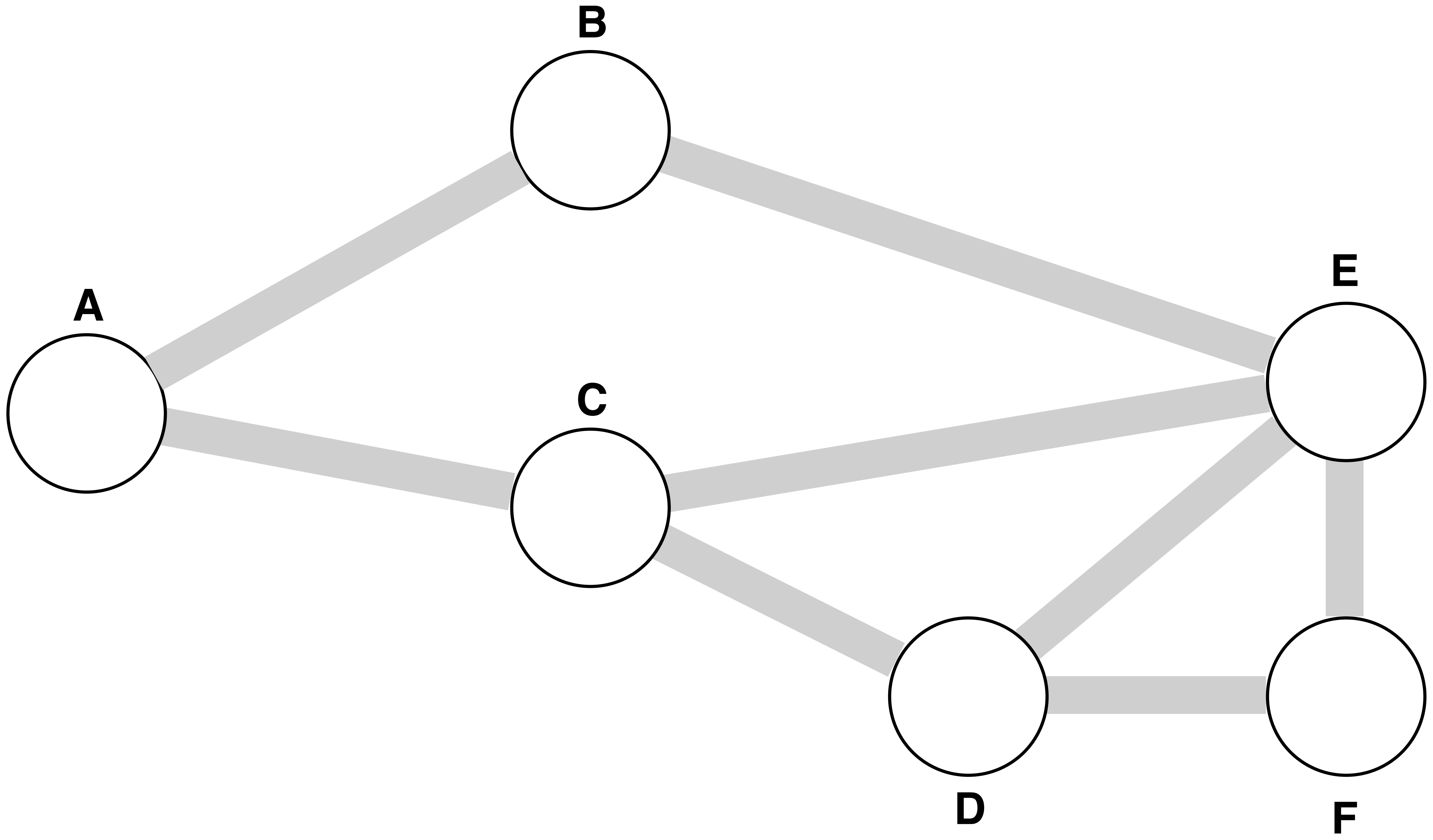
Wide Area Network Traffic Engineering

Future Internet Project

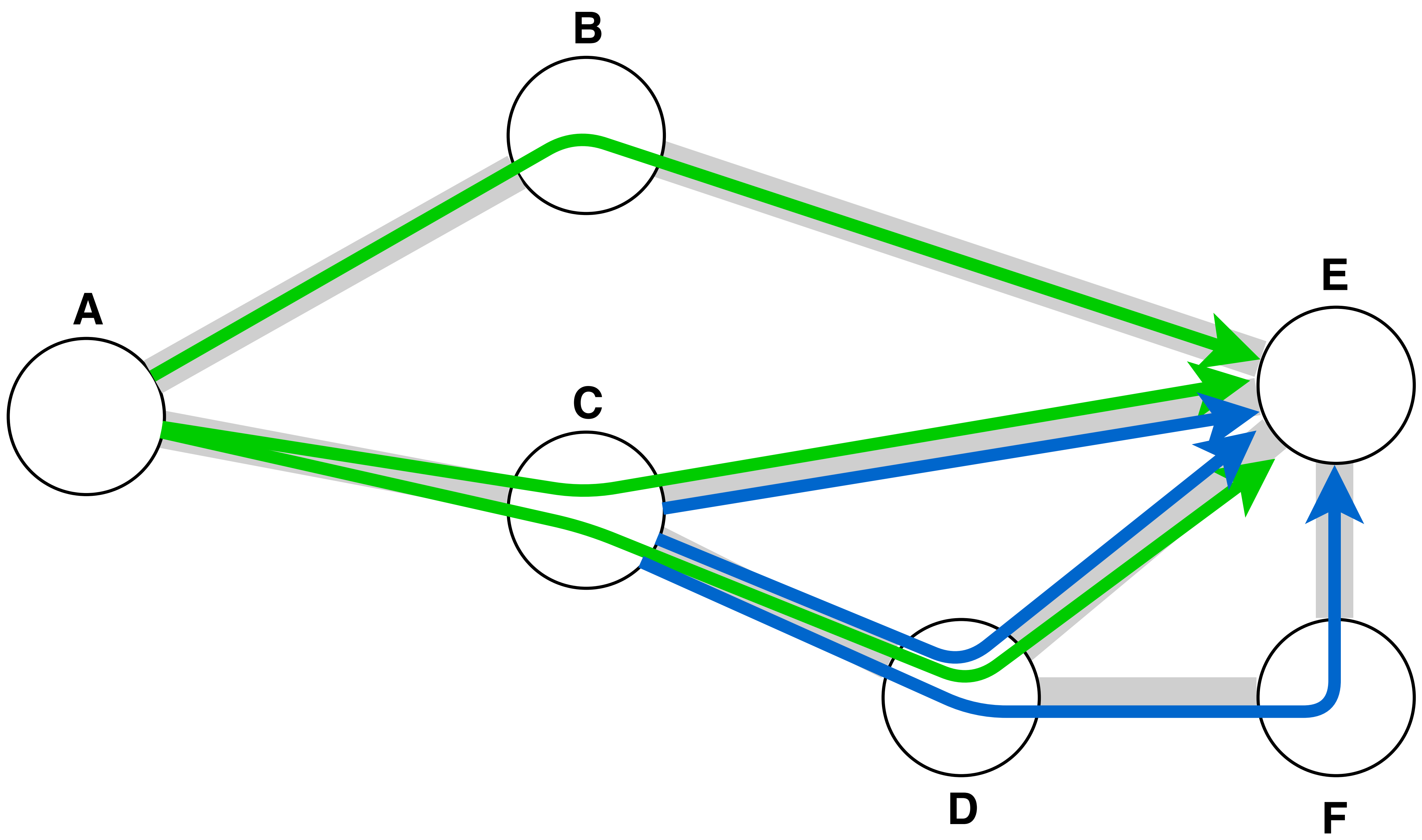
