Project 1 | Phase 1

DISTRIUTED COMPUTING

Dhruval Darji

Deion Anthony

Mirela Memic

Table of Contents

Introduction:

For Phase I of this project we implemented a file text transfer service using TCP protocol. The purpose of this project is to transfer data from a client to a server via a server-router and then receive a response from the server to the client. We used four main modules to implement this scenario. The modules are named TCPServerRouter, SThread, TCPServer, and TCPClient. Additionally, we used a custom statistics module to monitor and keep track of variable data. Variables include amount of data sent versus the amount of data received, as well as the time it takes to send data and receive a response. This module was called Stats, and it was built to be invoked on the TCPClient and TCPServer modules. Finally, we used a main module to dynamically run either the TCPServerRouter, TCPServer, or TCPClient modules from any machine without having to drastically change code for variable inputs such as IP addresses. We also implemented a feature in the main module to enable running the client and server modules continuously for data collection purposes. This module was called Main. More detail on each module can be found in the System Modules section.

Design Approach:

We began our project with the Java code provided to us for the four TCP program modules. We installed, compiled, and ran the code to implement the distributed system. We chose IntelliJ IDEA as our Java IDE for this project and integrated it with the GitHub remote storage. This allowed us to easily make the necessary changes to the project files and share them among the group members due to the version control provided by IntelliJ and Git. Once we had all of the given project files up and running properly, we added our statistics module and our main module and proceeded with the data collection.

Design Modules and Implementation:

In order to run our processes, we need at a minimum 2 machines, though using 3 is preferable. This is because the TCPServerRouter module can be run on either machines that are planning on running the TCPServer or TCPClient modules. In this explanation we will be using 3 machines. To start our process, the Main module on each machine should be updated with the IP address and port number of each other machines and their respective roles. Then machine 1 should run the Main module and input 'r' to run as the TCPServerRouter. Next, machine 1 and 2 should run when they see fit their Main modules. They should then input 's' for the TCPServer module and 'c' for the TCPClient module. Note that the machine running the TCPClient module must always specify the location of the text file they wish to read from and send to the machine running the TCPServer module. Now we get into the details of each module.

When the TCPServerRouter module is run it first creates a RoutingTable to store clients/servers that connect to it. This RoutingTable is a HashMap with String Keys and Socket Values. This enables multiple server/client connections to the TCPServerRouter that are mapped to their respective sockets. This makes searches to those addresses simple and retrievable in constant time. Additionally, addresses that have previously connected won't be marked as new connections, but will instead simply have their respective socket replace the old unused socket. Next the module creates a ServerSocket for accepting connections on a pre-specified port number. By default, we use port number 5556. Finally, for every instance of a client/server connecting it creates an instance of the SThread module to handle the connection between a Server and a Client. It then passes in an instance of the RoutingTable and the connecting client to that instance of the SThread module. The TCPServerRouter module only disconnects and shuts down if it's Running variable is switched to false, or if it is forcefully terminated. This is how the TCPServerRouter module runs.

When the SThread module is invoked by the TCPServerRouter module it instantiates new PrintWriter's, InputStreams, and OutputStreams using the socket that was passed to it it from the TCPServerRouter module. Next it takes the RoutingTable instance that was passed to it from the TCPServerRouter module and updates the RoutingTable, by setting the Key to be the

IP of the server/client and the Value be the new socket that was also passed to it. The next step occurs once the TCPServerRouter module invokes the Run method of the SThread module. When this is done the SThread module starts listening for incoming messages that contain the destination the client/server wish to connect to and prints out to notify the client/server that a connection to the router has been made. Then it sleeps for 10 seconds to allow the RoutingTable to update and for changes to take place. Then the module searched the RoutingTable to find an instance of the destination that the client/server is seeking. Once the destination is found, an instance of a new outSocket or Output Stream is created to communicate with that destination. Then, while there is still input the SThread module will do two things. The first thing is to check if the input was a termination sequence. If it was, then it will print the termination sequence to the destination and then terminate. Otherwise, if the destination socket still exists, it will print the incoming message to the destination. This is process of the SThread module.

