Lab

6

Transport Layer Protocols: UDP and TCP

What you will learn in this lab:

- How UDP and TCP react to IP Fragmentation.
- How to do detailed throughput measurements for a TCP connection.
- TCP interactive data flow and TCP bulk data flow.
- How TCP schedules retransmissions.

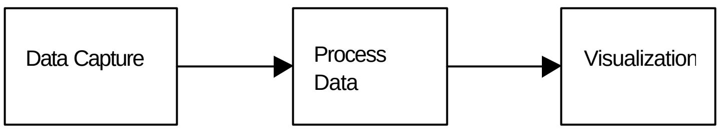
© Copyright 2001, Jörg Liebeherr. All Rights Reserved.

Table of Content

Prelab 6

1. Go to the ITLab web page `www.cs.virginia.edu/~itlab/prelab/tcp`, download the man file of `ttcp`, `ttcp`-man.doc or `ttcp`-man.txt.
   Read the man file of `ttcp`. Provide the syntax of the `ttcp` command for both client and server which executes the following scenario:
   A TCP server is on 10.0.2.6 and a TCP client is on 10.0.2.7. The TCP server is waiting on port number 2222 for a connection request. The client connects to the server and transmits 2000 bytes to the server, which are sent as 4 write operations with each 500 bytes.

In the lab exercises, you will capture traffic, and subsequently process and visualize the captured data.



`tcpdump` or `ethereal`   awk script   `gnuplot`

The data capture is done using `tcpdump` or `ethereal`. The processing of the data is done using `awk`. The visualization of the data is performed by `gnuplot`. The following prelab exercises ask you to become familiar with `awk` and `gnuplot.`

2. Login to a computer where you have a Unix account. Read the man page of `awk` and solve the following problems:
   2.1. Assume you have a file named test.data containing 4 columns of integers. Provide the syntax for awk for extracting the first and third field of each row from the input file test.data.
   2.2. Consider the file `www.cs.virginia.edu/~itlab/prelab/tcpdump.data`. The file contains the traffic of a TCP connection, which transfers a file. Write an awk script, which takes the above file, and generates from it a file as shown in `www.cs.virginia.edu/~itlab/prelab/plot.data`.
   The file plot.data has two columns. The first column has a timestamp at which a data segment was transmitted, and the second column contains the relative sequence numbers of the highest sequence number in the transmitted segment divided by 1024. The timestamp is normalized, so that the first timestamp is at time 0 sec. Further, the sequence numbers are normalized and start with "0" for the first byte of the transfer. The awk script should skip over acknowledgement packets and all packets with flag S, F or R.

3. Refer to the tutorial on `gnuplot` available at http://www.cs.uni.edu/Help/gnuplot/. In the lab you will use gnuplot for generating and modifying 2-D plots.
   3.1. The following is a set of gnuplot commands which plot the results of the content of the file "plot.data".

```
set xtics
set ytics
```

```
set xlabel "Time (sec)"
set ylabel "Bytes"
set terminal X11
plot "plot.data"  title  "Bytes Transmitted" with
linespoints
```

Explain each command in detail.

3.2. Suppose that, after the gnuplot commands above, you type the following commands.

```
set xrange[0:5]
set yrange[0:200]
replot
```

Explain what the following commands do.

3.3. Suppose that, after the gnuplot commands above, you type the following commands.

```
set terminal post eps color "Times-Roman" 14
set output "test.eps"
replot
set terminal x11
replot
```

What is the result?

3.4. Execute the sequence of commands from 3.1—3.3 for the file in `www.cs.virginia.edu/~itlab/prelab/plot.data.`
Submit a printout of the result.

4. Answer the following questions:
   - 4.1. How does TCP decide which segment size to use in the first data packet?
   - 4.2. What ICMP error is generated if a router needs to fragment a datagram with the DF bit set? How does the TCP sender learn what MTU should be used?

Deliverables: Answer all problems and tum the answers in at the beginning of the lab.

Lab 6

This lab involves the 6 workstations, Ethernet Hubs and 2 Cisco 7000 routers.

