DISTRIBUTED SYSTEMS

Distributed Systems – Assignment 1

Submitted: October 03, 2014

By: Devan Shah 100428864

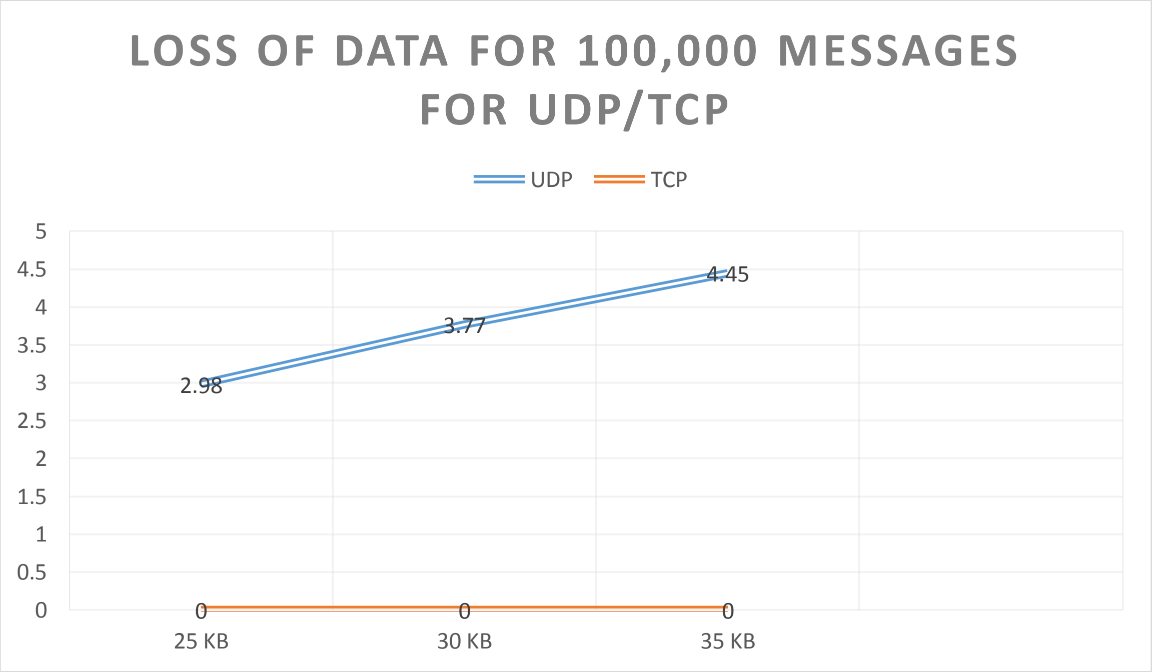
Submitted to: Weina ma & Ying Zu

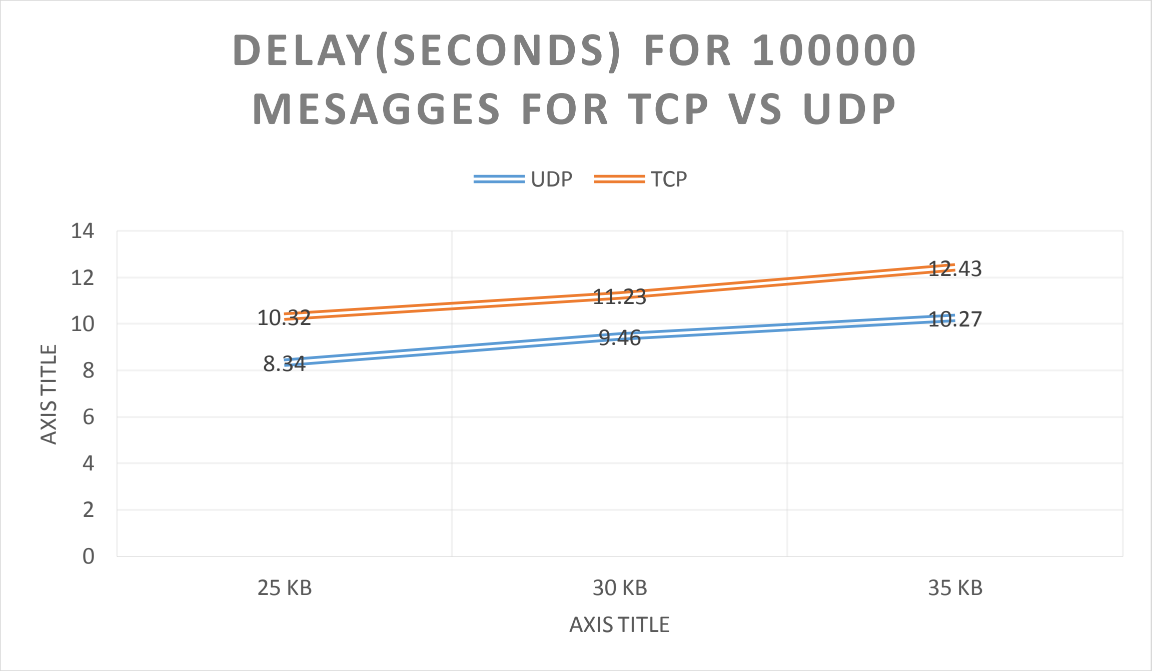
1. Read the use case study in section 1.6 in the text book on the Web and for each of the DS challenges listed below give a brief description of how that challenge is reflected in the Web case study. The challenges are: Heterogeneity, Openess, Security, Scalability, Failure Handling, Concurrency, Transparency, Quality of Service. [16]

