		v	S:	Decode As				
d	011	erie	c)			•	Disable Protocol		L		
	Qu		5		+		<u>R</u> esolve Name		L		
	• W	ww.n	map.	org:	τy		Go to Corresponding Packet				
		Name	e: vw	w.nma	ip.		Show Packet Reference in New Window	_			
		INan	ie Le	nath	1	21					
		F1 ~ L		~+	_	٦ ٦					

_		t	r-dnserrors.pcapng [Wireshark
<u>F</u> ile <u>E</u> dit <u>V</u> ie	w <u>G</u> o <u>C</u> apture <u>A</u> nalyze <u>S</u> tatistics Tele	phon <u>y T</u> ools	<u>I</u> nternals <u>H</u> elp
•• •	📕 🔏 🖻 🗎 🗙 🔁 🔍 🍬 🔶	🥪 ዥ 👱	🗏 📑 I 🗨 Q 🔍 🖭 I 👹
Filter: dns	s.flags.rcode > 0		✓ Expression Clear A
No.	Name	Time	Source
5	www.nmap.org	5.095	192.168.0.1
6	www.nmap.org	5.095	68.87.76.183
7	www.nmap.org	6.047	192.168.0.1
8	www.nmap.org	6.047	68.87.78.136
17	www.insecure.org	49.026	192.168.0.1
18	www.insecure.org	49.026	68.87.76.184
21	www.insecure.org	50.061	192.168.0.1
22	www.insecure.org	50.061	68.87.78.136

The errors can be because of a server upstream from our local server is not responding to recursive DNS queries.

Step 8: Click the Clear button to remove filter.

Topic 172: Wireshark Lab 78

This topic uses an HTTP errors filter in Wireshark.

We will create a button, which when clicked will quickly identify HTTP errors in your trace files.

Step 1: Open tr-chappellu.pcapng.

Step 2: Type **http.response.code >= 400** in the display filter area and click the **Save** button. Let's name the button **HTTP Errors** and click **OK**.

1					tr-chapp	ellu.pcapng	[Wireshark 1.12.	3 (v1.12 .3	3-0-gbb3e9a0 fr	om
	<u>File</u> <u>E</u> dit	<u>V</u> iew <u>G</u> o <u>C</u> apture	<u>Analyze</u> <u>Statistics</u>	Telephony	<u>T</u> ools <u>I</u> nternals	<u>H</u> elp				
	00	(🛛 🔏 🕒	X 2 9, 4	• 🔿 🖏 1	7 1 3	$ \oplus $	Q 🖭 🌉 🖬 🥊	8 % I	ġ	
	Filter:	http.response.code >	·= 400		✓ E	xpression	Clear Apply	Save	TCP Dealy HTT	P De
١	No.	Time	Source		Destination		Length Pr	otocol Ir	nfo	
	1	0.000	24.6.17	3.220	198.66	5.239.1	L46 66 T	СР 3	35621→80)
	2	0.251	24.6.17	3.220	Wires	hark: Save Fil	ter – 🗆	x	Standard	
	3	1.252	24.6.17	3.220	oave Filter as				Standard	(
	4	1.253	24.6.173	3.220	http.response.co	de >= 400 H	TTP Errors		Standard	0
	5	2.252	24.6.173	3.220	Lista		01		Standard	(
	6	2.252	24.6.173	3.220	Teib				Standard	
	7	3.007	24.6.17	3.220	198.66	5.239.1	L46 66 1	СР	[TCP Ret	:ra
I	8	4.252	24.6.17	3.220	75.75.	75.75	77 c	DNS S	Standard	(
	-									

Step 3: Click the newly created **HTTP Errors** button. This will identify the two HTTP error responses in this trace file. The codes are 404 "Not Found" errors.

<u>F</u> ile <u>E</u> dit	<u>V</u> iew <u>G</u> o <u>C</u> ap	ture <u>A</u> nalyze	Statistics Telephor	i <u>y T</u> ools <u>I</u> nterna	s <u>H</u> elp						
00	(🔳 🔬 E	🗋 🗙 🛃	🔍 🌾 🏟 🤪	₮ 🕹 🗉 🛛	∎ ⊕, Q, 0	R 🖭 🌌 🕅	8 %	ġ			
Filter:	http.response.co	de >= 400		¥	Expression	Clear Apply	Save	TCP Dealy H	ITTP Delay SM	B Delay Bad TCP	HTTP Errors
No.	Time	Source		Destination		Length	Protocol Ir	ifo			
61	8.700) 141.	.101.125.	1924.6.	173.220) 1514	HTTF	HTTP/1.	1 404	Not Fou	und [Un
104	10.61	L7 141.	101.125.	1924.6.	173.220) 1514	HTTE	TTP/1.	1 404	Not Fou	ind [Un

Step 4: Right-click on **Packet 61** and select **Follow TCP Stream**. This will allow us to figure out what item was not found on the server. **Follow TCP Stream** applies **a** filter based on the Stream Index number. Also, a window is opened that depicts the conversation without headers.

The 64	New Co. Control	Laska Ostatio Takaban	tr-chappelle Tools Interacts H	upcapng [Wireshark 1.12.3 (vl.12.3-0-gbb3e9a0 from master-1.12]]		
0.0					_	
				🧉 Follow TCP Stream (tcp.stream eq 5) – 🔍	×	
nus	streamed 1	P	•	Streen Context		
51	8.655	24 6 173 220	141.10	GET /legacy/graphics/promo/reader_2_728x90.png	1	1
52	8.675	141, 101, 125, 1	924.6.1	HIIP/1.1		_
54	8 676	24 6 173 220	141 10	Host: www.nitropdf.com	a a	0
56	8 676	24 6 173 220	141 10	User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64;	5	8
60	8 695	141 101 125 1	924 6 1	rv:16.0) Geck0/20100101 FireTox/16.0	5	0
61	8 700	141 101 125 1	974 6 1	Accept: Image/prg,Image/~;q=0.8,*/*;q=0.5		P
62	8 700	141 101 125 1	924 6 1	Accept-Language. en-us,en,g=0.5	50	0
63	8 704	24 6 173 220	141 10	Connection: keen-alive	3	0
64	8 705	141 101 125 1	924 6 1	Referer: http://www.pdfdownload.org/pdf2html/	5	0
65	8 705	141 101 125 1	0 24 6 1	top.php	5	n
			ALCONCEAN ADMIN	Cookie:		
s Fra	ame 61: 15	514 bytes on wi	re (121	cfduid=dc5addb72352aff81134d2a6cd8246b611351116		0
8 Eth	ernet II,	Src: Cadant_3	1:bb:c1	340	a	a7
- Int	ernet Pro	otocol Version	4, Src:		2	24
Tra	ansmission	Control Proto	col, sr	HTTP/1.1 404 NOT FOUND	.6	67
Hyp	ertext Tr	ansfer Protoco	1	Server: CloudTlare-ng1nx	-	

Step 5: Click **Clear** to remove the filter.

Figures and Material used for Part3 (from Topics 173 to 192) have been adapted from A. Jesin's "Packet Tracer Network Simulator", 2014.

Topic 173: Introduction to Packet Tracer

This topic provides an introduction to Packet Tracer.

Packet Tracer is a network simulator from Cisco. It allows you to get a taste of everything in Cisco without needing to purchase real hardware. With Packet Tracer, complex topologies with tens of Cisco devices can be designed and troubleshot. You can also watch the packets moving between them. A wide range of Cisco switches and routers, several end devices such as PCs and servers and wireless devices from Linksys, are available in Packet Tracer.