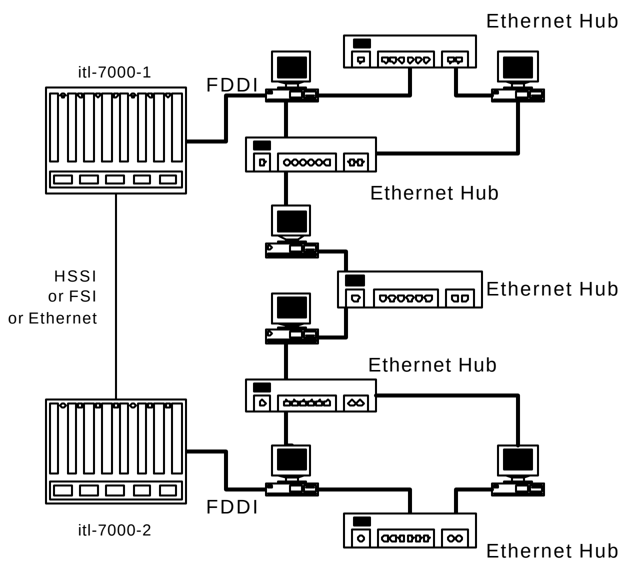


Figure 1: Network Topology in Lab 6.

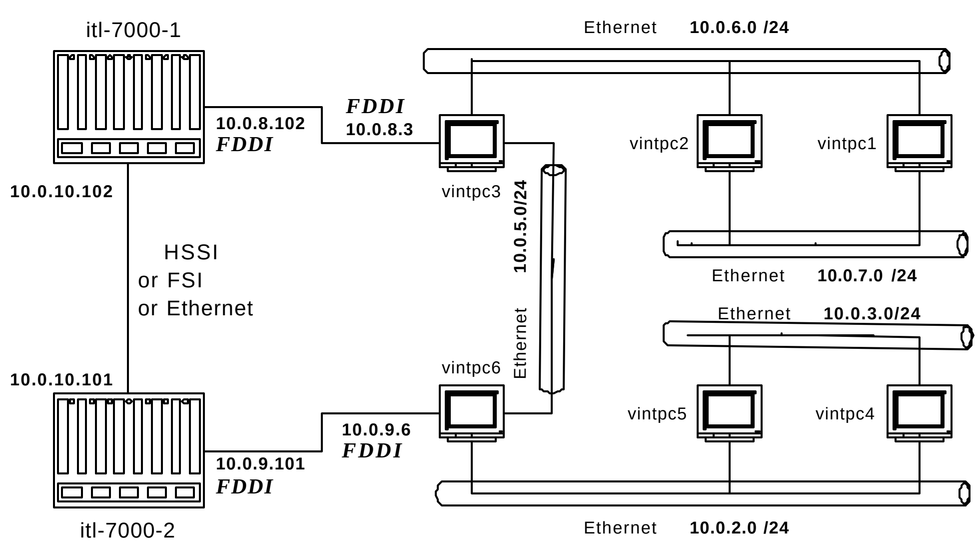


Figure 2: Logical Configuration with IP Addresses.

Setup for Lab 6

The interfaces of the PCs should be setup as shown in Table 1.

Machine	First Ethernet Interface [eth0]	Second Ethernet Interface [eth1]	Fddi Interface [fddi0]
vintpc1	**10.0.7.1/24**	**10.0.6.1/24**	
vintpc2	**10.0.7.2/24**	**10.0.6.2/24**	
vintpc3	**10.0.5.3/24**	**10.0.6.3/24**	**10.0.8.3/24**
vintpc4	**10.0.3.4/24**	**10.0.2.4/24**	
vintpc5	**10.0.3.5/24**	**10.0.2.5/24**	
vintpc6	**10.0.5.6/24**	**10.0.2.6/24**	**10.0.9.6/24**

Table 1: IP addresses of the PC's.

The IP addresses of the routers should be as shown in Table 2.

Machine	Interface type / IP	Interface type / IP
itl-7000-1	FDDI0/0 /10.0.8.102/24	HSSI or FSI or Ethernet/ 10.0.10.102/24
itl-7000-2	FDDI0/0 /10.0.9.101/24	HSSI or FSI or Ethernet/ 10.0.10.101/24

Table 2: IP addresses of the Cisco 7000 routers.

Part 1. Setting up the network topology and routing table

Exercise 1-1.

Verify that the machines and cables are setup as shown in Figure 1. If the setup is not correct, coordinate with the Lab instructor and other groups and setup the machines properly.