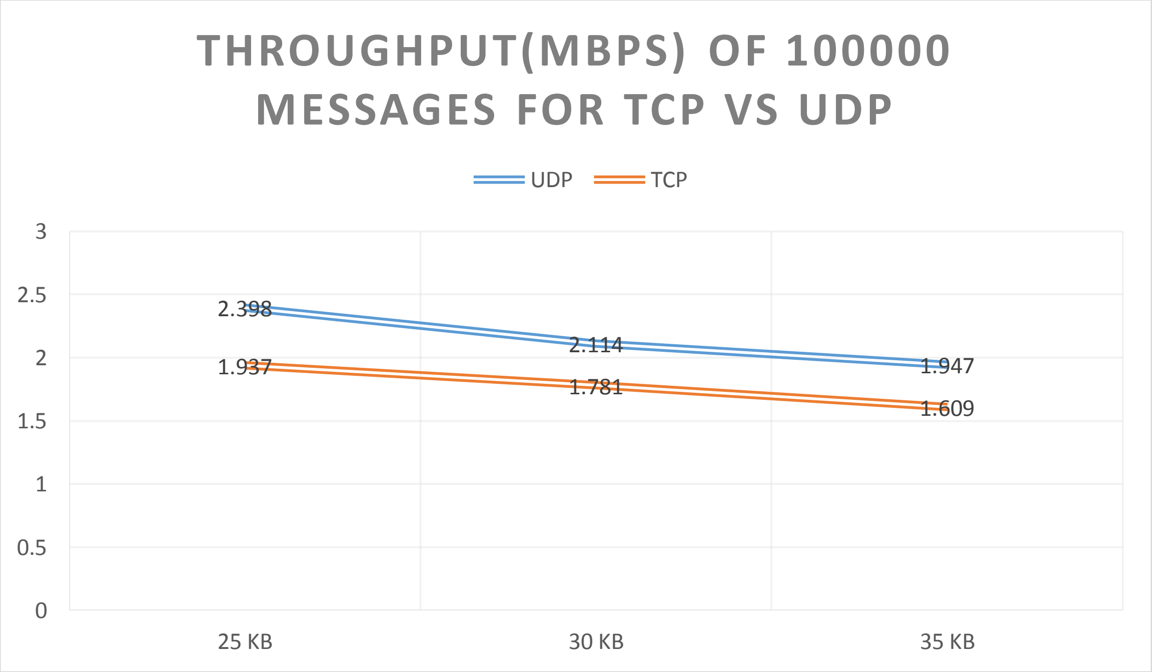
Distributed System Challenge	Brief challenge meaning	How the challenge is reflected in the World Wide Web case study?
Heterogeneity	Heterogeneity refers to making sure that the distributed systems is created with different types of computers and networks.	The World Wide Web is an enormous entity that is widely used in day to day activities. WWW is constructed with many different networks which include OS, computer hardware and programming languages. The users of the WWW would not be aware that WWW is comprised of so many different networks, all this information would be masked by the internet protocols and some would be dealt by the middleware.
Openness	Openness refers to having the distributed system being able to be extended in various ways, the ways that resources-sharing services can be added and made available for clients of the distributed system.	The internet has come out of many new services and has expanded over the years all of this is an open. The components can added to internet by simply publishing component, which would make it available to all users of the world wide web.
Security	Security refers to having the information and resources that are made available and maintained in a distributed system having a high core value of their users.	WWW uses https as a secure way to use World Wide Web, this allows for greater security for the users to keep the sensitive information encrypted and not available for distribution.
Scalability	A distributed system needs to be able to be scalable if there is a significant increase in the number of resources and number of users.	The cost of adding resources to the internet is constant in terms of the resources that must be added. The amount of users that use the internet increase rapidly over the years and the internet is able to handle and accommodate the resources for all old and new users with little or no performance issues.
Failure Handling	When faults occurs in the hardware or the software, programs that are producing wrong results. Sometimes having the computer system fail is an issue.	Since the internet uses a wide verity of different networks there can be an issues anywhere in this process. For example the computer or network may fail independent of the others. So all components need to handle all possible failures that can be causes from other components that it communicates with to handle such errors.
Concurrency	Distributed systems can have the possibility where the resources can be used by the multiple users at the same time. The system needs to handle this for multiple users.	All the web pages on the internet are capable to be accessed from multiple environments at the same time. The request of the resources can be called upon from a lot of users at once with on issues.