Layer	Cisco Packet Tracer Supported Protocols				
Application	 FTP, SMTP, POP3, HTTP, TFTP, Telnet, SSH, DNS, DHCP, NTP, SNMP, AAA, ISR VOIP, SCCP config and calls ISR command support, Call Manager Express 				
Transport	 TCP and UDP, TCP Nagle Algorithm & IP Fragmentation, RTP 				
Network	 BGP, IPv4, ICMP, ARP, IPv6, ICMPv6, IPSec, RIPv1/ v2/ng, Multi-Area OSPF, EIGRP, Static Routing, Route Redistribution, Multilayer Switching, L3 QoS, NAT, CBAL , Zone-based policy firewall and Intrusion Protection System on the ISR, GRE VPN, IPSec VPN 				
Network Access/ Interface	Ethernet (802.3), 802.11, HDLC, Frame Relay, PPP, PPPoE, STP, RSTP, VTP, DTP, CDP, 802.1q, PAgP, L2 QoS, SLARP, Simple WEP, WPA, EAP				

Packet Tracer Workspaces

Two workspaces are supported; logical and physical. In the logical workspace, we can build logical network topologies by placing, connecting, and clustering virtual network devices. The physical workspace provides a sense of scale and placement in how network devices such as routers, switches, and hosts would look in a real environment

Packet Tracer Modes

It supports two operating modes—real-time mode and simulation mode. In real-time mode, all network activities take place with immediate real-time response. The simulation mode allows a user to control time intervals, and the propagation of data across a network.

Installation of Packet Tracer

You can download the latest version of Packet Tracer for free from www.netacad.com/about-networking-academy/packet-tracer.

Topic 174: Packet Tracer's Interface Overview

This topic has a look at the Interface of Packet tracer.

The layout of Packet Tracer can be divided into several components.

Menu bar: This is a commonly found menu. It is used to open, close, print, save, change preferences, and so on.

File Edit Options View 1	Tools Extensions Help			
🗋 💳 🖬 🗁 🗊 🕯	🗊 🕑 🕑 💊 🍾 🍗 🖿 🕿	5		1) ?
Logical	[Root] New Cluster	Move Object	Set Tiled Background	Viewport
				â
4				
Time: 00:00:33 Power	Cycle Devices Fast Forward Time			Realtime
Routers	1841 1941 2620XM 2621XM 2811	Scenario 0 New Delete	Fire Last Status S	ource Destination Ty
🛃 🖉 👄 💐 👄	A Device to Drag and Drop to the Work	Toggle PDU List Window	<	>

Main toolbar: This bar contains shortcut icons to menu options that are frequently accessed. For example, open, save, zoom, undo, and redo.

File Edit Options V	iew Tools Extensions	Нер	~		
	<u>1000</u>	ククク 回 る	5		(1) 7
Logical	[Root]	New Cluster	Nove Object	Set Tiled Background	Viewport
					, <mark>, , , , , , , , , , , , , , , , , , </mark>
< Time: 00:00:33 F	Power Cycle Devices	Fast Forward Time			Realtime
2 - H C	4 ma ma		3 Scenario 0 v	Fire Last Status So	surce Destination Ty
Routers	3541 2941	262000 262100 2811	New Delete		
2 0 2	a Device to D	rag and Drop to the Work	Toggle PDU List Window	<	

Physical/Logical workspace tabs: These tabs allow you to choose between the **Logical** and **Physical** work areas.

File Edit	Options	View Tools	Extensions	Help								
De	8		6 A .	00	۵ 📖 🔍							1 ?
Logic	al	(Ro	ot]	Ne	w Cluster	Nove Obje	at)	Set	Tiled Backgro	und	Viewpor	t.
	9											🧾 🐎 🗆 🗙 🛁 👘
Time: 00	:00:33	Power Cyc	e Devices P	ast Forwar	d Time						Realt	ime
0.0	Routers	12	841 (1941)	262004 B	21.09	Scenario 0 New Dek	ete f	Fire	Last Status	Source	Destinatio	n Ty
1 State 1	- 4		Device to D	rag and Dre	p to the Work	Toggle PDU List Wir	ndow	¢				>

Common tools bar: With help of this toolbar, we can manipulate topologies. For example, select, move layout, place note, delete, resize shape, add simple/complex Protocol data unit (PDU).

File	Edit	Options	View	Tools	Exte	ensions	Help													
		8) []	B.	ı م	<mark>ر (</mark>	•	8											i	?
Γ	ogic	al		[Roo	ot]			Ne	w Cluster		М	ove Object		Set	Tiled Bac	kground	d I	Viewp	ort	
	8)																	Ŷ		
Tir	me: 00	:00:33	Pow	er Cycle	e Dev	vices F	ast Fo	orward	l Time									Realtime		
		Routers	5		841	1941	2620	CM 26	21XM 281	,	Scer New Toggle PD	ario 0 Delete	• •	Fire	Last Stat	tus	Source	Destinat	ion	Ту
				a Device to Drag and Drop to the Work					lork				<						>	

Workspace: Here, we create topologies and display simulations.



Real-time/Simulation tabs: With these tabs, the operating mode can be switched between the real and simulation modes. Buttons are also provided to control the time.
File Edit Options View T	iools Extensions Help	đ		
Logical	[Root] New Cluster	Nove Object	Set Tiled Background	Viewport
18				î Liz
				- C
				×
				Q
				8
t				, O B
Time: 00:00:33 Power	Cycle Devices Fast Forward Time			Realtime
eg 🛹 🔳 📷 🗲	66666	Scenario 0 v	Fire Last Status Sou	urce Distination
Routers	[[1841]/ [[1941]/ [282009] [282009] [[2811]/	New Delete		
9 9 0 0	a Device to Drag and Drop to the Work	Toggle PDU List Window	د	,

User-created packet box: From this area, you can create highly-customized packets in order to test your topology.

File Edit Options View	w Tools Extensions	нер			
🗋 🗁 🖬 🗁 🗐	000	PPP 🕅 🐺			I) ?
Logical	[Root]	New Cluster	Move Object	Set Tiled Background	Viewport
< <tr> C Time: 00:00:33 Po (2) (2) (2) (2)</tr>	wer Cycle Devices	Fast Forward Time	Scenario 0 v	Fire Last Status Sour	Realtime
Routers	a Device to D	ran and Drop to the Work	New Delete Toggle Politist Window		,

Network component box: All of the network and end devices available in Packet Tracer are listed in this component. It can be sub-divided into two areas:

File Edit Options View Tools Extensions Help				
📋 🖿 🖶 🗁 🗊 💭 🐢 🔎 🖊	- 🔤 🌅			i) ?
Logical [Root] New	Cluster	Move Object	Set Tiled Background	Viewport
				Pealtime
Hine: 0.00.35 Prover Cycle Devices Past forward			Fire Lock Obstan	Reduine
Routers		Scenario 0 V New Delete Toggle PDU List Window	rire Last Status	Source Destination Iy

Device-type selection box: Device categories are given here.

Device-specific selection box: Given a device category, this box displays different device models within that category.

Topic 175: Creating a Simple topology

In this topic, we create our first simple topology in Packet Tracer.

Step 1: Run Packet Tracer.

Step 2: Click **End Devices** in **Device-type selection box**. This will select the end devices category in **Device-specific selection box**.



Drag-and-drop a **Generic** PC icon into the Workspace. Do the same for a **Generic** laptop icon.

Step 3: Click on **Connections** in **Device-type selection box**. Then, click on **Copper Cross-Over** in **Device-specific selection box**.