General

- Groups
- 12.5 points
- Part A: 3pt
- Part B: 3pt
- Part C: 6.5pt

Two demands: **A → E and C → E**



Three paths each for A → E and C → E



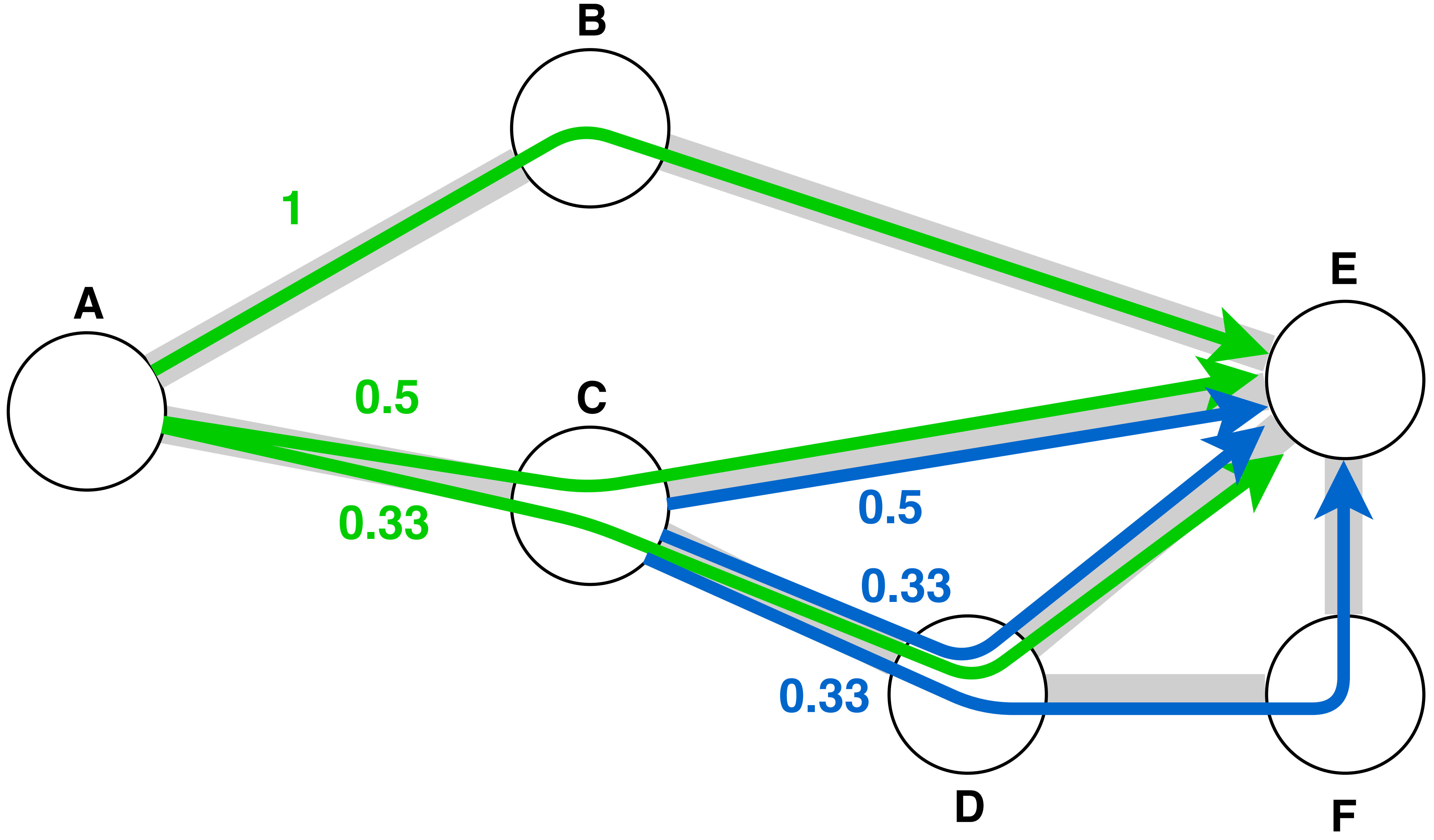
Decentralized allocation: play fair together

