Aleks Murauskas

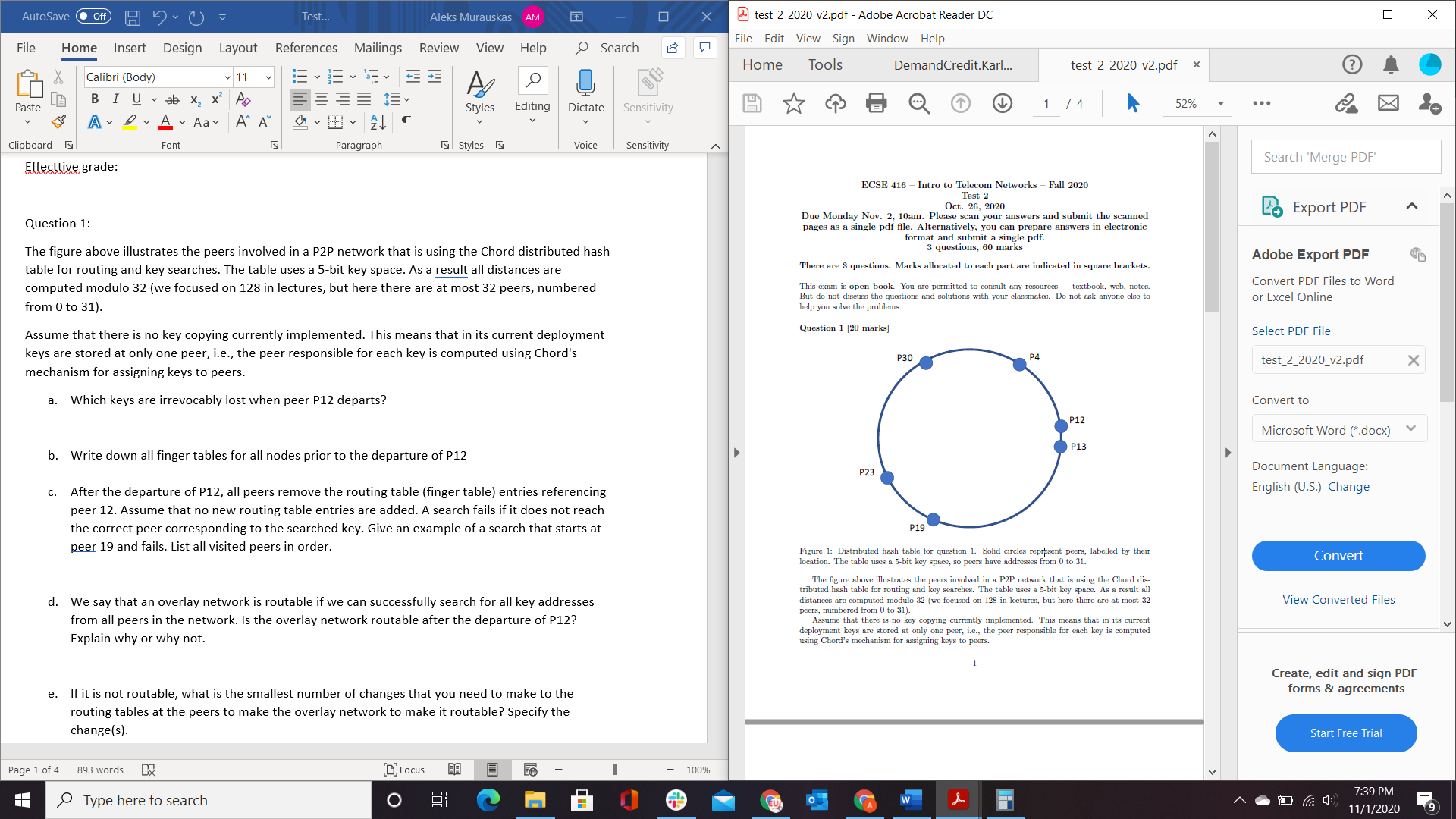
260718389

ESCE 416

Question 1:

The figure above illustrates the peers involved in a P2P network that is using the Chord distributed hash table for routing and key searches. The table uses a 5-bit key space. As a result all distances are computed modulo 32 (we focused on 128 in lectures, but here there are at most 32 peers, numbered from 0 to 31).

Assume that there is no key copying currently implemented. This means that in its current deployment keys are stored at only one peer, i.e., the peer responsible for each key is computed using Chord's mechanism for assigning keys to peers.



