IS8055556: Data and Computer Communications	Recitation 2
Semester 2 5786	17 March 2026
Lecturer: Michael J. May	Tel Hai College

Network Download, Latency and Bandwidth Tradeoff, UDP Echo Client and Server

1 Network Download Experiments

The next thing we will do is do some network experiments using two Linux command line tools: `wget` and `time`.

1.1 wget

`wget` is a command line tool for downloading files over the network. It supports many protocols and features, but we'll use it for just one protocol today: HTTP. HTTP is the protocol used for downloading most web content today. wget also supports HTTPS, a combination of HTTP with a security protocol as well. We can use wget's statistics to figure out bandwidth and time in real communication exchanges.

1.2 time

`time` is a command line tool for showing how much time a particular process took on Linux. When used without any flags, it prints out three values:

1. How much **wall clock** time passed from the time the process started until it finished. This includes time that the process being monitored was running and time that it wasn't running (waiting, sleeping, etc).

2. How much time the process spent running on the CPU in **user mode**. This is time that the process being monitored was running its own code (not syscalls, interrupt handling, etc).

3. How much time the process spent running on the CPU in **kernel mode**. This is time that the process being monitored was running syscalls, interrupt handling, and so on.

1.3 What to do

Bandwidth calculations   We'll perform a few experiments with wget and time to test out download times and actual bandwidth. We'll try the following commands. For each command, write down the size of the transfer (in byte) and the wall clock time (the "real" value time prints).

- `time wget http://www2.kinneret.ac.il/mjmay/se331/331-Lecture1-NetworksIntro.pdf`

- `time wget`
  `https://mirror.accum.se/mirror/eclipse.org//oomph/epp/2024-09/R/eclipse-inst-jre-win64.exe`

- `time wget https://download.eclipse.org/oomph/epp/2024-09/R/eclipse-inst-jre-win64.exe`

- `time wget https://ftp.linux.org.tr/eclipse//oomph/epp/2024-09/R/eclipse-inst-jre-win64.exe`

- `time wget https://ftp.yz.yamagata-u.ac.jp/pub/eclipse/oomph/epp/2024-09/R/eclipse-inst-jre-win64.exe`

  – You can stop this one after a few minutes if it takes too long

Calculate the actual bandwidth for the download using the file size and elapsed time. Notice that wget prints out the bandwidth for the download at the end. Do your numbers match wget's? Why does the bandwidth differ between the last four downloads?

RTT checking   Use the following commands to try to measure the RTT to the sites that we used in the previous tests:

- `ping www2.kinneret.ac.il`

- `ping ftp.acc.umu.se`

- `ping download.eclipse.org`

- `ping ftp.linux.org.tr`

- `ping ftp.yz.yamagata-u.ac.jp`

Ping is sometimes not implemented by the remote servers, so you might not get answers from all of the above servers. If you do get answers from the server, use the RTT measurements (you can use the average) to recalculate the bandwidth for the previous downloads. Taking the RTT into consideration should help improve the accuracy of the bandwidth measurement.

2   Latency/Bandwidth Tradeoff

A company has two computers (A and B) at different locations. The job of the computers is to send fixed-sized files (of size X bytes) back and forth across a dedicated full duplex network link (no one else is using the link) which currently offers them 100Mbps of bandwidth and an RTT of 100ms.

One day, a network genie appears to the network administrator and offers her one wish. The genie will either (a) double the bandwidth of the dedicated line to 200Mbps **or** (b) reduce the dedicated line's RTT to 50ms (the genie must have some power over the speed of light).

The network administrator knows that the value of X will determine whether it's better to increase the bandwidth or reduce the RTT. For what values of X should the network administrator choose option (a)? For what values of X should she choose option (b)? Show your calculations.

3   UDP Echo Client

This week we'll introduce the basics of network communication using a UDP echo client and server. The code is written in C and Java. Both programs send packets back and forth between two computers.

We're going to use the echo client and server for future recitations and assignments, so it's worthwhile to spend time figuring out how they work.