The next module we would like to discuss is the Stats module. As mentioned before, this module was built to be implemented via both the TCP Server module and the TCP Client module. This module consists of a list of outgoing message sizes, a list of incoming message sizes and a list of transmission times. The module also has averages of all the the lists, methods to compute the averages, methods store and track data, and lastly a toString method that both prints the data to console as well as to a CSV (Comma Separated Values) file in tables. This file is then used in conjunction with Microsoft Excel to tabularize and generate charts based on the data that was outputted over the course of over a hundred simulations. This is process of the Stats module.

The TCPServer module is also invoked via the Main module in a similar fashion as the TCPServerRouter module. On instantiation, the address of the router, the port number, and the address of the client are passed in to the module's constructor. These variables are saved while the Socket, BufferedReader, and PrintWriter are instantiated. The module is invoked once the Run method is called. The first thing the module does is connect to the router via the router address and port that were passed through during instantiation. Once a connection is established the server prints out to the router the client destination address that it would like

to communicate with. The module should then receive and print to console a welcome message from the router. Finally, while the input socket is not null, the module does two things. The first thing is to check if the input was a termination sequence. If it was, then it will print the termination sequence to the client and then terminate. Otherwise, it will convert the incoming message to uppercase and then print the converted message back to the client. This is process of the TCPServer module.

The TCPClient is similar to the TCPServer except in one regard. This is the fact that this module must read in text from a file and initiate the communication to the server once connection has been established. This means that this module is also instantiated and invoked via the Run method from the Main module. On instantiation, the address of the router, the port number, and the address of the server, and the location of the input text file are passed in to the module's constructor. These variables are saved while the Stats module, Socket, BufferedReader, and PrintWriter are instantiated. Once the Run method is invoked the module connects to the router via the router address and port that were passed through during instantiation. Next, the module prepares the input file to be read from. Then, once a connection to the router is established, the module prints the server destination address that it would like to communicate with. The module should then receive and print to console a welcome message from the router. Next the module prints to the router its own address and then begin a timer for tracking transmission time of the first message. Finally, while the input socket is not null, the module does three things. The first thing is to read in from the server, print to console, stop the timer, calculate the transmission time, and then send all the data to the Stats module. The second thing is to check if the input was a termination sequence. If it was, then the module will terminate communication and break from the sequence. Otherwise, the third thing to do is read in the next line from the input file, send it to the server, and start the timer again to measure transmission time. Finally, once communication is terminated, the ComputeAverages method in the Stats module is invoked and then the toString method is invoked to print the data to the console and save it to the external file. This is how the TCPClient module works. This is process of the TCPClient module.

Client Sample Output:

Thank you for running the Client/Server Manager

What would you like to run? ('(r)outer', '(s)erver'/ (s100), or '(c)lient'/ (c100))

c

Running Client...

ServerRouter: Connected to the router.

Server: 192.168.1.5

Cycle time: 10077

Client: Non eram nescius, Brute, cum, quae summis ingeniis exquisitaque doctrina philosophi Graeco sermone tractavissent, ea Latinis litteris mandaremus, fore ut hic noster labor in varias reprehensiones incurreret. nam quibusdam, et iis quidem non admodum indoctis, totum hoc displicet philosophari. quidam autem non tam id reprehendunt, si remissius agatur, sed tantum studium tamque multam operam ponendam in eo non arbitrantur. erunt etiam, et ii quidem eruditi Graecis litteris, contemnentes Latinas, qui se dicant in Graecis legendis operam malle consumere. postremo aliquos futuros suspicor, qui me ad alias litteras vocent, genus hoc scribendi, etsi sit elegans, personae tamen et dignitatis esse negent.