Exercise 1-2.

- Verify that the network interfaces are setup as shown in Table 1. If the setup is not correct, reconfigure the interfaces.
- The IP addresses of the interfaces of the Cisco 7000's are shown in Table 2.
- Please make sure that you change the netmask to `255.255.255.0` every time you make a change to the interface IP address.

Exercise 1-3.

Set up the routing tables on your machine as shown in the table below. You need to be able to `ping` every interface on your local Ethernet network, all the 7000 routers, and the remote Ethernet networks.

On Machine	Destination Network	Gateway
vintpc1	10.0.5.0/24	10.0.7.2
	default	10.0.6.3
vintpc2	default	10.0.6.3
vintpc3	10.0.3.0/24	10.0.5.6
	10.0.7.0/24	10.0.6.2
	default	10.0.8.102
vintpc4	10.0.5.0/24	10.0.3.5
	default	10.0.2.6
vintpc5	default	10.0.2.6
vintpc6	10.0.7.0/24	10.0.5.3
	10.0.3.0/24	10.0.2.5
	default	10.0.9.101

Use `traceroute` to test that the routing tables are properly setup. For example, on vintpc3, `traceroute 10.0.3.4` should show a route with 2 hops via vintpc6, vintpc5.

Get a check-off when you are done.

Part 2. Learning how to use `ttcp`

The Unix utility `ttcp` tests TCP and UDP throughput over a network. Along with tcpdump or ethereal, ttcp is often used for debugging network related problems.

Running `ttcp` requires to setup `ttcp` receiving process ("`ttcp` receiver") and a `ttcp` sending process ("ttcp sender"). To start a `ttcp` receiver in TCP mode on a remote machine, telnet to the remote machine and execute the command:

```
ttcp –r –lbuflen –nnumbufs –pport
```

To start a TCP sending process on a local machine, run the command:

```
ttcp –t –lbuflen –nnumbufs –pport -D host
```

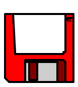
To execute `ttcp` in UDP mode, the receiving and sending processes are started with:

```
ttcp –r –lbuflen –nnumbufs –pport -u
```

```
ttcp –t –lbuflen –nnumbufs –pport –u -D host
```

Exercise 2-1. Transmit TCP data.

Experiment with the `ttcp` command by sending packets to your neighbors in TCP mode. You will setup the following scenario. The `ttcp` receiver sets up a server which waits on port 4444, the `ttcp` sender establishes a client which sends data to the server. By default, the server discards all received data. After finishing the transmission of data, both server and client processes are closed.



- Use `ethereal` to capture packets. Save the results to a file.
- Get the IP address from one of your neighbors. Telnet to your neighbor and start a `ttcp` receiver:

```
ttcp –r –l4096 –n12 –p4444
```

- On your local machine, type:

```
ttcp –t –l4096 –n12 –p4444 –D IPNeighbor
```

where ***IPNeighbor*** is the IP address of your neighbor machine.

Exercise 2-2. Transmit UDP data.

You basically repeat the previous exercise with UDP as transport protocol.

- Use `ethereal` to capture packets. Save the results to a file.
- Telnet to your neighbor machine and execute command:

  `ttcp –r –l4096 –n12 –p4444 -u`

- On the local machine, execute command:

  `ttcp –t –l4096 –n12 –p4444 –u –D` *`IPNeighbor`*

Lab Report: Use the data captured with `ethereal` to answer the following questions. (Do not include the captured data in your report.)

a. How many packets are exchanged in each of the two examples?

b. What is the range of the size of data sent in each sending packet?

Get a check-off when you are done with this part.

Part 3. MTU and Fragmentation

Next, you will study the effect of IP Fragmentation on UDP and TCP traffic.

Exercise 3-1.

- Use `ttcp` to send UDP traffic between two machines and `ethereal` to capture traffic.
- Run experiments where you send UDP datagrams between your machine and a remote machine on the same Ethernet segment.
- Increase the size of the datagrams (via the –l argument of `ttcp`) until fragmentation occurs.
- Determine the maximum size of the UDP datagram that the machines can send and receive, regardless of fragmentation.
  Save the data of two captured packets where IP fragmentation is used.

- From the `ethereal` output, determine for at least two IP datagrams the values of the fields in the IP header which are used for fragmentation (Identification, fragment offset, Don't fragment bit).