Now, click on **PCO**, and select **FastEthernet**. Click on **LaptopO** and select **FastEthernet**. When the link is up, the color of link status LED becomes green.

Step 4: Click on the PC, and go to the **Desktop** tab.



Click on **IP Configuration**, and enter an IP address and subnet mask. No information required for a default gateway and DNS server. Close the window

2			PC0	
Physical	Config	Desktop	Custom Interface	
IP Co IP Co DHC IP Add Subnet	onfiguration nfiguration P ress : Mask : Gateway	● Stati	c	X
DNS Se	erver			
	Configurati P 🔿 Auto	on Config 🔘	Static	
IPv6 A	ddress			/
Link Lo	cal Addres	s FE80:	:290:21FF:FE83:DC	IA
IPv6 G	ateway			
IP∨6 D	NS Server			

Step 5: Repeat step 4 for the laptop. Make sure that both of the IP addresses are in the same subnet.

Step 6: Click on the laptop, open the **command prompt**, and ping the IP address of the device at the other (PC) end to check connectivity.

Command Prompt X
Packet Tracer PC Command Line 1.0 PC>ping 10.1.1.2
Pinging 10.1.1.2 with 32 bytes of data:
Reply from 10.1.1.2: bytes=32 time=6ms TTL=128
Reply from 10.1.1.2: bytes=32 time=7ms TTL=128
Reply from 10.1.1.2: bytes=32 time=7ms TTL=128
Reply from 10.1.1.2: bytes=32 time=7ms TTL=128
Ping statistics for 10.1.1.2:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 6ms, Maximum = 7ms, Average = 6ms
PC>

Step 7: This topology can be saved by going to **File** | **Save As**. The topology with the devices in the same state will be saved with a **.pkt**.

Topic 176: Introduction to Cisco and PT devices

This topic describes Cisco and PT devices available in Packet Tracer.

Network devices form the core of networking. Routers, switches are used to Interconnect end devices such as PCs, laptops, servers.

Routers in Packet Tracer: To examine the Cisco and **Generic** routers. Select **Routers** in the **device-type selection box**, then, go to the **device-specific selection box**.

Cisco 1841: An **Integrated Service Router (ISR)**. Contains two Fast Ethernet ports, two slots for **High Speed WAN Interface Cards (HWICs**), and one slot for **Advanced Integration Module (AIM)**.

Cisco 1941: Similar to **Cisco 1841.** But runs on Cisco IOS Version 15. There are two ports that operate at Gigabit Ethernet speeds.

Cisco 2620XM: A multiservice router. It contains one Fast Ethernet port, two slots for WAN interface cards, and one slot for AIM.

Cisco 2621XM: Similar to Cisco 2620XM, except that this router has two Fast Ethernet ports.

Cisco 2811: An ISR router with two Fast Ethernet ports, four WIC slots, and a dual slot for AIM.

Cisco 2901: Two Gigabit Ethernet ports, four WIC slots, and two **Digital Signal Processor (DSP)** slots. Cisco IOS Version 15 runs on it.

Cisco 2911: Three Gigabit Ethernet ports and all the other features of **Cisco 2901**.

Generic Router: contains 10 slots, and is a custom router running on Cisco IOS.

Switches in Packet Tracer: To examine the Cisco and **Generic** switches. Select **Switch** in the **device-type selection box**, then, go to the **device-specific selection box**.

Cisco 2950-24: a managed switch. There are 24 Fast Ethernet ports.

Cisco 2950T-24: It has two Gigabit Ethernet ports in addition to the 24 Fast Ethernet ports.

Cisco 2960-24TT: It contains 24 ports and has **Small Form-factor Pluggable (SFP)** modules.

Cisco 3560-24PS: It is a layer 3 switch and performs routing and switching. The suffix **PS** means it supports **Power over Ethernet** (**PoE**). It can power up IP phones without using power adapters.

Bridge PT: It is a device used to segment a network with only two ports.

Generic Switch PT: It is a customizable Packet-Tracer switch with 10 slots and several modules and runs on Cisco IOS.

Hub PT: It is a Packet Tracer device with 10 slots, and is an oldest way to connect end devices.

Repeater PT: It is used to boost the signal on a wire when the distance between two points is high.

Coaxial Splitter PT: It is used to split a single coaxial connector into two. It contains three coaxial ports.

Topic 177: Customizing Devices with Modules

In this topic, we customize devices with modules and learn naming conventions in PT.

A piece of hardware that contains several device interfaces e.g. **HWIC-4ESW** (High-Speed WAN Interface Card) is a module having four Ethernet (10 MBps) ports. There is a power switch located on the right-hand side of each device. If you see a green LED on the power switch, then it indicates that device is powered up. If you want to add a module, you need to click on this switch to turn the device off. This is similar to a real router/switch. Drag a module from the modules list and drop it onto an empty slot. The module will automatically return to the module list, in case it does not fit into that slot.



A module can be removed by first, powering off the device and then, dragging it from the slot back to the module list.

Naming convention for Interfaces:

Router Interfaces can be identified by their names. The interfaces can be broadly grouped as:

Copper Ethernet Interface: is a normal LAN interface, employs RJ-45 connector with a copper cable. **Ethernet** (10 MBps), **FastEthernet** (100 MBps), and **GigabitEthernet** (1000 MBps). A number followed by **E**, **FE**, **CE**, **CFE**, or **CGE** is used to identify modules with **Ethernet** interfaces. **SW** is used for those modules, which provide switching features when used on routers. **Examples**:

- HWIC-4ESW (four Ethernet switching ports)
- WIC-1ENET (single Ethernet port)
- **NM-1E** (single Ethernet port)
- NM-1FE-TX (single Fast Ethernet port)

Fiber Ethernet Interface: similar to Copper Ethernet Interface but employs fiber cables. Examples:

• NM-1FE-FX (single Fast Ethernet fiber media).

Serial Interface: synchronous modules are identified by T while asynchronous – A/S.

Examples:

• WIC-1T (a single synchronous serial port)

Modem Interface: these interfaces consist of RJ11 ports with analog telephone cables and are identified by **AM**.

Examples:

• WIC-8AM (eight RJ11 ports)

If you have created a device that you have customized with a set of modules, you can save it for future use. 1- Drag-and-drop a network device into the work area. 2-Turn the device off. 3-Add modules to it. 4- Click **Tools** | **Custom Devices Dialog.** 5-Click **Select**, and then click on the device that was just customized. 6- Provide a name, and then click on **Add** and **Save.** The device is saved with a .ptd extension.

Topic 178: Accessing CLI of a Device

In this topic, we describe how to access Command Line Interface (CLI) of a device in Packet Tracer.