Server: NON ERAM NESCIUS, BRUTE, CUM, QUAE SUMMIS INGENIIS EXQUISITAQUE DOCTRINA PHILOSOPHI GRAECO SERMONE TRACTAVISSENT, EA LATINIS LITTERIS MANDAREMUS, FORE UT HIC NOSTER LABOR IN VARIAS REPREHENSIONES INCURRERET. NAM QUIBUSDAM, ET IIS QUIDEM NON ADMODUM INDOCTIS, TOTUM HOC DISPLICET PHILOSOPHARI. QUIDAM AUTEM NON TAM ID REPREHENDUNT, SI REMISSIUS AGATUR, SED TANTUM STUDIUM TAMQUE MULTAM OPERAM PONENDAM IN EO NON ARBITRANTUR. ERUNT ETIAM, ET II QUIDEM ERUDITI GRAECIS LITTERIS, CONTEMNENTES LATINAS, QUI SE DICANT IN GRAECIS LEGENDIS OPERAM MALLE CONSUMERE. POSTREMO ALIQUOS FUTUROS SUSPICOR, QUI ME AD ALIAS LITTERAS VOCENT, GENUS HOC SCRIBENDI, ETSI SIT ELEGANS, PERSONAE TAMEN ET DIGNITATIS ESSE NEGENT.

Cycle time: 4

Client: Contra quos omnis dicendum breviter existimo. Quamquam philosophiae quidem vituperatoribus satis responsum est eo libro, quo a nobis philosophia defensa et collaudata est,

cum esset accusata et vituperata ab Hortensio. qui liber cum et tibi probatus videretur et iis, quos ego posse iudicare arbitrarer, plura suscepi veritus ne movere hominum studia viderer, retinere non posse. Qui autem, si maxime hoc placeat, moderatius tamen id volunt fieri, difficilem quandam temperantiam postulant in eo, quod semel admissum coerceri reprimique non potest, ut propemodum iustioribus utamur illis, qui omnino avocent a philosophia, quam his, qui rebus infinitis modum constituant in reque eo meliore, quo maior sit, mediocritatem desiderent.

Server: CONTRA QUOS OMNIS DICENDUM BREVITER EXISTIMO. QUAMQUAM PHILOSOPHIAE QUIDEM VITUPERATORIBUS SATIS RESPONSUM EST EO LIBRO, QUO A NOBIS PHILOSOPHIA DEFENSA ET COLLAUDATA EST, CUM ESSET ACCUSATA ET VITUPERATA AB HORTENSIO. QUI LIBER CUM ET TIBI PROBATUS VIDERETUR ET IIS, QUOS EGO POSSE IUDICARE ARBITRARER, PLURA SUSCEPI VERITUS NE MOVERE HOMINUM STUDIA VIDERER, RETINERE NON POSSE. QUI AUTEM, SI MAXIME HOC PLACEAT, MODERATIUS TAMEN ID VOLUNT FIERI, DIFFICILEM QUANDAM TEMPERANTIAM POSTULANT IN EO, QUOD SEMEL ADMISSUM COERCERI REPRIMIQUE NON POTEST, UT PROPEMODUM IUSTIORIBUS UTAMUR ILLIS, QUI OMNINO AVOCENT A PHILOSOPHIA, QUAM HIS, QUI REBUS INFINITIS MODUM CONSTITUANT IN REQUE EO MELIORE, QUO MAIOR SIT, MEDIOCRITATEM DESIDERENT.

Cycle time: 3

Client: Sive enim ad sapientiam perveniri potest, non paranda nobis solum ea, sed fruenda etiam [sapientia] est; sive hoc difficile est, tamen nec modus est ullus investigandi veri, nisi inveneris, et quaerendi defatigatio turpis est, cum id, quod quaeritur, sit pulcherrimum. etenim si delectamur, cum scribimus, quis est tam invidus, qui ab eo nos abducat? sin laboramus, quis est, qui alienae modum statuat industriae? nam ut Terentianus Chremes non inhumanus, qui novum vicinum non vult 'fodere aut arare aut aliquid ferre denique' -- non enim illum ab industria, sed ab inliberali labore deterret --, sic isti curiosi, quos offendit noster minime nobis iniucundus labor.