1. Which keys are irrevocably lost when peer P12 departs?

As each node is responsible for keys having lower than its key value and greater than key of the previous node, All keys between (P4,P12] are lost. Therefore, P5,P6,P7,P8,P9,P10,P11, and P12’s keys are lost.

1. Write down all finger tables for all nodes prior to the departure of P12

32 Nodes need support so

For P4: n=4

|  |  |  |
| --- | --- | --- |
| i | Succ | Next available Node |
| 1 | 5 | 12 |
| 2 | 6 | 12 |
| 3 | 8 | 12 |
| 4 | 12 | 12 |
| 5 | 20 | 23 |

For P12: n=12

|  |  |  |
| --- | --- | --- |
| i | Succ | Next available Node |
| 1 | 13 | 13 |
| 2 | 14 | 19 |
| 3 | 16 | 19 |
| 4 | 20 | 23 |
| 5 | 28 | 30 |

For P13: n=13

|  |  |  |
| --- | --- | --- |
| i | Succ | Next available Node |
| 1 | 14 | 19 |
| 2 | 15 | 19 |
| 3 | 17 | 19 |
| 4 | 21 | 23 |
| 5 | 29 | 30 |

For P19: n=19

|  |  |  |
| --- | --- | --- |
| i | Succ | Next available Node |
| 1 | 20 | 23 |
| 2 | 21 | 23 |
| 3 | 23 | 23 |
| 4 | 27 | 30 |
| 5 | 3 | 4 |

For P23: n=23

|  |  |  |
| --- | --- | --- |
| i | Succ | Next available Node |
| 1 | 24 | 30 |
| 2 | 25 | 30 |
| 3 | 27 | 30 |
| 4 | 31 | 4 |
| 5 | 7 | 12 |

For P30: n=30

|  |  |  |
| --- | --- | --- |
| i | Succ | Next available Node |
| 1 | 31 | 4 |
| 2 | 0 | 4 |
| 3 | 2 | 4 |
| 4 | 6 | 12 |
| 5 | 14 | 19 |

1. After the departure of P12, all peers remove the routing table (finger table) entries referencing peer 12. Assume that no new routing table entries are added. A search fails if it does not reach the correct peer corresponding to the searched key. Give an example of a search that starts at peer 19 and fails. List all visited peers in order.

Affected Peers:

Peer 12 Table entirely removed.

For P4: n=4

|  |  |  |
| --- | --- | --- |
| i | Succ | Next available Node |
| 1 |  |  |
| 2 |  |  |
| 3 |  |  |
| 4 |  |  |
| 5 | 20 | 23 |

For P23: n=23

|  |  |  |
| --- | --- | --- |
| i | Succ | Next available Node |
| 1 | 24 | 30 |
| 2 | 25 | 30 |
| 3 | 27 | 30 |
| 4 | 31 | 4 |
| 5 |  |  |

For P30: n=30

|  |  |  |
| --- | --- | --- |
| i | Succ | Next available Node |
| 1 | 31 | 4 |
| 2 | 0 | 4 |
| 3 | 2 | 4 |
| 4 |  |  |
| 5 | 14 | 19 |

All other tables remain the same.

Step 1: We begin at Peer 19

Step 2: Peer 19 seeks to find the closest value to Peer 16

Step 3: Peer 16 should be within Modulo 3 and 20. The smaller value is 3, so we hop to peer 4

Step 4: peer 4 was heavily affected by the removal of Peer 12, only has one successor at 20

Step 5: Since the only successor available is 20, which is higher than the target Peer 16, The search has failed.

1. We say that an overlay network is routable if we can successfully search for all key addresses from all peers in the network. Is the overlay network routable after the departure of P12? Explain why or why not.

No, the Overlay network is not routable. Due to the changes in the finger tables shown in 1.c, and the keys lost (4,12], and the example given 1.c.

Since Peer 16 can no longer be successfully searched for, the overlay network is not routable after the departure of P12.

1. If it is not routable, what is the smallest number of changes that you need to make to the routing tables at the peers to make the overlay network to make it routable? Specify the change(s).

Another Peer could enter the network and made Peer 12’s replacement, or another peer already within the network could be store a hash table of keys.

Let Peer 11 replace Peer 12:

First Peer 11 should notify all peers that hold hash tables It is now a hash table holder.