There are two ways to access the Command-line Interface of a device in Packet Tracer:

The CLI Tab: It can be used to configure devices e.g. you can assign IP addresses to router interfaces. You can use Paste and Copy buttons to copy text to and from the

command line. Click on a network device, go to the **CLI** tab to access the Cisco IOS command line interface, and you can enter commands as on a real device (router).

a contractor a con			Rout	ter0		
Physical	Config	CLI				
MOE HWIC HWIC HWIC- WIC- WIC WIC WIC WIC	DULES IC-2T 4ESW IC-8A AP-AG-B 1AM 1ENET C-1T C-2AM C-2T Cover		P Zoom In	hysical Device Original Size	e View	coom Out
The HWIC providing	-2T is a Cis 2 serial por	co 2-Ports.	Customize Icon in Physical View t Serial High-Speed WA	N Interface Card,	Customize Icon in Logical View	

R.	Router0	- 🗆	×
Physical Config CLI			
	IOS Command Line Interface		
third-party authority to Importers, exporters, di compliance with U.S. and agree to comply with app	rryptographic products does not impry o import, export, distribute or use encryption. istributors and users are responsible for d local country laws. By using this product you plicable laws and regulations. If you are unable		^
A summary of U.S. laws of http://www.cisco.com/ww	local laws, return this product immediately. governing Cisco cryptographic products may be found a l/export/crypto/tool/stqrg.html	it:	
export@cisco.com. Cisco 1841 (revision 5.0 Processor board ID FTX0)	0) with 114688K/16384K bytes of memory. 9472185		
M860 processor: part nur 2 FastEthernet/IEEE 802 191K bytes of NVRAM. 63488K bytes of ATA Comp	mber 0, mask 49 .3 interface(s) pactFlash (Read/Write)		
Cisco IOS Software, 1843 RELEASE SOFTWARE (fc2) Technical Support: http: Copyright (c) 1986-2007	<pre>l Software (C1841-ADVIPSERVICESK9-M), Version 12.4(18 ://www.cisco.com/techsupport by Cisco Systems, Inc.</pre>	5)T1,	
Complied wed 18-Jul-07 (J4:52 by pt_team		
Continue with configurat	tion dialog? [yes/no]:		\sim
	Сору	Pa	ste

You should refer to the HELP files which lists the IOS commands supported by Packet Tracer.

The Console port: No difference with respect to CLI tab method. The use of console port makes the topology look similar to the real world. Let's illustrate with an example.

Step 1: Click **End Devices** in **Device-type selection box** and then drag and drop a PC or laptop from **Device-specific selection box** to the workspace.

Step 2: In a similar fashion, add a router, let's say 2621 XM.

Step 3: Click **Connections** in **Device-type selection box** and then select console port from **Device-specific selection box** to the workspace.

Step 4: Connect the console cable of the network device to the RS-232 port of the PC/laptop.

Step 5: Click the PC/laptop, go to the **Desktop** tab, and open **Terminal.** Click **OK** with the default settings to view the console. The router's console through its terminal is displayed:

Physical	Config Desktop Custom Interface	
		<u>^</u>
Term	ninal	ĸ
Cisco IOS (t Techni Copyri Compil	Internetwork Operating System Software tm) C2600 Software (C2600-I-M), Version 12.2(28), RELEASE SOFTWARE (fc5) ical Support: http://www.cisco.com/techsupport ight (c) 1966-2005 by cisco Systems, Inc. led Wed 27-Apr-04 19:01 by miwang	^
Cisco memory	2621 (MPC860) processor (revision 0x200) with 253952K/8192K bytes of y	
Proces M860 p Bridgi X.25 s	ssor board ID JAD05190MTZ (4292891495) processor: part number 0, mask 49 ing software. software, Version 3.0.0.	
2 Fast 32K by 63488K	ETHERNEY/IEEE 802.3 interface(s) ytes of non-volatile configuration memory. K bytes of ATA CompactFlash (Read/Write)	
	System Configuration Dialog	_
Contin	nue with configuration dialog? [yes/no]: no	
Press	RETURN to get started!	
Router	e>	¥
<		>
-		

Topic 179: Configuring Devices with Config Tab

In this topic, we configure routers/switches using the config tab in Packet Tracer.

You can configure Cisco routers and switches without using a single command. This becomes possible by using a **Config** tab. The **Config** tab contains a Graphical User Interface (GUI) options for the most common configurations. As you fiddle with a GUI, the equivalent Cisco IOS commands are also displayed.

×.	Router0 – 🗆 🗙
Physical Config	CLI
GLOBAL	Global Settings
Algorithm Settings	Display Name Router0
Static	Hostname Router
	NVRAM Erase Save
FastEthernet0/0 FastEthernet0/1	Startup Config Load Export
	Running Config Merge Export
V	
Equivalent IOS Comman	ds
System	Configuration Dialog
Continue with confi-	guration dialog? [yes/no]: n
Press RETURN to get	started!
	*

There are 3 general levels of configurations: Global, Routing, and Interface.

Global Settings: You can change the router's name. This can also be done directly in the workspace. The configuration file of a device can also be saved, erased, or exported.

2		Router0 -	×
Physical	Config	CLI	
GLO	BAL	Global Settings	
Sett	ings		
Algorithm	Settings	Display Name Router0	
ROUT	TING		
Sta	tic	Hostname Router	
RI	IP		
INTER	FACE	NVRAM Erase Save	
FastEthe	ernet0/0		
FastEthe	rnet0/1	Startup Config Load Export	
Equivalent Press RE	IOS Comm TURN to g	ands at started!	^
Router>e Router#c Enter co Router(c Router(c	nable onfigure nfiguratio onfig)#in onfig-if);	cerminal on commands, one per line. End with CNTL/Z. cerface FastEthernet0/0	~

The **Global's Algorithm Settings** are meant for advanced users. You can minutely tweak your device to see how it responds to certain situations.

Interface Settings: The IP, MAC addresses and subnet mask can be entered here. To view the port status, you need to check the "On" of "Port status".

8	Ø			Route	er0 – 🗆 🗖	x
	Physical	Config	(CLI		
	GLO	BAL	~	F	FastEthernet0/0	
	Sett	ings	1	Port Status		Dn
	Algorithm	Settings		Bandwidth	🔍 100 Mbps 💛 10 Mbps 🗹 Au	to
	ROUT	FING		Duplex	🥑 Half Duplex 🔘 Full Duplex 🗹 Au	to
	Sta	atic		MAC Address	0030.F247.7301	
	RI			IP Configuration		
	EastEthe	ernet0/0		IP Address		1
	FastEthe	ernet0/1		Subnet Mask		
)
				Tx Ring Limit	10	
				5		
			\sim			
	Equivalent	IOS Comm	an	ls .		
	Router>e	nable				
	Router#c	onfigure t	e:	minal		
	Enter co	nfiguratio	n	commands, one per line.	End with CNTL/Z.	
	Router (c	onfig_if)	eı	tace FastEthernet0/0		
	Router (c	onfig-if)	te:	it		
	Router (c	onfig) #int	e	face FastEthernet0/0		
	Router (c	onfig-if)#	ŧ			\checkmark

Routing Settings: Here, there are two options for configuring **static** and **dynamic** routing. For static routing configuration, you can enter network address, net mask, and its next hop address, and then click on **Add**.

GLOBAL	Static Poutos	
Settings	Static Routes	·
Algorithm Settings	Network	
ROUTING	Mack	
Static	Mask	
RIP	Next Hop	
INTERFACE		
FastEthernet0/0		Add
EastEthernet0/1		
rabiterioritoto, 1	Natural Address	
	Network Address	
	,	Remove
	,	Remove
		Remove
		Remove
quivalent IOS Comma	ands	Remove
guivalent IOS Comma	ands n commands, one per line. End with CNTL/Z.	Remove
guivalent IOS Comma Enter configuratio Reuter (config) #int	ands n commands, one per line. End with CNTL/Z. erface FastEthernet0/0	Remove
quivalent IOS Comma Inter configuratio Router (config) #int Router (config-if) #	ands n commands, one per line. End with CNTL/Z. erface FastEthernet0/0	Remove
guivalent IOS Comma Enter configuratio Router(config)#int Router(config)#int Router(config)=f)#	ands n commands, one per line. End with CNTL/Z. erface FastEthernet0/0	Remove
quivalent IOS Comma inter configuratio Router (config-if) # Router (config-if) # Router (config-if) #	ands n commands, one per line. End with CNTL/Z. erface FastEthernet0/0 exit erface FastEthernet0/0	Remove
guivalent IOS Comma Enter configuratio Router(config)#int Router(config)=int Router(config=if)# Router(config=if)#int	ands n commands, one per line. End with CNTL/Z. erface FastEthernet0/0 exit erface FastEthernet0/0	Remove
guivalent IOS Comma Enter configuratio Router (config-if) # Router (config-if) # Router (config-if) # Router (config) #int Router (config) #int	ands n commands, one per line. End with CNTL/Z. erface FastEthernet0/0 exit erface FastEthernet0/0	Remove

Dynamic routing such as Routing Information Protocol (RIP) can be configured by adding only the network IP.