Server: SIVE ENIM AD SAPIENTIAM PERVENIRI POTEST, NON PARANDA NOBIS SOLUM EA, SED FRUENDA ETIAM [SAPIENTIA] EST; SIVE HOC DIFFICILE EST, TAMEN NEC MODUS EST ULLUS INVESTIGANDI VERI, NISI INVENERIS, ET QUAERENDI DEFATIGATIO TURPIS EST, CUM ID, QUOD

QUAERITUR, SIT PULCHERRIMUM. ETENIM SI DELECTAMUR, CUM SCRIBIMUS, QUIS EST TAM INVIDUS, QUI AB EO NOS ABDUCAT? SIN LABORAMUS, QUIS EST, QUI ALIENAE MODUM STATUAT INDUSTRIAE? NAM UT TERENTIANUS CHREMES NON INHUMANUS, QUI NOVUM VICINUM NON VULT 'FODERE AUT ARARE AUT ALIQUID FERRE DENIQUE' -- NON ENIM ILLUM AB INDUSTRIA, SED AB INLIBERALI LABORE DETERRET --, SIC ISTI CURIOSI, QUOS OFFENDIT NOSTER MINIME NOBIS INIUCUNDUS LABOR.

Cycle time: 3

Client: Iis igitur est difficilius satis facere, qui se Latina scripta dicunt contemnere. in quibus hoc primum est in quo admirer, cur in gravissimis rebus non delectet eos sermo patrius, cum idem fabellas Latinas ad verbum e Graecis expressas non inviti legant. quis enim tam inimicus paene nomini Romano est, qui Ennii Medeam aut Antiopam Pacuvii spernat aut reiciat, quod se isdem Euripidis fabulis delectari dicat, Latinas litteras oderit?

Server: IIS IGITUR EST DIFFICILIUS SATIS FACERE, QUI SE LATINA SCRIPTA DICUNT CONTEMNERE. IN QUIBUS HOC PRIMUM EST IN QUO ADMIRER, CUR IN GRAVISSIMIS REBUS NON DELECTET EOS SERMO PATRIUS, CUM IDEM FABELLAS LATINAS AD VERBUM E GRAECIS EXPRESSAS NON INVITI LEGANT. QUIS ENIM TAM INIMICUS PAENE NOMINI ROMANO EST, QUI ENNII MEDEAM AUT ANTIOPAM PACUVII SPERNAT AUT REICIAT, QUOD SE ISDEM EURIPIDIS FABULIS DELECTARI DICAT, LATINAS LITTERAS ODERIT?

Cycle time: 3

Client: Synephebos ego, inquit, potius Caecilii aut Andriam Terentii quam utramque Menandri legam?

Server: SYNEPHEBOS EGO, INQUIT, POTIUS CAECILII AUT ANDRIAM TERENTII QUAM UTRAMQUE MENANDRI LEGAM?

Cycle time: 3

Client: A quibus tantum dissentio, ut, cum Sophocles vel optime scripserit Electram, tamen male conversam Atilii mihi legendam putem, de quo Lucilius: 'ferreum scriptorem', verum, opinor, scriptorem tamen, ut legendus sit. rudem enim esse omnino in nostris poetis aut inertissimae segnitiae est aut fastidii delicatissimi. mihi quidem nulli satis eruditi videntur, quibus nostra ignota sunt. an 'Utinam ne in nemore . . .' nihilo minus legimus quam hoc idem

Graecum, quae autem de bene beateque vivendo a Platone disputata sunt, haec explicari non placebit Latine?

