**PROJECT REPORT-- Distributed Systems COMP90015 2014 SM1 Project 2**

|  |  |  |
| --- | --- | --- |
| Student Number/Name: |  | Hao Zheng (594644) |
| Student Number/Name: |  |  |
| Subject Code/Name: |  | Distributed SystemsCOMP90015 |
| Assignment Title: |  | Project 2 – SimpleStreamer Webcam Image Streaming |
| Due Date/Time: |  | 30/5/2014 |

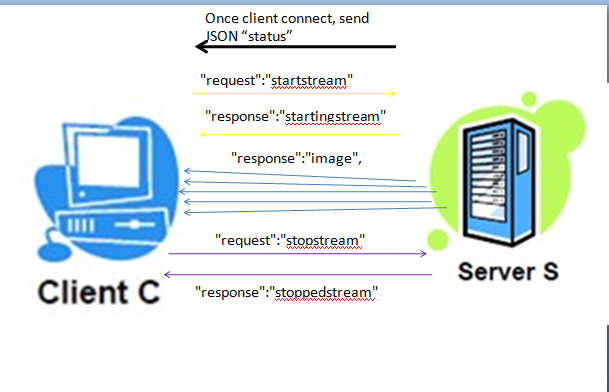
****

Figure 1-1. The connecting and transmitting mechanism of the system

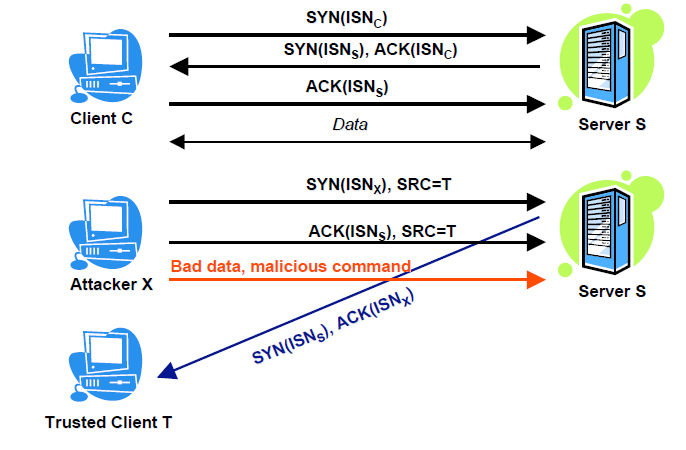


Figure 1-2. Attacker affects the TCP connection.

The project 2 requires building a complex distributed system – a streaming application that can stream webcam images. This system has two models, the remote model and the local model. In the remote model, the streaming application will connect to another system and receive images from that application. In the local model, it will receive images from local default webcam.

Figure 1-1 shows the connecting and transmitting mechanism between the clients and the server. Once the clients connect to the server, the server will send the stream of “status” with JSON format to the clients. The “startatream” request is asking the server to start sending a stream of images, when the server receive the request, it will send the acknowledge stream back to the clients to indicate the server has received the request. The processing of transmit images will not stop until the client send the “stopcream” order which is controlled by the user with the keyboard. When the server has noticed the request, it will send a JSON message of “stoppedstream” back to the client and shut down the connection. The server will close the TCP connection after sending this response and the client will close the TCP connection after receiving this response.

As it is known to all, a low-bandwidth client in TCP will be unable to receive data quickly, this may cause the stream blocking on the server side. Bandwidth refers to the capacity of carrying data or transmitting data. Bandwidths show the max amount of data that pass between the client and the server; it means how much data can be sent down the network. The bandwidth of a TCP connection is related to congestion, socket and size. This is either due to TCP itself (e.g., its congestion control algorithms and parameters), or because of local system configuration (e.g., default TCP socket buffer size). TCP does not perform well in high-performance networks, especially in long-distance transfers. The first reason would be packet loss. It causes multiplicative window reduction and an overall lower average throughput. The second reason would be limited socket buffers. TCP clients and servers would have a socket buffer during the data processing. As a result, if TCP wants to reduce the bandwidth, it can reduce the packet losses and increase the socket buffers.

Figure 1-2 exactly shows the way that an attacker will affect the TCP connection. On one hand, they will pretend the clients and send interrupt data to the server to affect the connection. On the other hand, they may send a huge amount of data to the server to interrupt the server, due to the limit of the buffer, the server cannot send the acknowledge data back to the client. As a result, TCP connection has potential security issues. TCP is designed for the public network. If the security of TCP is failure, it could expose the local network to attack originating from the public networks. In this project, not only the system can get images from local webcam, but also it can receive images from other distributed system, so there is a local and public security issue. In addition, TCP does not provide an encryption method. Therefore, all the data transmit through the TCP cannot be encrypted is another security issue. Due to it is particularly important to use encryption and authentication protocols when appropriate. SSL would be an answer to this problem. SSL supports server authentication and encryption by default, as a result, the server can make sure the clients are authenticated. Another option solution can be the SSH protocol. This allows an encrypted and (normally) both-side authenticated connection, over which one can route multiple channels. This is usually used for remote command execution or file transfer, which is suitable for this stream project. SSL is really well supported. Based on Public Key Infrastructure, it offers a variety of encryption and authentication schemes.

In this system, when sending images to a client, the server will by default, sleep for 100ms in between each image sent. In other words, the images will be sent at a rate no more than10 per second. Rate limit is trying to prevent the server from being flooding by these images. If let the thread sleep for a regular period, it can make sure that the server keeps the data consistency and accuracy. Scalable TCP modifies the congestion control. Effectively, this process keeps having the throughput until packet loss happens. If the packet loss reduces, the speed will goes back slowly and the spending time is called recovery time. When the rate limit is small, like 100ms, this recovery time is also small. However, if the transfer speed is quite fast, the recovery time may be more than an hour. As a result, because of the implementation of the rate limit, it achieves the TCP scalability.

Handover in this project means the server will only serve up to 3 clients concurrently. This is a crude attempt to ensure the server is not overloaded. When a client starts a TCP connection to a server, the client typically connects to a specific port on the server and requests that the server response to the client. If the number of clients continues to enhance, the only server might be overloaded because it cannot handle too many clients request at a time. Distributed applications often require data sharing with multiple remote client applications. In this project, the handover indicates that the limit of the client connection, which both has advantages and disadvantages. The advantages are avoiding blocking, multiple streams with different properties, error control and reasonable bandwidth. The blocking issue can be solved especially when the multiple TCP connections are between a pair of hosts. Multiple connections allow for multiple simultaneous transmits, in this project, is render images between client and server, which improve reliability and speed. In terms of multiple transmissions, if one thread error, it may not affect other connection, and this makes the network robust to failures and churn in participants. The disadvantages are low use of memory space and low utilization. We limit the number of clients that connect to the server; however, the TCP connection may handle more clients’ request at a time, results in the low utilization.

There can be further improvement on this project. Firstly, we can find the balance of handover and rate limiting. More specifically, server should try to connect more clients under the condition of not being over-loaded. Secondly, we can improve this system by connecting a non-SQL database with the system, so that images with the data type of JSON can be stored in the non-SQL database. Thus we can store more images rather than only the latest image at a time.