R	Router0 – 🗆 🗙			
Physical Config	CLI			
GLOBAL	A RIP Routing			
Settings	Network			
Algorithm Settings	Add			
Static	Network Address			
RIP				
SWITCHING				
VLAN Database				
INTERFACE				
FastEthernet0/0				
FastEthernet0/1				
	Remove			
	~			
Equivalent IOS Comm	ande			
Press PETURN to g	et started!			
Router>enable				
Router‡configure terminal Enter configuration commands, one per line. End with CNTL/Z.				
Router (config) #ros	uter rip			
Router (config-rou	ter)# V			

VLAN database: Available on a switch. It allows you to create and remove VLANs.

Topic 180: Generic IP End Devices in PT

This topic describes Generic IP End Devices in Packet Tracer.

Network devices such as switches, routers are the core of a network. End devices (PCs, servers) are the ones that use this core. PCs, laptops, tablets, personal digital assistant (PDAs), and a TV a wide range of end devices are available in PT. We can group them

- A) clients
- B) server
- C) Other devices

A) Clients: Desktops and laptops: As far as usability is concerned, there is no difference between them. Most usable and highly configurable client devices. Next, we describe those modules which are available for desktops and laptops. Addition and removal of a module require a device to be switched off.



Linksys-WMP300N: Through this wireless interface, you can configure a WLAN.

PC-HOST-NM-1AM: It can be used as a modem as it provides RJ11 interface.

PC-HOST-NM-1CE, PC-HOST-NM-1CFE, PC-HOST-NM-1CGE: Through these modules, Ethernet, FastEthernet, and GigabitEthernet connections are provided respectively.

PC-HOST-NM-1FFE, PC-HOST-NM-1FGE: Fiber version of the previous module.

PC-HOST-NM-1W, **PC-HOST-NM-1W-A**: Wireless interface for WLAN. The first has a frequency of 2.4 GHz and the second has 5GHz.

PC-HEADPHONE, **PC-MICROPHONE**, **PC-CAMERA**, **PC-USB-HARD-DRIVE**: No functionality associated with them.

B) Servers: Space for two network interfaces. Modules available are the same as PC modules, except **PC-HOST-NM-1AM** module.



Let's look at services that are available for servers. **HTTP service:** Both HTTP and HTTPS (HTTP employing Secure Sockets Layer (SSL) protocol) protocols can be supported by a web server. You can also create and edit static HTML pages. **DHCP service:** can assign IP addresses to routers. You can create and edit DHCP pools of IP addresses. The default pool is called **serverPool**, which cannot be removed or edited. **DNS service:** resolves domain names to IP addresses. **AAA service:** Authentication, Authorization, and Accounting and supports RADIUS and TACACS authentication protocols. **NTP:** Network Time Protocol ensures that the clocks of all devices are synchronized properly. **EMAIL services: SMTP** and **POP3** services are supported. **FTP Services:** Users can be created and permissions can be granted to them. **Firewall:** You can configure rules based on source/destination IP addresses and source/destination port numbers. Connections can be allowed or denied.

Topic 181: Configuring End Devices in PT

This topic configures End devices in Packet Tracer.

Click on an end device in the workspace, then go to the **Desktop** tab. A lot of utilities are available here. These can be used for testing and debugging the network.



IP Configuration: With this utility, you can assign a dynamic or static IP address to an end device. If you choose to enter a static IP address, **Subnet Mask** field gets filled according to the class of the IP address. In case you want your devices to obtain IP dynamically, then choose **DHCP** here. Configure **DHCP** on **Server-PT**.

Dial-up: End devices such as **PC-PT** and **Laptop-PT** have the **PC-HOST-NM-1AM.** This utility allows to simulate a modem dialer.

Terminal: This utility can be used for accessing the CLI through the console port. **Server-PT** device does not enjoy this utility as it does not have an RS-232 interface.

Command Prompt: It simulates the command line offered by Windows OS. Commands such as **arp**, **delete**, **dir**, **ftp**, **help**, **ipconfig**, **netstat**, **nslookup**, **ping**, **ssh**, **telnet tracert**.



Web Browser: It can be used with a **Server-PT** configured with HTTP. It provides only **back**, **forward**, **go**, and **stop** buttons. Cache or history are not available.

PC Wireless: Using this utility, you can display information such as signal strength of a **Linksys-WMP300N** module.

VPN: Virtual private network (VPN) is used to create connection for secure communication. This works when a router has been configured as a VPN server.

Traffic Generator: With this utility, you can create customized packets and send them at periodic intervals. Similar to the **Add Simple Protocol Data Unit (PDU)** and **Add Complex PDU** tools.

MIB Browser: The Management Information Base (**MIB**) Browser utility allows you to generate Simple Network Management Protocol (**SNMP**) requests.

Cisco IP Communicator: A computer can be turned into an IP phone with the help of this utility. It is not available on the **Server-PT.**

Email: You can send and receive emails with the help of this utility.

Text Editor: You can use this utility to create, edit, and save text files.

Topic 182: Packet Tracer's Simulation Mode

This topic discusses simulation mode of Packet Tracer.

In Packet Tracer's simulation mode, you can observe packets flowing from one device to another. Also, when you can click on a packet, you can see detailed information about the TCP/IP layers.

Step 1: Open Packet Tracer

Step 2: Add a PC and a server to the workspace.

Step 3: Connect two devices with a copper crossover cable.

Step 4: Assign IP addresses. **PC:** 192.168.0.1/255.255.255.0 and **Server:** 192.168.0.2/255.255.255.0. Both of them are in the same subnet.



Step 5: Click on the **real time/simulation** tab and switch to the simulation mode.

	^	Simulation Panel					Β×	
		Event List						
		Vis. Time(sec)	Last Device	At Device	Туре	Info		SVM
								×
								9
		Reast Simulation	Constant [) el e u		Captu	red to: *	<i>a</i> -
		Reset Simulation	Constant L	Jelay		(no cap	otures)	
		Play Controls						
		Back	AL	ito Capture /	Play	Capture / For	ward	10 ⁻¹
					_			2
		Event List Filters ACL Filter, ARP, BC HSRP, HSRPv6, HT NTP, OSPF, OSPFv SSH, STP, SYSLOG	- Visible Events SP, CDP, DHCP, E TTP, HTTPS, ICM 6, PAgP, POP3, , TACACS, TCP,	HCPv6, DNS, P, ICMPv6, IP RADIUS, RIP, TFTP, Telnet,	, DTP, E Sec, IS , RIPng, , UDP, V	IGRP, EIGRPv6, FTF AKMP, LACP, NDP, N , RTP, SCCP, SMTP, 3 TP	P, H. 323, IETFLOW, SNIMP,	
		Ed	it Filters			Show All/None		
>	~							<u>l</u> el
						Event List	Simu	lation
_		_		_				

Step 6: Click on the Auto Capture / Play button. Packet capture begins.