Server: A QUIBUS TANTUM DISSENTIO, UT, CUM SOPHOCLES VEL OPTIME SCRIPSERIT ELECTRAM, TAMEN MALE CONVERSAM ATILII MIHI LEGENDAM PUTEM, DE QUO LUCILIUS: 'FERREUM SCRIPTOREM', VERUM, OPINOR, SCRIPTOREM TAMEN, UT LEGENDUS SIT. RUDEM ENIM ESSE OMNINO IN NOSTRIS POETIS AUT INERTISSIMAE SEGNITIAE EST AUT FASTIDII DELICATISSIMI. MIHI QUIDEM NULLI SATIS ERUDITI VIDENTUR, QUIBUS NOSTRA IGNOTA SUNT. AN 'UTINAM NE IN NEMORE . . .' NIHILO MINUS LEGIMUS QUAM HOC IDEM GRAECUM, QUAE AUTEM DE BENE BEATEQUE VIVENDO A PLATONE DISPUTATA SUNT, HAEC EXPLICARI NON PLACEBIT LATINE?

Cycle time: 3

Client: Quid? si nos non interpretum fungimur munere, sed tuemur ea, quae dicta sunt ab iis quos probamus, eisque nostrum iudicium et nostrum scribendi ordinem adiungimus, quid habent, cur Graeca anteponant iis, quae et splendide dicta sint neque sint conversa de Graecis? nam si dicent ab illis has res esse tractatas, ne ipsos quidem Graecos est cur tam multos legant, quam legendi sunt. quid enim est a Chrysippo praetermissum in Stoicis? legimus tamen Diogenem, Antipatrum, Mnesarchum, Panaetium, multos alios in primisque familiarem nostrum Posidonium. quid? Theophrastus mediocriterne delectat, cum tractat locos ab Aristotele ante tractatos? quid? Epicurei num desistunt de isdem, de quibus et ab Epicuro scriptum est et ab antiquis, ad arbitrium suum scribere? quodsi Graeci leguntur a Graecis isdem de rebus alia ratione compositis, quid est, cur nostri a nostris non legantur?

Server: QUID? SI NOS NON INTERPRETUM FUNGIMUR MUNERE, SED TUEMUR EA, QUAE DICTA SUNT AB IIS QUOS PROBAMUS, EISQUE NOSTRUM IUDICIUM ET NOSTRUM SCRIBENDI ORDINEM ADIUNGIMUS, QUID HABENT, CUR GRAECA ANTEPONANT IIS, QUAE ET SPLENDIDE DICTA SINT NEQUE SINT CONVERSA DE GRAECIS? NAM SI DICENT AB ILLIS HAS RES ESSE TRACTATAS, NE IPSOS QUIDEM GRAECOS EST CUR TAM MULTOS LEGANT, QUAM LEGENDI SUNT. QUID ENIM EST A CHRYSIPPO PRAETERMISSUM IN STOICIS? LEGIMUS TAMEN DIOGENEM, ANTIPATRUM, MNESARCHUM, PANAETIUM, MULTOS ALIOS IN PRIMISQUE FAMILIAREM NOSTRUM POSIDONIUM. QUID? THEOPHRASTUS MEDIOCRITERNE DELECTAT,

CUM TRACTAT LOCOS AB ARISTOTELE ANTE TRACTATOS? QUID? EPICUREI NUM DESISTUNT DE ISDEM, DE QUIBUS ET AB EPICURO SCRIPTUM EST ET AB ANTIQUIS, AD ARBITRIUM SUUM SCRIBERE? QUODSI GRAECI LEGUNTUR A GRAECIS ISDEM DE REBUS ALIA RATIONE COMPOSITIS, QUID EST, CUR NOSTRI A NOSTRIS NON LEGANTUR?