- Part A shows how TCP flow single path flows would ideally converge:
    - “fairness” is achieved by everyone using an as-aggressive congestion control mechanism
    - Deploying multiple flows as the same aggression and combining them application-layer can give you an edge (browsers do this)

Part A: **Max-min fair sharing**

A → E : 1.833

C → E : 1.166



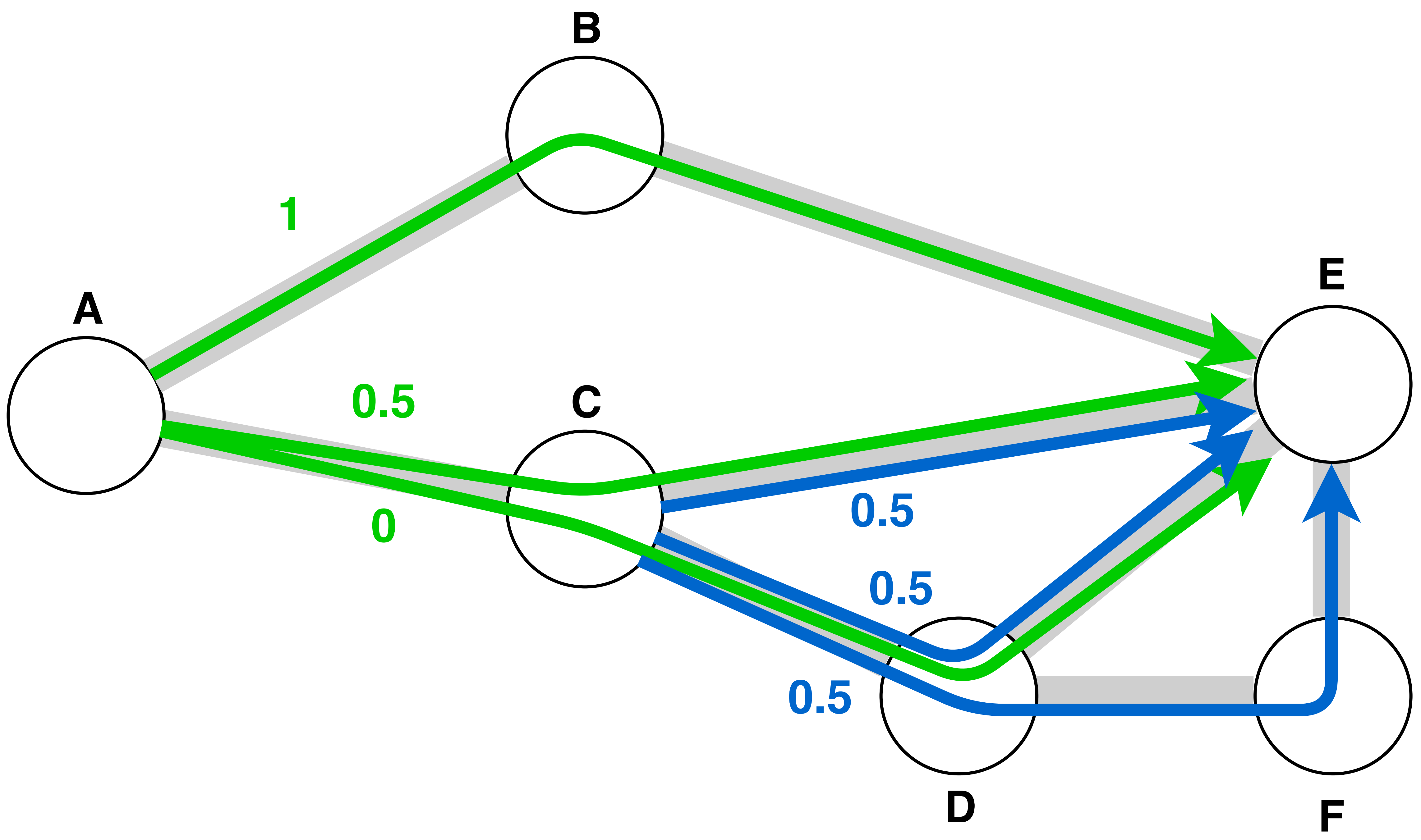
Central rate allocation: efficient but slow