Transparency	Transparency is classified in distributed systems is by having the independent parts concealed from the user and the applications programmers.	The internet is an extremely complex and has multiple moving parts so not all aspects need to be visible to the end users that access the internet. For example, the users of the internet do not need to know about the

		locations of all the computers and networks that make up the internet.
Quality of Service	Quality of service refers to retrieving information from the users of the services and ask about the quality of the service that is being provided by this distributed system.	With relation to the internet the quality of service can be measured based on performance, security and reliability.

2. Modify the UDP and TCP programs included for the assignment so that you can compare the performance of TCP and UDP with regards to latency, packet loss, and throughput. For the tests you can try have the Client send 100000 messages to server with the size of 25, 30, 35 kilobytes respectively and plot the delay ,loss of data, and throughput for both the UDP and TCP connections. [15]







These graphs represent the calculations performed with the data that was received from the programs. All the source code of the program is available in the submission zip file, under the folder name for this question. (Distributed Systems – Assignment 1 Question 2). Instructions to compile are also available in the readme file.

3. Describe the serialized form that a serialization algorithm would produce when serializing an instance of the following class [5]

```
Class Couple implements Serializable {
        private Person one;
        private Person two;
        public Couple(Person a, Person b) {
           one = a;
           two = b;
        }
}
```

With provided sample instance of Couple below:

```
        Couple coupleSample = new Couple ( new Person (“Devan”, “Vaughan”, 1992 ),
                                           new Person ( “Bob”, “Markham”, 2014 )
                                          ) ;
```

The output of the serialized object would be the following:

Table 1 Serialized Values in a Tabular form

Couple (class)	8 byte version number			handler0
2	Person one		Person two	
Person (class)	8 byte version number			handler1
3	int year	String name	String place	
	1992	5 Devan	7 Vaughan	handler2
	2014	3 Bob	7 Markham	handler3

This table shows the Class names and the data that is serialized in each of the cases.