Cycle time: 3

Client: Quamquam, si plane sic verterem Platonem aut Aristotelem, ut verterunt nostri poetae fabulas, male, credo, mererer de meis civibus, si ad eorum cognitionem divina illa ingenia transferrem. sed id neque feci adhuc nec mihi tamen, ne faciam, interdictum puto. locos quidem quosdam, si videbitur, transferam, et maxime ab iis, quos modo nominavi, cum inciderit, ut id apte fieri possit, ut ab Homero Ennius, Afranius a Menandro solet. Nec vero, ut noster Lucilius, recusabo, quo minus omnes mea legant. utinam esset ille Persius, Scipio vero et Rutilius multo etiam magis, quorum ille iudicium reformidans Tarentinis ait se et Consentinis et Siculis scribere. facete is quidem, sicut alia; sed neque tam docti tum erant, ad quorum iudicium elaboraret, et sunt illius scripta leviora, ut urbanitas summa appareat, doctrina mediocris.

Server: QUAMQUAM, SI PLANE SIC VERTEREM PLATONEM AUT ARISTOTELEM, UT VERTERUNT NOSTRI POETAE FABULAS, MALE, CREDO, MERERER DE MEIS CIVIBUS, SI AD EORUM COGNITIONEM DIVINA ILLA INGENIA TRANSFERREM. SED ID NEQUE FECI ADHUC NEC MIHI TAMEN, NE FACIAM, INTERDICTUM PUTO. LOCOS QUIDEM QUOSDAM, SI VIDEBITUR, TRANSFERAM, ET MAXIME AB IIS, QUOS MODO NOMINAVI, CUM INCIDERIT, UT ID APTE FIERI POSSIT, UT AB HOMERO ENNIUS, AFRANIUS A MENANDRO SOLET. NEC VERO, UT NOSTER LUCILIUS, RECUSABO, QUO MINUS OMNES MEA LEGANT. UTINAM ESSET ILLE PERSIUS, SCIPIO VERO ET RUTILIUS MULTO ETIAM MAGIS, QUORUM ILLE IUDICIUM REFORMIDANS TARENTINIS AIT SE ET CONSENTINIS ET SICULIS SCRIBERE. FACETE IS QUIDEM, SICUT ALIA; SED NEQUE TAM DOCTI TUM ERANT, AD QUORUM IUDICIUM ELABORARET, ET SUNT ILLIUS SCRIPTA LEVIORA, UT URBANITAS SUMMA APPAREAT, DOCTRINA MEDIOCRIS.

Cycle time: 18

Client: Ego autem quem timeam lectorem, cum ad te ne Graecis quidem cedentem in philosophia audeam scribere? quamquam a te ipso id quidem facio provocatus gratissimo mihi libro, quem ad me de virtute misisti. Sed ex eo credo quibusdam usu venire; ut abhorreant a

Latinis, quod inciderint in inculta quaedam et horrida, de malis Graecis Latine scripta deterius. quibus ego assentior, dum modo de isdem rebus ne Graecos quidem legendos putent. res vero bonas verbis electis graviter ornateque dictas quis non legat? nisi qui se plane Graecum dici velit, ut a Scaevola est praetore salutatus Athenis Albucius.

Server: EGO AUTEM QUEM TIMEAM LECTOREM, CUM AD TE NE GRAECIS QUIDEM CEDENTEM IN PHILOSOPHIA AUDEAM SCRIBERE? QUAMQUAM A TE IPSO ID QUIDEM FACIO PROVOCATUS GRATISSIMO MIHI LIBRO, QUEM AD ME DE VIRTUTE MISISTI. SED EX EO CREDO QUIBUSDAM USU VENIRE; UT ABHORREANT A LATINIS, QUOD INCIDERINT IN INCULTA QUAEDAM ET HORRIDA, DE MALIS GRAECIS LATINE SCRIPTA DETERIUS. QUIBUS EGO ASSENTIOR, DUM MODO DE ISDEM REBUS NE GRAECOS QUIDEM LEGENDOS PUTENT. RES VERO BONAS VERBIS ELECTIS GRAVITER ORNATEQUE DICTAS QUIS NON LEGAT? NISI QUI SE PLANE GRAECUM DICI VELIT, UT A SCAEVOLA EST PRAETORE SALUTATUS ATHENIS ALBUCIUS.