Exercise 3-2.

Modify the MTU of a network interface with the `ifconfig` command, e.g.,

```
ifconfig eth0 mtu 100
```

sets the MTU of interface eth0 to 100 bytes.

You are asked to observe how TCP discovers the maximum allowed datagram size (MTU) on the route from sender to receiver. This process is called *Path MTU discovery*. All TCP segments are sent in IP datagrams with the DF (Don't fragment) bit set. When a router needs to fragment a datagram with the DF bit set, it discards the datagram and generates an ICMP Unreachable message, which includes the MTU required. Upon receiving the ICMP error message, TCP reduces the segment size to the required MTU.

- Modify the MTU of the interfaces as shown in the table below. (Modify the MTU only on your local machine. You need to coordinate with other groups about the MTU settings on remote machines.)

	MTU on eth0	MTU on eth1
vintpc1	1500	1500
vintpc2	500	500
vintpc3	1500	800
vintpc4	1500	1500
vintpc5	500	500
vintpc6	1500	800

- Use `ttcp` to generate TCP traffic and use `ethereal` to measure traffic

  On the remote machine: `ttcp –r –l1000 –n2 –p4444`
  On the local machine: `ttcp –t –l1000 –n2 –p4444 –D` ***`IPrem`***

- The source and the destination of TCP traffic along with the interfaces you need to observe with `ethereal` is shown in the table below. The source and destinations have been assigned, so that TCP traffic goes through an intermediate router that needs to fragment the datagram. Refer to Figure 2 to determine the path of the TCP traffic and note the MTU's on the interfaces of the source, destination and the intermediate routers on the path.
- Start `ethereal` on both interfaces of your machine and set filters that capture only the TCP data you need to measure.

If you are on	Source	Destination
vintpc 1	vintpc1 (10.0.7.1)	vintpc3 (10.0.5.3)
vintpc2	vintpc3 (10.0.6.3)	vintpc1 (10.0.7.1)
vintpc3	vintpc6 (10.0.5.6)	vintpc1 (10.0.6.1)
vintpc4	vintpc4 (10.0.3.4)	vintpc6 (10.0.5.6)
vintpc5	vintpc6 (10.0.5.6)	vintpc4 (10.0.3.4)
vintpc6	vintpc3 (10.0.5.3)	vintpc4 (10.0.2.4)

Lab Report:

Provide answers to the questions posted in the exercises. Use the saved data to answer the questions. Do not include the data in your report.

a. If you observed fragmentation, describe the fields in the `ethereal` output. Moreover give an explanation why the ICMP Unreachable (Fragmentation required) error was not generated. Was the DF bit set in the datagrams?

b. If you observed an ICMP Unreachable (Fragmentation required) error message, describe how the message was used for Path MTU discovery?

c. Explain why a TCP connection on a local network does not exhibit fragmentation, while a UDP connection does?

Reset the MTU to the default value (1500) and get a check-off.

Part 4. TCP connection management

In this exercise, you will study TCP connection management.

Exercise 4-1.

- Decide on a remote machine to establish a telnet connection.
- Start `ethereal` to capture the packets that will be transmitted for the telnet connection. Save the captured data to a file.
- Establish a telnet connection as follows:

  `telnet` *RemoteIP* `discard`

  where *RemoteIP* is the IP address of a remote machine. When the login prompt appears, type Ctrl-] and type quit, to terminate the connection.

Lab Report:

Use the captured data to explain in detail the steps performed during TCP connection establishment and termination. Include the captured TCP headers of the 3-way handshake and the segments involved in closing the TCP connection.

- What is the maximum segment size of the TCP connection?
- What are the initial sequence numbers of the telnet client and the telnet server?

Get a check-off when you are done.

Part 5. TCP data exchange

In this exercise, you will observe the TCP data exchange for interactive data flow and for bulk data flow.

Exercise 5-1. Interactive data transfer

- Choose a remote machine with the IP address *RemoteIP*. Start an `ethereal` session that captures the traffic between your machine and the machine with IP address *RemoteIP*.
- Login to the remote machine, with IP address *RemoteIP*, using

  `rlogin` *RemoteIP*

and start to type characters in the window that contains the rlogin session. Observe the number of IP packets that are exchanged between the machines for each keystroke.