4. Write a Java client application that creates an instance of Couple and instantiates one Person. It then sends this serialized instance of Couple to a server. The server adds a second Person to Couple and returns the value back to the client. In order to validate your client/server application print out the values of Couple before sending it to the server, after the server received it and when the client received the modified value of Couple from the server. [10]

Client is creating Couple with one person like so:

```
        Couple c  = new Couple ( new Person ( "Bob", "Markham", 2014 ), null ) ;
```

Results from the client before sending data:

```
(Couple) Sent:
  Person one
   Name: Bob
   Place: Markham
   Year: 2014
```

Results from the server after client has send the data and server has received the data:

```
(Couple) Server Received:

  Person one
   Name: Bob
   Place: Markham
   Year: 2014

  Person two
   Name: NULL
   Place: NULL
   Year: NULL
```

This last person is null because client is sending null person 2, and the server is adding the Person 2.

Results from the client after the second person is added by the server and the data is sent back:

```
(Couple) Received:

  Person one
   Name: Bob
   Place: Markham
   Year: 2014

  Person Two
   Name: Sandy
   Place: Vaughan
   Year: 1992
```

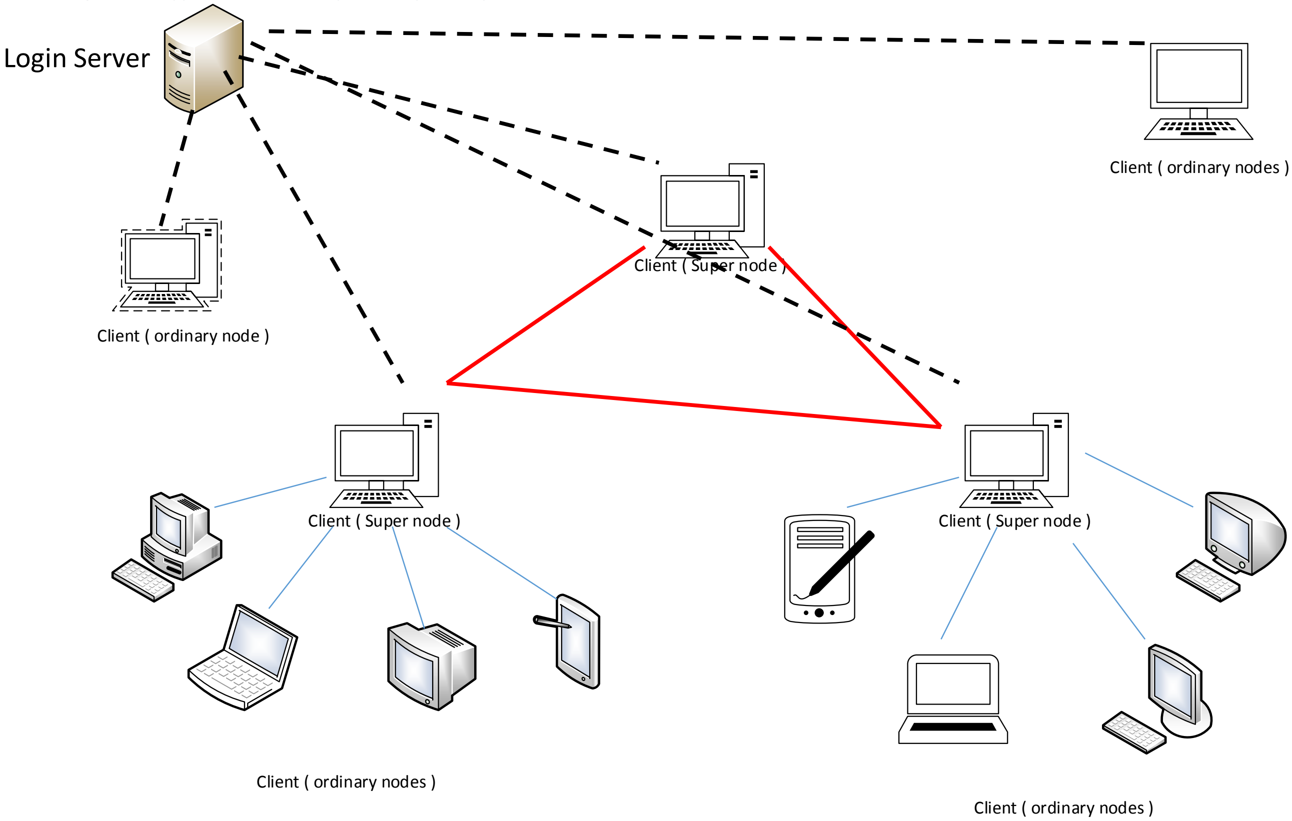
All the source code of the program is available in the submission zip file, under the folder name for this question. (Distributed Systems – Assignment 1 Question 4). Instructions to compile are also available in the readme file.