Then Peer 11 develops it’s own Hash table for routing.

All table holders repeat their hash functions to update their hash tables.

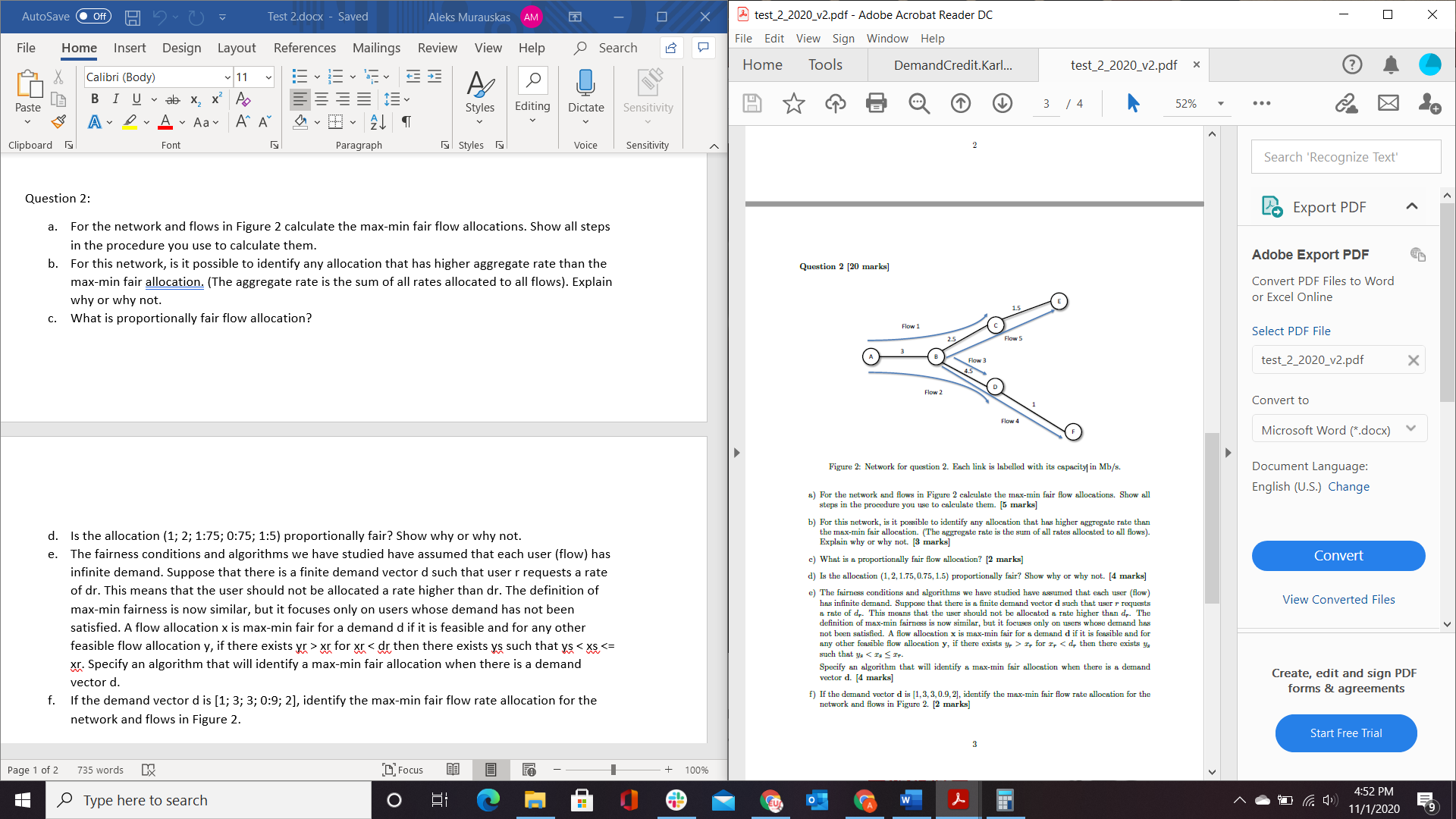
All Peers in the network are notified that Peer 11 is now a hash table holder.

1. Suppose that peer P19 discovers that its successor link is no longer consistent, because peer P23 has updated its predecessor link. What must have happened in order for this to occur?

A new node must have joined the network between 19 and 23.

