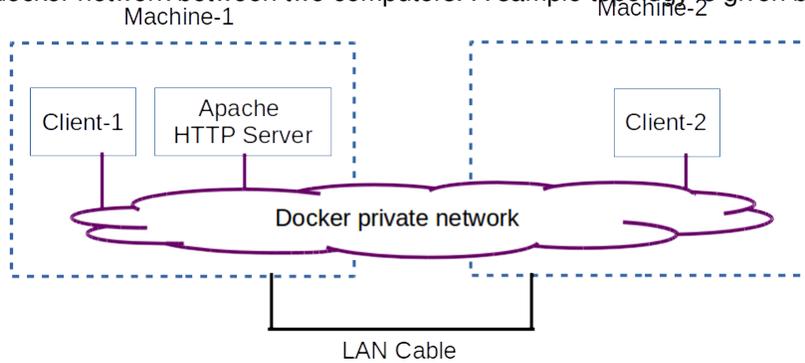


- Survey Service** Implement the *Survey Service* client and server described in section 8.3 of the following book. Michael J. Donahoo and Kenneth L. Calvert, TCP/ IP Sockets in C, Second Edition, Morgan Kaufman, 2009. Implement the service in any programming language other than C / C++. In order to receive credit for this question, your client must successfully interact with the given C-server and your server must interact successfully with the given C-client. **(10M)**
- Implement a private docker network between two computers. A sample topology is given below.



The deliverables are:

- Docker files to spawn Apache HTTP Server, Client-1 and Client-2. **(5M)**
 - Access of content on web server from the two clients. **(5M)**
 - Private network configuration in docker such that the server is not accessible directly to programs running outside docker containers. **(5M)**
- UDP-based data transfer (UDT) protocol**

UDT is a reliable UDP based application level data transport protocol for distributed data intensive applications over wide area high-speed networks. UDT uses UDP to transfer bulk data with its own reliability control and congestion control mechanisms. The new protocol can transfer data at a much higher speed than TCP does. UDT is also a highly configurable framework that can accommodate various congestion control algorithms.

More information on UDT is available in the following paper:

Yunhong Gu and Robert L. Grossman, UDT: UDP-based Data Transfer for High-Speed Wide Area Networks, Computer Networks (Elsevier). Volume 51, Issue 7. May 2007.

You need to write a complementary implementation of UDT that communicates with any of the UDT implementations listed on <http://udt.sourceforge.net/software.html> page. A thorough documentation of the protocol is available at: <http://udt.sourceforge.net/udt4/index.htm>.

The deliverables are:

- Implementation of skeleton UDT software architecture given in Fig. 4 of the paper. **(5M)**
- Implementation of skeleton APIs - accept(), bind(), close(), connect(), recv(), startup(), sendfile() and recvfile(). **(5M)**
- Successful connection setup and release using UDT packet structure. **(5M)**
- Successful file transfer using any congestion control algorithm. **(10M)**

Evaluation Schedule

Date	Deliverables
14-April-2018	Questions - 1 and 2, review of progress on 3(a) and 3(b)
28-April-2018	Questions - 3