- Part B shows how a central rate allocator can guarantee some definition of fairness
- However, a linear program at large scale is time costly!
- In reality: updates of new flows arriving must come in, and must be assigned a rate on-time

Part B: **Finding the optimum allocation**

A → E : 1.500

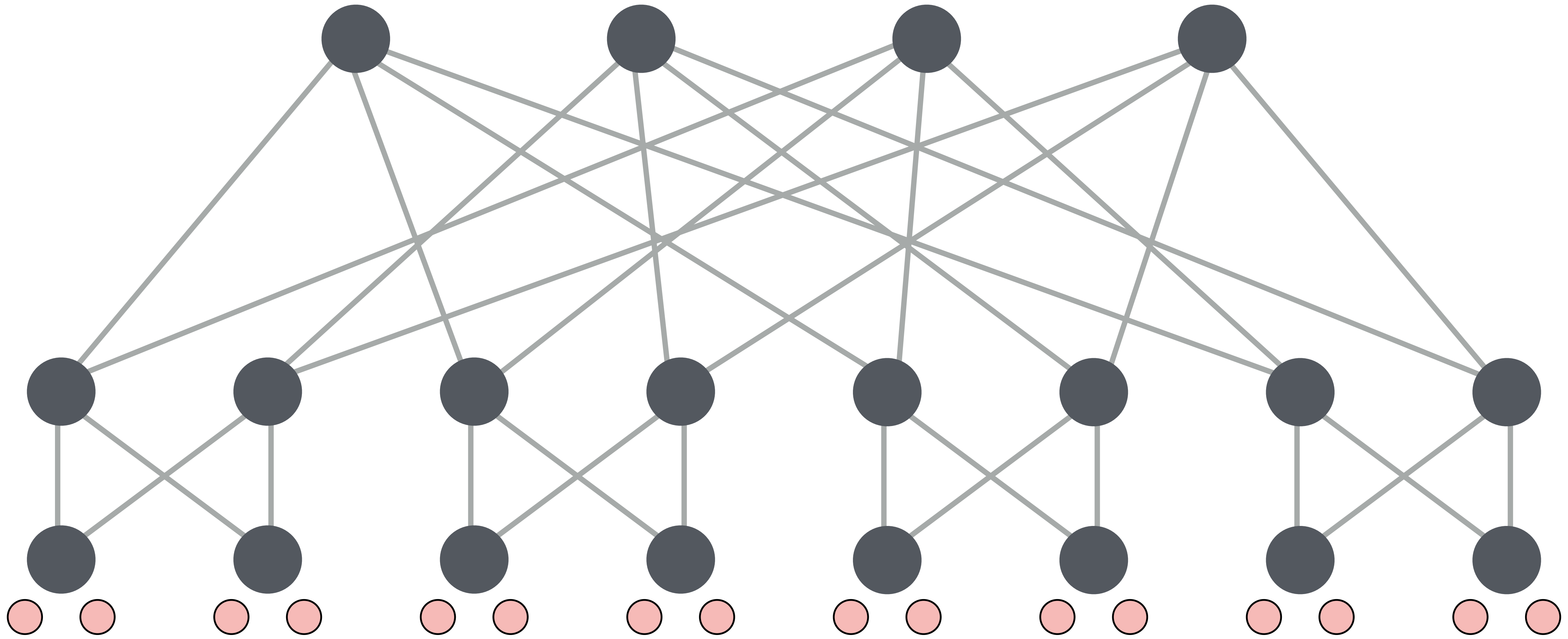
C → E : 1.500

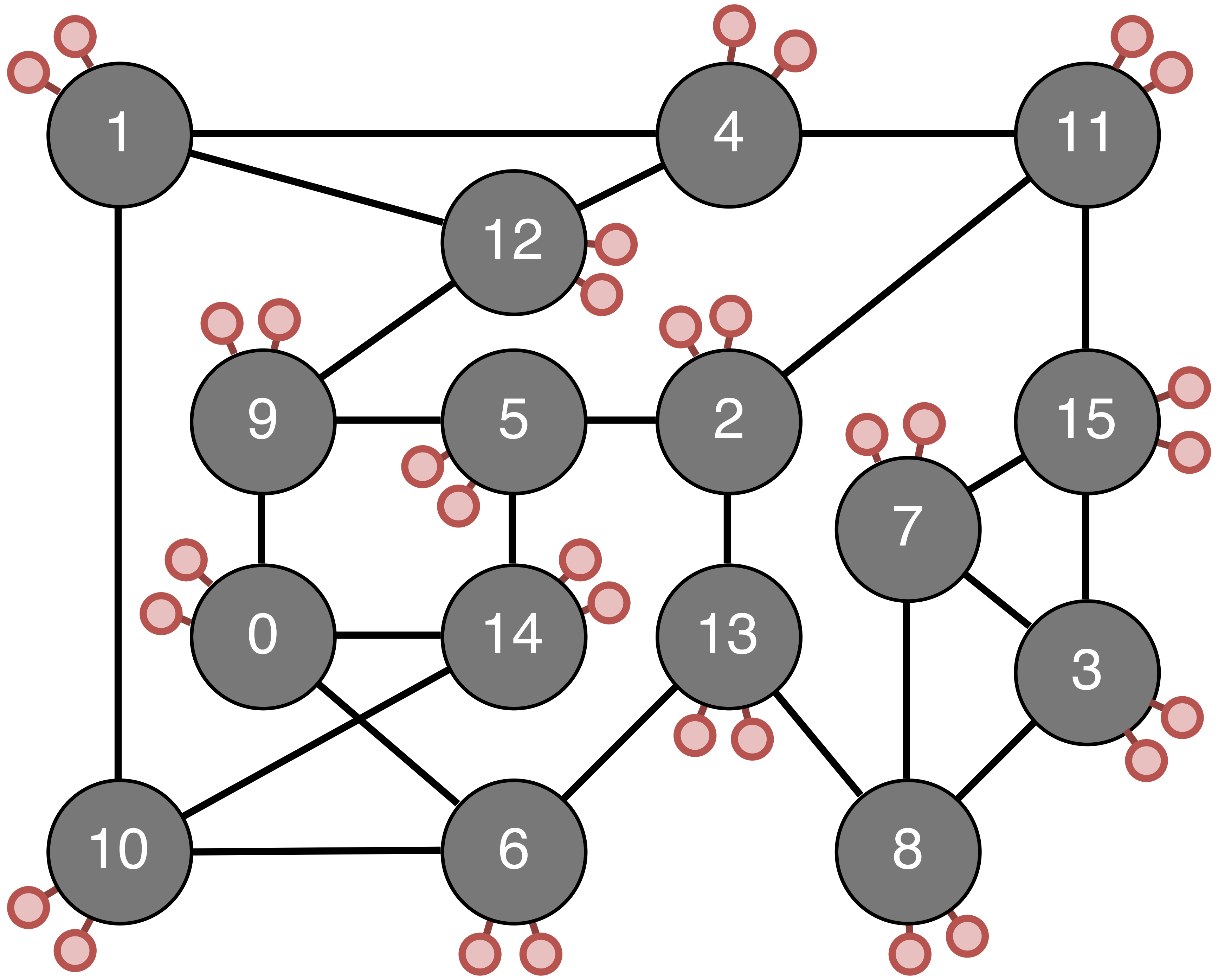


What is the best possible graph out there for any demand matrix?