Cycle time: 18

Client: Bye.

Server: Bye.

Stats {

Average Transmission In Size: 500 chars.

Average Transmission Out Size: 549 chars.
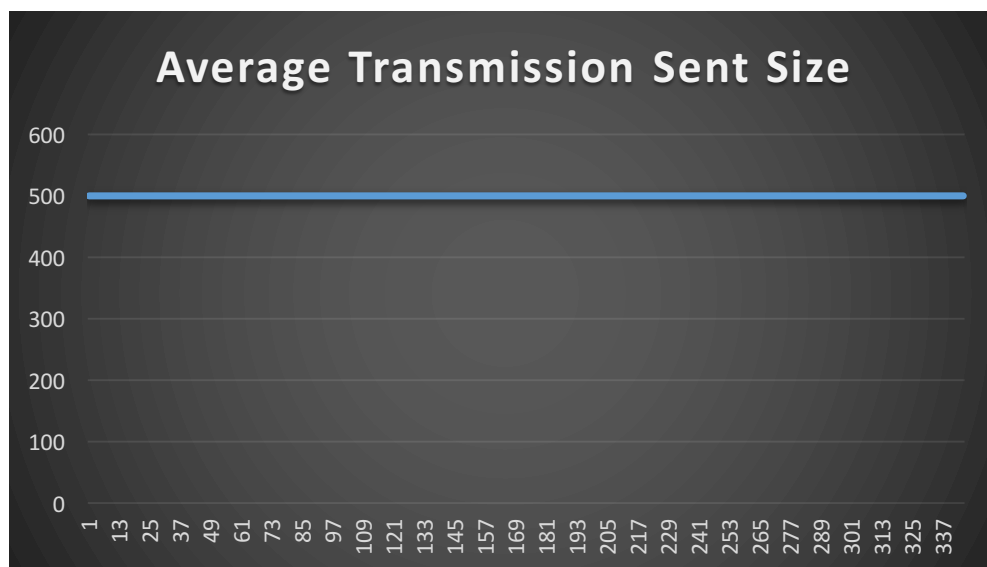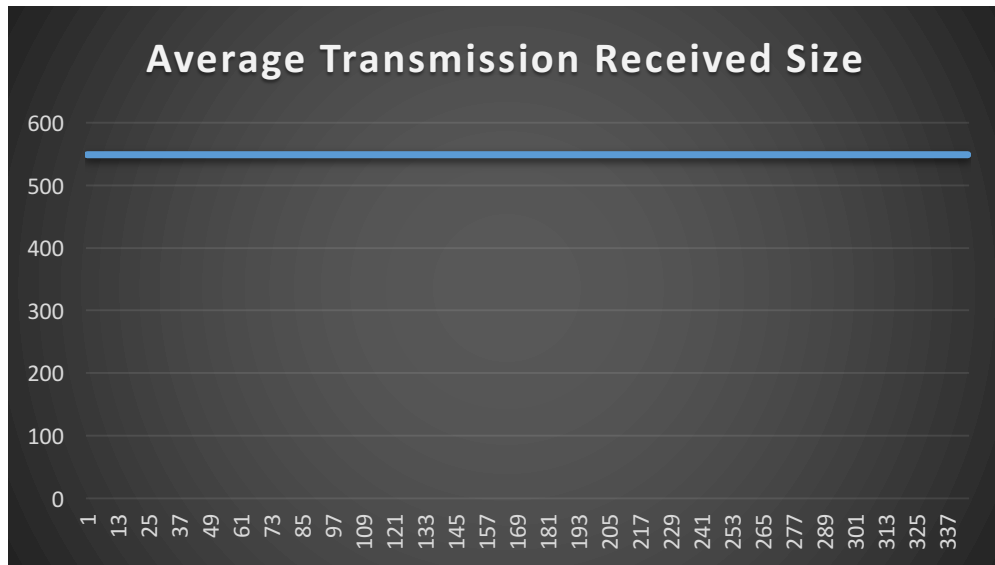
Average Transmission Time: 1013 ms.