- Note the time at which ACKs are sent from your local machine.
- Save the captured `ethereal` data to a file.

Exercise 5-2. Bulk data transfer

- Start a new ethereal session to capture ftp traffic between your local machine and the remote machine.
- Start a ftp session to the remote machine and transfer a large file. Observe the behavior of two machines, where the sender transmits a window of data in a burst and then waits for it to be acknowledged.
- Determine the rules used by the receiver for transmitting ACKs to the sender.
- Save the captured `ethereal` data to a file.

Lab Report:

Use the captured data to answer the following questions.

a. Did you observe delayed acknowledgments in the rlogin session? Did you observe delayed acknowledgments in ftp session? What is the maximum time at which ACKs are delayed in your measurements?
b. In the ftp session, what is the range of the advertised window for the receiver of the file? Can you detect a rule which is used for setting the advertised window?
c. Which TCP flags do you observe in the captured packets? Explain their role? Are TCP flags used differently in rlogin data segments, and ftp data segments? Explain.
d. Describe and explain the TOS bits that are set in the IP headers of the rlogin and the ftp session. Explain if there are differences.

Get a check-off when you are done.

Part 6. TCP retransmission and throughput measurement

In the last part, you observe some of the error correction features of TCP, and use `ttcp` to measure the throughput of TCP transmission between two machines.

Here you will use `tcpdump` to capture data. Use the tcpdump syntax

```
tcpdump –tt –i  eth0 <expression> |tee filename
```

where "-tt" gives unformatted timestamps, "-i eth0" specifies the interface, "<expression>" is a boolean expression which describes a capture filter. Recall that "|tee filename" writes the results of the captured data to both the monitor and a file.

Exercise 6-1. TCP throughput measurement

- You will use `ttcp` to send TCP traffic to the machine with IP address *RemoteIP*.

- Start `tcpdump` with the options given above. You need to determine on which interface of your machine data will be flowing. Set the capture filter so that you only capture the traffic between your local machine and the remote machine. Save the `tcpdump` data in a file.

- Start a `ttcp` session as follows:

  On the remote machine: `ttcp –r –n2000 –p4444`
  On the local machine: `ttcp –t –n2000 –p4444 –D` ***`RemoteIP`***

The following table has the assignment of remote machines for each PC:

If you are on	Remote machine
vintpc1	vintpc4 (10.0.2.4)
vintpc2	vintpc5 (10.0.2.5)
vintpc3	vintpc6 (10.0.2.6)
vintpc4	vintpc1 (10.0.6.1)
vintpc5	vintpc2 (10.0.6.2)
vintpc6	vintpc3 (10.0.6.3)

- Once the `ttcp` session terminates, run the awk script given in file `filtertcpdump.awk`. The awk script performs the operations described in problem 2.2 of the prelab. The scripts takes as input the tcpdump output of a traffic capture of a single TCP connection, and generates a file which has one row for each TCP segment transmission. Each row has two columns: (1) a timestamp, and (2) the highest sequence number of the transmitted segment.

- Execute the awk script as follows.

```
awk –f /root/lab6/filtertcpdump.awk filename > output.data
```

where filename is the name of the file that you saved the tcpdump output. Verify that the resulting file has the desired format.

- Use `gnuplot` to visualize the data of the file output.data. You start a gnuplot session simply by typing “`gnuplot`”. You need to execute commands for:

  - Generating a 2-D plot from data in a file
  - Modifying the ranges of the axes of the plot,
  - Annotate a plot (add labels to axis, add a title)
  - Save plots to a postscript file.

All commands are given in the prelab exercises.

- Generate plots (and save the plots as postscript files), which can provide answers to the following questions. Make sure that the labels of the plots are correctly annotated.

  - As in the previous exercise, generate and save a plot which shows the rate of data transmission of your experiment. What is the approximate data rate of the transmission of the `ttcp` sender, measured in Bits per Second?

  - If you observe the visualized data over very short time periods, you will observe that data is transmitted in bursts, that is, periods of rapid transmissions of multiple segments are followed by time periods where no transmission takes place. Try to answer the following questions:

    - What is the range of the burst sizes?
    - What is a “typical” size of a burst?
    - What is the typical distance between two bursts

    Explain which factors determine the sizes of the bursts and the delay between two bursts.