Step 7: From the **common tools** bar, click on the closed envelope icon. Click on the PC first and then on the server.



In the event list, you will observe three entries: creation of an ICMP packet, ICMP echo sent, and ICMP reply received.

Europhi	int oner						
Event	list		4.0.1				
Vis.	Time(sec)	Last Device	At Device	Туре	Info		
9	284.054		PC0	ICMP			
	284.055	PC0	Server0	ICMP			
	284.056	Server0	PC0	ICMP	• J		
							:
							0
							1
Deres C		Constant Dala			Capt	tured to: 🐭	
Reset 5		Constant Dela	ý		18-	48.948 s	
Play Co	ontrols						
	Back	Auto (Capture / Play	1	Capture / Fo	orward	- 4
							1
Event	ist Eilters Mis	ible Evente					
ACL Filte	r, ARP, BGP, CI	DP, DHCP, DHC	v6, DNS, DT	P. EIGRF	EIGRPv6, F	TP. H. 323.	
HSRP, HS	SRPv6, HTTP, H	ITTPS, ICMP, IC	MPv6, IPSec	, ISAKM	P, LACP, NDP,	NETFLOW,	
NTP, OSP	PF, OSPEV6, PA	AgP, POP3, RAI	DIUS, RIP, RIP Telpet UD	Png, RTF	P, SCCP, SMTP	, SNMP,	
331, 31	Edit Filt	erc	r, lenet, ob	Sho			
-	Corerine	cra		0110	Allynonic		6
						(0
							9

To view a packet's TCP/IP layers information, click on a packet (the envelope icon). The **Outbound PDU Details** tab lists each layer's information.

PDU In	format	tion at	Device	: PC0)			×
OSI Model Outbound PDU Det	tails							
PDU Formats								
								~
<u>Ethernet II</u>							- 1	
0 4 PREAMBLE: 1010101011	8 0001.	ST MA	C: DSEB	4 51 0001	RC MAC: .9778.765	19 iC	Byt:	
TYPE: DATA 0x800	(VARIA	BLE LEI	NGTH)		FCS: 0x0			
		16 19				31	Bits	
4 IHL DSCP:	0x0		TI	L: 28			- 1	
ID: 0x6	1~1	0x0	CH		:0 4	_		
SRO	C IP: 19	2.168.	0.1	IK30F	1			
DS	T IP: 19	2.168.	D.2					
OPT: 0)×0				0×0			
DATA	(VARIA	BLE LEI	NGTH)					
	16			21	Dite			
TYPE: 0x8 CODE: 0x0	10	CHEC	KSUM	31	BIUS			
ID: 0x4	S	EQ NUI	4BER: 3					
<							>	

The simulation mode has control buttons:

Back: Each time you click on it, the simulation moves one step back.

Auto Capture / Play: The network traffic will be continuously captured when it is pressed.

Capture/Forward: Each time, you want to move the packet from one place to another you have to press this button.

Topic 183: Connecting Devices and Link Status

This topic discusses how to connect devices and to check the link status in Packet Tracer.

In Packet Tracer, there are a number of cables available to connect devices. Click **Connections** icon from the **device-type selection box**. Now, you can view the various cables in **device-specific selection box**.

Automatically choose connection type: If you have no idea which cable to use, this option automatically allows two devices to get connected with the best cable.

Console: The console port of a network device can be connected to the RS-232 port on a PC/laptop. This allows you to view the network device's console from a PC/laptop.

Copper straight-through: It is a standard Ethernet cable that connects devices operating in different layers of the OSI model. For example, hub to router, switch to PC.

Copper cross-over: This Ethernet cable connects devices such as hub to hub, PC to PC, PC to router, and PC to printer i.e. devices operating in the same OSI layer.

Fiber: Connects Fast Ethernet and Gigabit Ethernet ports of a fiber port.

Phone: RJ11 cable. It connects the analog phone to a VoIP phone. Also, modem interface of routers can be connected.

Serial DCE and DTE: Serial cables connect routers together. There is a clock symbol on the Data Circuit-terminating Equipment (DCE) end. If you have selected Serial Data Terminal Equipment (DTE), the DTE end will be the first device connected with this cable and the DCE end will be the next. In case of the Serial DCE cable, just the opposite happens.

Link status: Once you have connected devices together, you will find a light, at each end of the cable. The light indicates the state of the connection.

- **Bright green**: When the physical link is up, it is indicated by bright green. Status of the line protocol is not indicated by it.
- Blinking green: Link activity is represented by it.
- **Red**: In case a physical link is down, it is indicated by red. This can be due to incorrect cables or by a port being administratively shut down.
- Amber: When the port of a switch is running the Spanning Tree Protocol (STP) algorithm to detect layer 2 loops, the light becomes amber.

Topic 184: Testing Connectivity with PDUs

In this topic, we test connectivity with Protocol Data Units (PDUs) in Packet Tracer.

When you have created a topology in Packet Tracer, then you would like to test connectivity between devices. Connectivity can be tested by: using either simple or complex Protocol Data Units (PDUs), or pinging devices from their command-line interface. For large topologies, the PDU option is quicker.

Simple PDU

The Add Simple PDU tool relies on Internet Control Message Protocol (ICMP). Let's create a simple topology to demonstrate the working of simple PDU.

Step 1: Add a PC and a server to the workspace and connect them using a copper crossover cable.

Step 2: Click on PC and use IP configuration. Type: 192.168.0.1/255.255.255.0. Do the same for the server and type: 192.168.0.2/255.255.255.0.

Step 3: Go to the common tools bar, click on the closed **envelope icon** or simply press key *P*. Click the envelope symbol on PC first and then on the server. Then, look at the **User Created Packet** box.

Fire	Last Status	Source	Destination	Туре	Color	Time (sec)	Periodic	Num	Edit
•	Successful	PC0	Server0	ICMP		0.000	N	0	(edit
•									►

Complex PDU

We use the previous example to understand working of Complex PDU.

Step 1: Click on the open envelope icon or press *C*.

Step 2: Click on the PC. This opens the Create Complex PDU dialog box.

Create C	omplex PDU ×
Source Settings Source Device: PC0 Outgoing Port: FastEthernet0	 Auto Select Port
PDU Settings	
Select Application:	HTTP -
Destination IP Address:	192.168.0.2
Source IP Address:	192.168.0.1
TTL:	32
TOS:	0
Starting Source Port:	12345
Destination Port:	80
Size:	0
Simulation Settings	
One Shot Time: 0	Seconds
O Periodic Interval:	Seconds
	Create PDU

Select application as HTTP, fill the Destination IP address = 192.168.0.2, Starting Source Port, (let's say 12345), and Time = 0, and then click on the Create PDU button.

Step 3: Now, click on the server. Examine the **user-created packet box**.

Fire	Last Status	Source	Destination	Туре	Color	Time(sec)	Periodic	Num
٠	Successful	PC0	192.168.0.2	TCP		0.000	Ν	0
<								>

The entry indicates a successful TCP three-way handshake.

Topic 185: Clustering a Topology

This topic discusses how to cluster a topology in Packet Tracer.

When large topologies are created, understanding them becomes difficult. Clustering combines several devices that you choose into a single cloud icon. Upon double-clicking the cluster, it will get expanded and will display the devices normally.

Step 1: Let's create a topology that consists of three switches and nine PCs. Select **End Devices** in **device-type selection box.** From the **Device-specific selection box**, drag-and-drop 9 PCs in the workspace. Select **switches** in **device-type selection box**. From the **Device-specific selection box**, drag-and-drop three 2950-24 switches in workspace. Select **Connections** in **device-type selection box**. From the **Device-specific selection box**, drag-and-drop three 2950-24 switches in workspace. Select **Connections** in **device-type selection box**. From the **Device-specific selection box**, drag-and-drop Copper Straight-through cable in workspace. Connect a PC with a switch. Use copper cross-over cables to connect the switches.