}


Process finished with exit code 0

Collected Data:

There are three main statistics we focused on keeping track of for the first phase of our project. They are the Average Transmission Time, Average Transmission Sent, and Average Transmission Received.

For the data we collected to make our statistics, we kept everything in a more controlled environment and used the same input file. This is why the Average Transmission Sent and the Average Transmission Received are constant. The Average Transmission Sent stayed at 500 and the Average Transmission Received stayed at 549. We chose to include these two statistics because they can help explain possible differences in the average transmission times. For example, we would generally expect a larger transmission sent/received to take a longer amount of time.

Average Transmission Received Size

The Average Transmission Time fluctuated from 43 to 2012 milliseconds. Most of the transmissions averaged around 1000 milliseconds. There were a few spikes that reached upwards to 2012 milliseconds and a major dip that dropped to just 43 milliseconds. It's interesting to think about what could have cause these drastic changes in our data. Since transmission sizes were kept constant that eliminates some explanations. It could be network constraints, such as fluctuating speeds or interference. A loss of connectivity or attempting to reconnect to the network, could be a culprit as well. Hardware issues could be another possibility. For example, the CPU performance could be fluctuating caused by anything from multi-tasking to limitations being enforced to conserve battery life.

Average Transmission Time
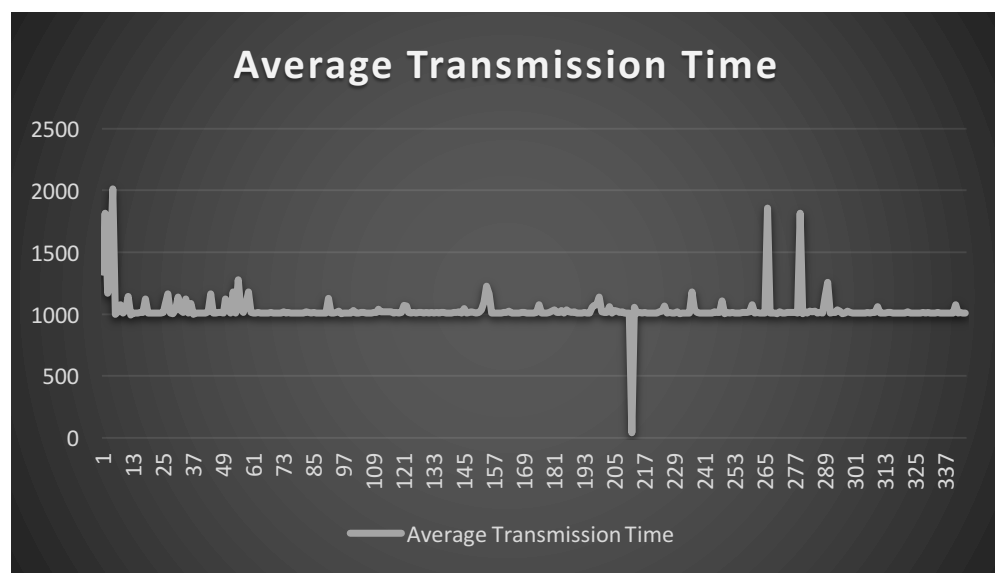
Conclusion:

       In Phase I of this project we implemented and analyzed a distributed system that uses TCP protocol to transfer data between a client and a server via a server router. Our output shows how the client and server establish a connection and communicate through the server router in order to send and receive files between each other. Our data shows how the transmission time between the client and server is affected based on the file that is sent through. The completion of this phase has provided us with more knowledge and a better understanding of the Client-Server paradigm.