1. Before peer P12 has departed, what is the expected number of hops required for a query to be resolved, if the query starts uniformly at random from one of the six peers and targets a key that is selected uniformly at random from the range [0,31]?

Question 2:



1. For the network and flows in Figure 2 calculate the max-min fair flow allocations. Show all steps in the procedure you use to calculate them.

We Have 5 total flows. 1,2,3,4,5. We will use the water filling procedure

Step 1: Fill all flows until we hit a capacity: Flow 4 hits the capacity 1 at D-F. [x,x,x,1,x]

Step 2: Continue to fill 1,2,3,5 until next flow is reached. Flow 5 reaches capacity 1.5 at C-F [x,x,x,1,1.5]

Step 3: However, Step 2 has met the capacity constraint 2.5 of B-C since Flow 5=Flow 1 as they should increase equally. And flow 1 and 5 being 1.5 would increase the flow beyond that capacity. 2.5 equally across both flows is 1.25 [1.25,x,x,1,1.25]

Step 4: By the capacity constraint of 3 at A-B. C(A-B)>=Flow 1 +Flow 2, since flow 1=1.25. Flow 2 = 3-1.25=1.75. Therefore [1.25,1.75,x,1,1.25]

Step 5: Since Flow 2= Flow 3, the final flow allocation vector is [1.25,1.75,1.75,1,1.25]

1. For this network, is it possible to identify any allocation that has higher aggregate rate than the max-min fair allocation. (The aggregate rate is the sum of all rates allocated to all flows). Explain why or why not.

The aggregate rate found is 7. There is no other higher aggregate, because any proportional change is non positive. Proportional fairness maximizes session utility.

1. What is proportionally fair flow allocation?

I am Interpreting this question in the form that it is asking for the definition of proportionally fair flow allocation.

A flow rate allocation x is proportionally fair if it is feasible and for any other allocation y the following equation is true.

1. Is the allocation (1, 2, 1.75, 0.75, 1.5) proportionally fair? Show why or why not.

Let (1, 2, 1.75, 0.75, 1.5) =X, Let Y be the allocation found by the water filling algo in 2a. [1.25,1.75,1.75,1,1.25].

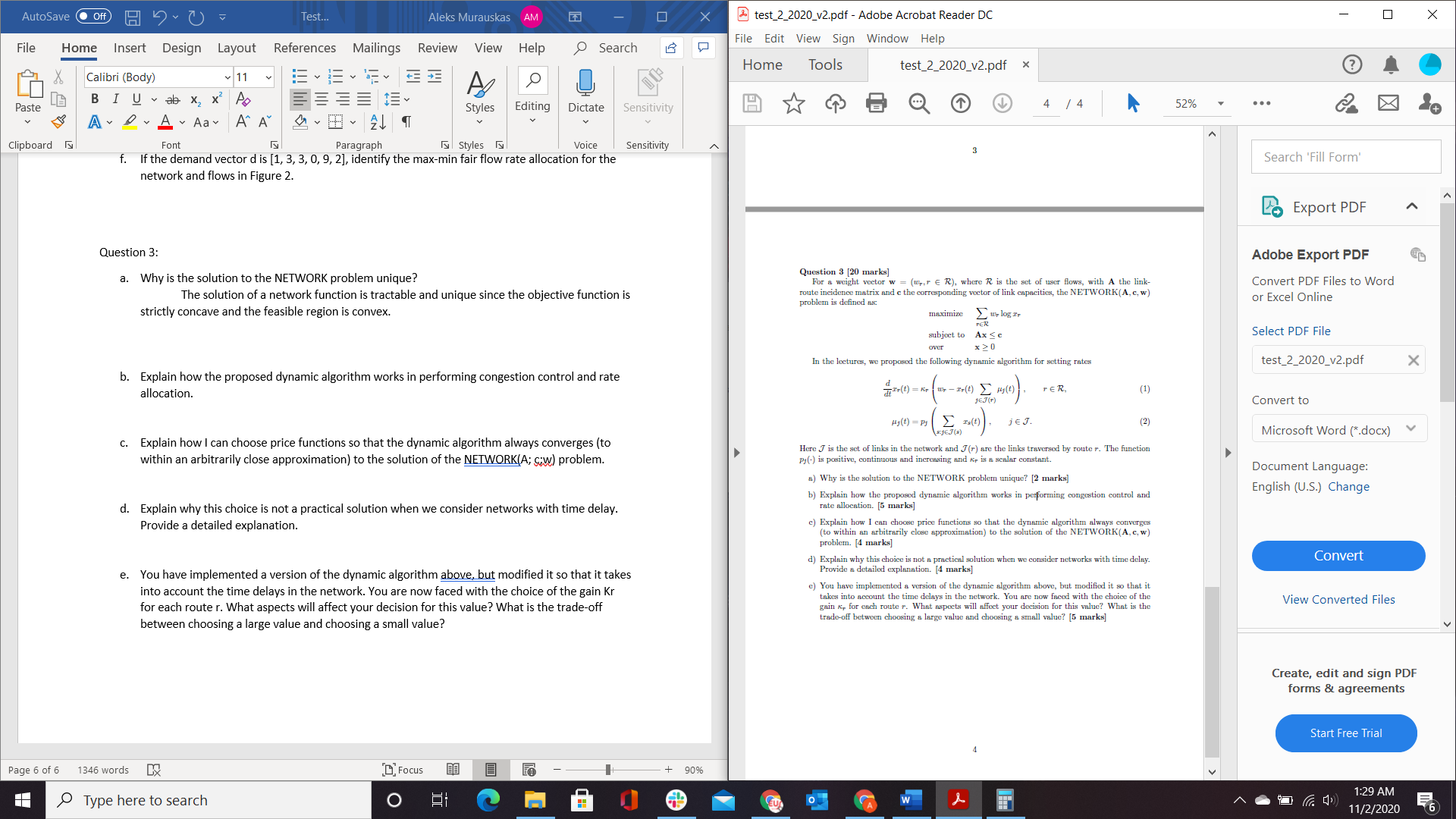
1. The fairness conditions and algorithms we have studied have assumed that each user (flow) has infinite demand. Suppose that there is a finite demand vector d such that user r requests a rate of dr. This means that the user should not be allocated a rate higher than dr. The definition of max-min fairness is now similar, but it focuses only on users whose demand has not been satisfied. A flow allocation x is max-min fair for a demand d if it is feasible and for any other feasible flow allocation y, if there exists yr > xr for xr < dr then there exists ys such that ys < xs <= xr. Specify an algorithm that will identify a max-min fair allocation when there is a demand vector d.