Step 2: Combine PC0, PC1, PC2 and Switch0. Select PC0, PC1, PC2, and Switch0 by clicking on whitespace. Click on the **New Cluster** button on the top-right corner.

Step 3: Form a group of PC3, PC4, PC5 and Switch1. Click on whitespace next to PC3. Drag your mouse to select PC3, PC4, PC5, and Switch1. Click on the **New Cluster** button on the top-right corner.

Step 4: Repeat the same procedure done in **Steps 2 and 3** for combining PC6, PC7, PC8 and Switch2.



Double-clicking on a cluster expands it and displays only the devices within it.

Topic 186: Creating Cities, Offices & Wiring Closets

This topic discusses how to create cities, offices, and wire closets in Packet Tracer.

Packet Tracer can simulate the required environment logically and physically. In the physical workspace, it gives logical topology a physical dimension. Thus, making logical topologies more realistic. There are 4 environments available in the physical workspace:

- Intercity,
- City,
- Building, and
- Wiring closet.

1-Intercity: Being the largest environment, it consists of cities. You can create cities, buildings, and wiring closets in this layer.

2-Cities: Buildings and wiring closets are part of it. The default city name is **Home City**. You can drag and place cities anywhere on the intercity map.

3-Buildings: It contains only wiring closets. **Corporate Office** is default name.

4-Wiring closet: This layer contains only devices. Default name is Main Wiring Closet.

Moving devices physically: Packet Tracer places all devices that are used in the logical workspace in Main Wiring Closet. Now we learn how to move them.

Step 1: Create a topology consisting of two PCs in the logical workspace.

Step 2: As Ethernet has distance restrictions, switch off both the PCs and replace their default modules with **PT-HOST-NM-1FGE.**

Step 3: Connect both of the PCs with a fiber cable and assign IP addresses.



Step 4: Switch to the physical view, and click on the **New City** button. Rename it Lahore. Click on this city and create a new building, and then create a new wiring closet within this building.



Step 5: Use the **NAVIGATION** button and go to **Home City** | **Corporate Office** | **Main Wiring Closet**. Both the PCs we inserted in the logical workspace are located here. Jump to their location.



Step 6: Use the **Move Object** button and move one of the PCs to **Lahore** | **Office Building** | **Wiring Closet.**





Topic 187: Managing Cables and Distances

In this topic, we discuss how to manage cables and distances in physical space of Packet Tracer.

Assume that we have created two cities in Packet Tracer. Each consists of a PC. Both PCs are connected via fiber. In physical view, we can measure a cable's distance by placing the pointer on the cable.



The length of Standard copper Ethernet cables can extend up to 100 meters. Let's test that the connection comes down due to the distance.

Step 1: Create a topology consisting of two PCs in the logical workspace.

Step 2: Connect both of the PCs with a copper cable and assign IP addresses.

Step 3: Switch to the physical view, and click on the **New City** button. Rename it Lahore. Click on this city and create a new building, and then create a new wiring closet within this building.

Step 4: Use the **NAVIGATION** button and go to **Home City** | **Corporate Office** | **Main Wiring Closet**. Both the PCs we inserted in the logical workspace are located here. Jump to their location.

Step 5: Use the **Move Object** button and move one of the PCs to **Lahore** | **Office Building** | **Wiring Closet.**

Step 6: Check the distance between them. In case the distance is less than 100 meters, move them further apart, so that the distance becomes greater than 100 meters.

Step 7: When you go to the logical view, you will find that the connection comes down due to the distance as indicated by red color of the link status.

Cable manipulation

As number of devices increase, it becomes confusing to see which cable connects to what. This can be reduced by adding colors to cables. In the physical view, click on a wire and then, choose **Color Cable.** Pick a color from the Select Color dialog box.



Topic 188: Static Routing with GUI

This topic configures static routing with GUI in Packet Tracer.

In **static routing algorithms**, routes change very slowly over time, often as a result of human intervention. A human manually edits a router's forwarding table. Let's learn the usage of the GUI feature with the help of a network consisting of four routers in a ring topology, with no PCs or loopback interfaces.

Step 1: Drag and drop 4 routers in the workspace.

Step 2: Click on a router icon, go to the **Config** tab, select an interface, and configure the IP address. Check the **On** checkbox. This will bring the port status up.



Router	Interface	IP Address
R1	FastEthernet0/0	192.168.10.1
	FastEthernet0/1	192.168.20.1
R2	FastEthernet0/0	192.168.10.2
	FastEthernet0/1	192.168.30.1
R3	FastEthernet0/0	192.168.20.2
	FastEthernet0/1	192.168.40.1
R4	FastEthernet0/0	192.168.30.2
	FastEthernet0/1	192.168.40.2



Step 3: Now go to the **ROUTING** section, and click on **Static**. The settings that we use for configuring static routing using the GUI are shown next:

Device	Network/Mask	Next Hop
R1	192.168.30.0/255.255.255.0	192.168.10.2
	192.168.40.0/255.255.255.0	192.168.20.2
R2	192.168.20.0/255.255.255.0	192.168.10.1
	192.168.40.0/255.255.255.0	192.168.30.2
R3	192.168.10.0/255.255.255.0	192.168.20.1
	192.168.30.0/255.255.255.0	192.168.40.2
R4	192.168.10.0/255.255.255.0	192.168.30.1
	192.168.20.0/255.255.255.0	192.168.40.1

Step 4: Test the connectivity between all of the routers with the help of **simple PDU**. Click on a router first and then click on another router. Look at the **User Created Packet** box.

Fire	Last Status	Source	Destination	Туре	Color	Time(sec)
•	Successful	Router4	Router2	ICMP		0.000
•	Successful	Router2	Router4	ICMP		0.000
•	Successful	Router1	Router2	ICMP		0.000

Step 5: Let's view the routing table of a router. Go to the **Common tools bar,** click on the inspect icon. Select a router and click on it. Then select **Routing Table**.

					Routing Table for	or Router1
	Туре	Network	Port	Next Hop IP	Metric	
С		192.168.10.0/24	FastEthernet0/0		0/0	
С		192.168.20.0/24	FastEthernet0/1		0/0	
S		192.168.30.0/24		192.168.10.2	1/0	
S		192.168.40.0/24		192.168.20.2	1/0	

Topic 189: Static Routing with CLI

This topic configures static routing with CLI in Packet Tracer.

To learn the usage of the CLI feature, assume a network consisting of four routers in a ring topology, with no PCs or loopback interfaces.

Step 1: Drag and drop 4 routers in the workspace.

Step 2: Click on a router icon, go to the **CLI** tab. As the device boots up, then you will see the prompt.

