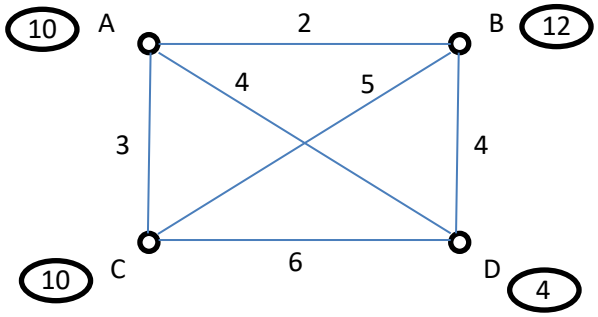
Replica Management (10 points)

This homework covers the concepts regarding the replication within distributed systems. We focus on server-initiated content replicas in this homework.

The homework must be uploaded until Monday, **21.05.2018**, at **08:00**, and will be evaluated on 22th May. The tutorial with hints will be taken on Tuesday, 15.05.2018.

Assignments:

Develop a simulator (a program, not a theoretical solution) that will calculate the number and optimal placement of content replicas for a given set of requests. Use such replica placement that minimizes the weight function for used storage and latency *TotalCost* = *StorageCost* * *w1* + *LatencyCost* * *w2,* where *w*1 + *w*2 = 1. Consider the following topology:



It is a full mesh network of four data centres *A*, *B*, *C* and *D*, each located on different continent. The link between two data centers indicates the latency of the connections. Instead of given numbers for latency, use digits of your Matrikelnummer (use latency 10 instead of digit 0). A number in a circle near each data center denotes how many clients are located there. Assume that the latency between a client and corresponding data centre is 1. The cost for a single replica is 30.

Input:

- The number of requests from each data center (use random number 0-15)

Evaluate your program for five random set of inputs (e.g. A with 5 requests, B 3, C 2, D 0) for each combination of the three weight parameters (a total of 15 evaluations):

- Prefer storage cost (w1 = 0.75, w2 = 0.25)
- Prefer latency cost (w1 = 0.25, w2 = 0.75)
- Prefer equally storage cost and latency cost (w1 = w2 = 0.5)

Output:

- StorageCost and LatencyCost
- Optimal TotalCost
- Replica placement. E.g. AC.

Use these assumptions to simplify the algorithm:

1. The RIP routing protocol is used.
2. The bandwidth of each link is infinite.

***For brainstorming, how complex the life of distributed system is! If interested, could be defined a bachelor thesis in some of these extensions.

Think if you can model a system with arbitrary number of clients *N* per continent (node), arbitrary number (could be 0, as well, that is, no path exists) of path(s) (with arbitrary weights) from clients to each data center. Further on, each link is limited how many clients' requests can be sent through it (link capacity). Where to store *K*<*N* replicas?

Let's add a fault tolerance. Each link can be down with some probability. Each data center, as well. Place now K replicas in order to achieve *a*% availability and r% reliability. Assume to have at least two different paths to each content in order to guarantee minimum bandwidth for each request.

What if clients are moving? ☺ that is, the number of clients per data center (continent) is changing? When to replicate, when to migrate some file?

Data center capacity. That is, we assume in the homework a single content. In reality, many users require access to different contents. Each replica has some size and storing replicas in a data center is limited.

Different time zone: Add a probability that a client will access a file due to time zones.

All these issues were based on the assumption that we manage contents with a constant size. What if we consider a distributed database? That is, to achieve consistency and synchronization.