Known max min algorithm:

Use weighted proportional fairness, those who demand more should receive it.

1. If the demand vector d is [1, 3, 3, 0.9, 2], identify the max-min fair flow rate allocation for the network and flows in Figure 2.

Question 3:



1. Why is the solution to the NETWORK problem unique?

In order for the solution of a network problem to be unique, the objective function must be concave and optimized over a convex set.

The Function Log(x) is concave. The summation of multiple concave functions remains concave. Therefore the objective function is concave.

Therefore, The solution of a network function is unique since the objective function is strictly concave and the feasible region is convex.

1. Explain how the proposed dynamic algorithm works in performing congestion control and rate allocation.
2. Explain how I can choose price functions so that the dynamic algorithm always converges (to within an arbitrarily close approximation) to the solution of the NETWORK(A; c;w) problem.

As established previously p\_j is our price function. An ideal price function would be a step-function that rapidly jumps from zero to a significantly large value. However, step functions are not differentiable as they are not continuous. Our equations require that the price function de differentiable in order to detect rate of change of flow rates. Therefore our equation must remain close to zero for a significant amount of values before having an incredibly steep slope.

1. Explain why this choice is not a practical solution when we consider networks with time delay. Provide a detailed explanation.
2. You have implemented a version of the dynamic algorithm above, but modified it so that it takes into account the time delays in the network. You are now faced with the choice of the gain Kr for each route r. What aspects will affect your decision for this value? What is the trade-off between choosing a large value and choosing a small value?

The choice of gain is dependent on if the connection has yet found the equilibrium point.

Since K\_r represents a scalar value. Now that we can determine the scalar value at a router. Since the value of K\_r is directly multiplying the rate of change of flow rates. The larger the value of K\_r, the the rate of change will increase faster. Since the price function is dependent on x(t) approaching a value, as it is approximating a jump function, the frequency of the oscillations will increase. Inversely if the Scalar value is lower, x(t) has a lower rate of change and the frequency of oscillations is reduced.

A larger scalar should be used when the equilibrium point is not known as the increased frequency of oscillations will find the equilibrium point sooner. A lower scalar value would be used when the equilibrium point is found to stay as close to the equilibrium point for the maximum amount of time.