Step 3: Assign IP addresses to R1's interfaces:

- R1(config)#interface FastEthernet0/0
- R1(config-if)#ip address 192.168.10.1 255.255.255.0
- R1(config-if)#no shutdown
- R1(config-if)#exit
- R1(config)#interface FastEthernet0/1
- R1(config-if)#ip address 192.168.20.1 255.255.255.0
- R1(config-if)#no shutdown
- R1(config-if)#exit

Repeat Step 3 for the remaining routers with the following configurations for their interfaces:

Router	Interface	IP Address
R1	FastEthernet0/0	192.168.10.1
	FastEthernet0/1	192.168.20.1
R2	FastEthernet0/0	192.168.10.2
	FastEthernet0/1	192.168.30.1
R3	FastEthernet0/0	192.168.20.2
	FastEthernet0/1	192.168.40.1
R4	FastEthernet0/0	192.168.30.2
	FastEthernet0/1	192.168.40.2

Step 4: Configure static routing:

- R1(config)#ip route 192.168.30.0 255.255.255.0 192.168.10.2
- R1(config)#ip route 192.168.40.0 255.255.255.0 192.168.20.2

Repeat **Step 4** for the remaining routers using the configuration shown next:

Device	Network/Mask	Next Hop
R1	192.168.30.0/255.255.255.0	192.168.10.2
	192.168.40.0/255.255.255.0	192.168.20.2
R2	192.168.20.0 / 255.255.255.0	192.168.10.1
	192.168.40.0/255.255.255.0	192.168.30.2
R3	192.168.10.0/255.255.255.0	192.168.20.1
	192.168.30.0/255.255.255.0	192.168.40.2
R4	192.168.10.0/255.255.255.0	192.168.30.1
	192.168.20.0/255.255.255.0	192.168.40.1



Step 5: Test the connectivity between all of the routers with the help of **simple PDU**. Click on a router first and then click on another router. Look at the **User Created Packet** box.

Fire	Last Status	Source	Destination	Туре	Color	Time(sec)
	Successful	Router4	Router2	ICMP		0.000
•	Successful	Router2	Router4	ICMP		0.000
٠	Successful	Router1	Router2	ICMP		0.000

Topic 190: Configuring RIP with GUI

This topic configures RIP with GUI in Packet Tracer.

Dynamic Routing Protocols: A) Form "neighbor ship" with other routers. B) Send them the directly-connected routes and other received routes. Updates are also sent out by routers, when a topology change occurs. A GUI to configure a dynamic routing protocol called **Routing Information Protocol (RIP)** is available in Packet Tracer. Let's learn the usage of the GUI feature with the help of a network consisting of four routers in a ring topology, with no PCs or loopback interfaces.

Step 1: Drag and drop 4 routers in the workspace.

Step 2: Click on a router icon, go to the **Config** tab, select an interface, and configure the IP address.



Interface	IP Address
FastEthernet0/0	192.168.10.1
FastEthernet0/1	192.168.20.1
FastEthernet0/0	192.168.10.2
FastEthernet0/1	192.168.30.1
FastEthernet0/0	192.168.20.2
FastEthernet0/1	192.168.40.1
FastEthernet0/0	192.168.30.2
FastEthernet0/1	192.168.40.2
	Interface FastEthernet0/0 FastEthernet0/1 FastEthernet0/0 FastEthernet0/0 FastEthernet0/1 FastEthernet0/0 FastEthernet0/0 FastEthernet0/1

Step 3: Click on RIP. Enter Network IP of its own interfaces.

Device	RIP Network
R1	192.168.10.0
	192.168.20.0
R2	192.168.10.0
	192.168.30.0
R3	192.168.20.0
	192.168.40.0
R4	192.168.30.0
	192.168.40.0



Step 4: Once the topology is configured, use the simple PDU to check for connectivity.

Fire	Last Status	Source	Destination	Туре	Color	Time(sec)
٠	Successful	Router1	Router4	ICMP		0.000
•	Successful	Router2	Router3	ICMP		0.000

Step 5: Use the delete tool and remove one link. Let's say we remove link between **R1** and **R2**. Use the simulation mode and test connectivity with the simple PDU. The packet takes the alternate, longer route and succeeds in reaching the destination.

Topic 191: Configuring RIP with CLI

In this topic, we configure RIP with CLI in Packet Tracer.

Let's assume a network consisting of four routers in a ring topology, with no PCs or loopback interfaces.

Step 1: Drag and drop 4 routers in the workspace.

Step 2: Click on a router icon, go to the **CLI** tab. As the device boots up, then you will see the prompt.

Step 3: Assign IP addresses to R1's interfaces:

- R1(config)#interface FastEthernet0/0
- R1(config-if)#ip address 192.168.10.1 255.255.255.0

- R1(config-if)#no shutdown
- R1(config-if)#exit
- R1(config)#interface FastEthernet0/1
- R1(config-if)#ip address 192.168.20.1 255.255.255.0
- R1(config-if)#no shutdown
- R1(config-if)#exit

Repeat Step 3 for the remaining routers with the following configurations for their interfaces:

Router	Interface	IP Address
R1	FastEthernet0/0	192.168.10.1
	FastEthernet0/1	192.168.20.1
R2	FastEthernet0/0	192.168.10.2
	FastEthernet0/1	192.168.30.1
R3	FastEthernet0/0	192.168.20.2
	FastEthernet0/1	192.168.40.1
R4	FastEthernet0/0	192.168.30.2
	FastEthernet0/1	192.168.40.2

Step 4: Enter into the config mode of RIP:

• R1(config)#router rip

Step 5: Enter the network IP addresses:

- R1(config-router)#network 192.168.10.0
- R1(config-router)#network 192.168.20.0

Step 6: Test the connectivity between all of the routers with the help of **simple PDU**. Click on a router first and then click on another router. Look at the **User Created Packet** box.

Fire	Last Status	Source	Destination	Туре	Color	Time(sec)
•	Successful	Router1	Router4	ICMP		0.000
•	Successful	Router2	Router3	ICMP		0.000

Topic 192: Load Sharing

In this topic, we look at how load sharing works in Packet Tracer.

When a source router has multiple paths to a target path, then it can load balance traffic across them. If multiple paths exist from a source to a network destination with the same metric, then RIP automatically balances the traffic between them. Assume a network that consists of four routers in a ring topology with no PCs.

Step 1: Drag and drop 4 routers in the workspace.

Step 2: Configure the IP address.

Router	Interface	IP Address	
R1	FastEthernet0/0	192.168.10.1	
	FastEthernet0/1	192.168.20.1	
R2	FastEthernet0/0	192.168.10.2	
	FastEthernet0/1	192.168.30.1	
R3	FastEthernet0/0	192.168.20.2	
	FastEthernet0/1	192.168.40.1	
R4	FastEthernet0/0	192.168.30.2	
	FastEthernet0/1	192.168.40.2	

Step 3: Configure Network IP of its own interfaces.

Device	RIP Network
R1	192.168.10.0
	192.168.20.0
R2	192.168.10.0
	192.168.30.0
R3	192.168.20.0
	192.168.40.0
R4	192.168.30.0
	192.168.40.0

Step 4: On **Router 4**, let's add a loopback interface (a virtual interface that works like a real interface and needs IP address). Go to the **CLI** tab of **R4** and enter:

• R4(config)#interface loopback 0

• R4(config-if)#ip address 192.168.100.0 255.255.255.0

Step 5: Go to the RIP config mode and enter the network IP for this loopback interface.

• R4(config)#router rip

• R4(router-if)#network 192.168.100.0.

Step 6: Create a complex PDU that is sent every two seconds.

Crea	te Complex PDU	
Source Settings		
Source Device: Router	4	
Outgoing Port:		
FastEthernet0/0	 Auto Select Por 	t
PDU Settings		
Select Application:	PING	•
Destination IP Address:	192.168.100.1	
TTL:		32
TOS:		0
Sequence Number:	이	
Size:		0
Simulation Settings		
One Shot Time:		Seconds
Periodic Interval:	2	Seconds
		Create PDI

Step 7: Turn on the simulation mode. You will find that the first packet takes the **R1**-**R2**-**R4** route while the second takes the **R1**-**R3**-**R4** route.