5. Submit a short 1-page essay describing the overlay architecture of Skype. Describe how nodes are addressed; how connections are established and users are discovered. Show an architecture diagram where possible. Use the paper on Skype provided in the course folder. [10]

Skype is a widely used application that is equivalent other applications like MSN, Yahoo IM and Lotus Notes. All these applications are classified as distributed systems as they communicate with multi users and are made up of many networks. Skype is one of the few applications that support voice-over-IP service and instance messaging, this is a very handy combinations for an application to have. Skype was developed by Microsoft and it is currently one of the most used instant messaging client in the world. Skype allows users to connect to each other across countries by the use of microphone, webcam, voice-over-IP or instance messaging. To perform this action Skype uses a hybrid peer-to-peer system that involves a client-server relationship. The way that Skype works it now publically known as it is an extremely complex system that provides many functions.

The main architecture that Skype follows is outlined in the Skype paper thoroughly, Skype makes use of a hybrid overlay peer-to-peer network design with the use of structured nodes with client and server communications for login credentials. The Skype peer-to-peer network contains two types of nodes, which include ordinary nodes and super nodes. The responsibility of an ordinary node is to simply make voice calls and send instance messages and any other functions that core Skype application has. However a super node is used by the ordinary node to perform authentication with the Skype login server. The super node acts as an end-point for the ordinary node as the super node is the primary communication for Skype. This is not all ways the case, this is only the case when the public IP address, sufficient memory, sufficient CUP capacity and minimal bandwidth. The ordinary nodes that are not part of a super node would communicate with the login server directly and the ones that are part of a super node would communicate with the super node and then the super node would communicate with the Skype login servers. The main communications that are perform between the ordinary nodes to Skype server and ordinary nodes to super node to Skype server are authentication purposes and buddy list. Also the main host cache (HC) is stored on the Skype server which needs to be retrieved by all ordinary nodes in the Skype network, this is the master list of active Skype users. This list allows for communication with other Skype users, add new users and delete inactive users.

Figure 1 Skype Architecture peer-to-peer layover network



The above image gives a quick overview of how the peer-to-peer Skype network works. From the image the dash lines represents the connection between the server and ordinary nodes and/or server to super node. The red line represents the neighbouring connection of super nodes. The blue line represents the connection between the super node and the ordinary nodes in the Skype network.

Each of the Skype clients contain/fetch the host cache file which includes the list of all reachable super nodes' IP addresses and port numbers. The list is generated by the Skype server, the list only contains list of users that have been active in the last 72 hours. The main communications are performed using TCP during a login attempt, since there needs to be a communication of secure data and also need to make sure there is a connection oriented transaction.

In Conclusion, Skype peer-to-peer network is a great design to have such a stable version of the application that is cable of being used distance. The connection is fast and reliable as there are not a lot of dropped packets during connections and sending data over skype.