  - Present a plot which shows the operation of slow start and congestion avoidance.

Exercise 6-2. Comparison of UDP and TCP throughput..

Repeat the previous exercise for a UDP transmission. Start a ttcp session as follows:

On the remote machine: `ttcp –r –n2000 –p4444 –u`
On the local machine: `ttcp –t –n2000 –p4444 –D` *`RemoteIP`* `–u`

- Generate a plot (and save the plots as postscript files), which shows the rate of data transmission of your experiment. What is the approximate data rate of the transmission of the `ttcp` sender, measured in Bits per Second? Compare with the results from Exercise 6-1.

- Compare the pattern of UDP datagram transmissions with the burst pattern of TCP segment transmission from Experiment 6-2. What are the differences? Generate a plot which shows the burst pattern of UDP transmissions, over a short period of time.

Exercise 6-3. TCP Retransmission.

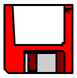
Setup an experiment, similar to the previous ones. Setup a tcpdump with filters set as before and start a `ttcp` session with the following parameters:

On the remote machine: `ttcp –r –n1000 –p4444`
On the local machine: `ttcp –t –n1000 –p4444 –D` *`RemoteIP`*

The remote PCs are different than before. Use the following table for the assignment of remote machines for each PC:

If you are on	Remote machine
vintpc1	vintpc2 (10.0.7.2)
vintpc2	vintpc3 (10.0.6.3)
vintpc3	vintpc6 (10.0.5.6)
vintpc4	vintpc5 (10.0.3.5)
vintpc5	vintpc6 (10.0.2.6)
vintpc6	vintpc3 (10.0.8.3)

Now, while the transmission is ongoing, disconnect the Ethernet cable of your machine, on which the `ttcp` traffic is flowing, for approximately one minute, and then reconnect the Ethernet cable.



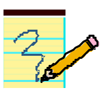
- Generate plots (and save the plots as postscript files), which can provide answers to the following questions. Make sure that the labels of the plots are correctly annotated.

    - Present a plot, which shows the rate of data transmission of your experiment. What is the approximate data rate of the transmission of the `ttcp` sender, measured in Bits per Second? Compare the results to the corresponding plots from Exercise 6-1 and 6-2.

    - Generate a plot, which enables you to observe the retransmission attempts after you disconnected the cables. How much time did it take after you reconnected the cable, until the data rate reached the same level that you observe before the cables were disconnected?

- In addition to the retransmission attempts after the cables were disconnected, determine if there were any other packet losses and retransmission attempts.

Lab Report:



Include all plots, and provide answers to the questions posed in the above exercises, and include a discussion where it was requested.

Get a check-off when you are done.

Check List for Lab 6

Complete this check list as you work through the laboratory exercises and hand the form in to a lab instructor before you leave.

Names (Please Print):____

- ☐ Turn-in Prelab 6 (Pledge your prelab.)
- ☐ Check-off for Part 1
- ☐ Check-off for Part 2
- ☐ Check-off for Part 3
- ☐ Check-off for Part 4
- ☐ Turn-in your feedback sheet.
- ☐ Check-off for Part 5
- ☐ Check-off for Part 6

Don't forget to copy your results to a floppy disk. The lab machines will be reset after each lab.

Feedback Form for Lab 6

- Complete this feedback form at the end of the lab and hand the form to a lab instructor before you leave. The feedback is anonymous. **Do not put your name on this form**.

- For each exercise, please record the following:

  - **Difficulty**: Grade the degree of difficulty using a range of –2,-1,0,1,2. (-2 = too easy, 0=just right, 2 = too difficult)

  - **Interest Level**: Grade your level of interest when completing the exercise using a range of –2,-1,0,1,2. (-2=low interest, 0= just right, 2 = high interest).

  - **Time**: How much time did you need to complete each part of the lab?

	Difficulty (-2,-1,0,1,2)	**Interest Level (-2,-1,0,1,2)**	**Time (minutes)**
Part 1. Setting up the network topology and routing table			
Part 2. Learning how to use ttcp			
Part 3. MTU and Fragmentation			
Part 4. TCP connection management			
Part 5. TCP data exchange			
Part 6. TCP retransmission and throughput measurements			

Please answer the following questions:

- What did you like about this lab?

- What did you dislike about this lab?

- Make a suggestion to improve the lab.