- This does not exist.
- However, what if we restrict ourselves such that you want it to be as good (or well, as bad) for all possible demand matrices? What is the ultimate average solution?
- A random regular graph (a.k.a. “Jellyfish”) intuitively guarantees that every single node is as well connected to any other node; in a perfect random graph: **there are no weaknesses**

Path diversity is no longer out of the box

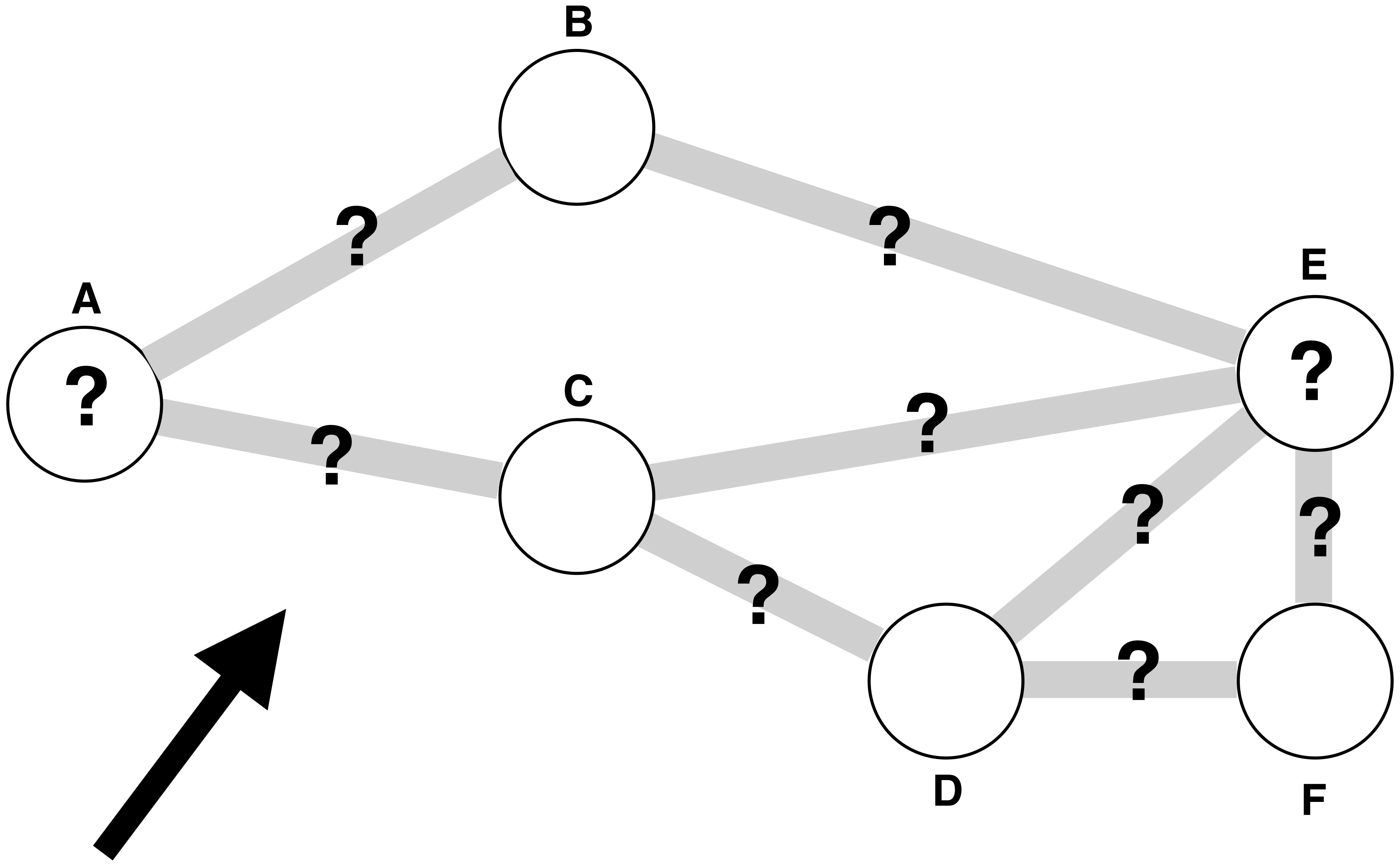




Part C: **Finding the optimal paths**

- The graph is a random regular graph (a.k.a. “Jellyfish”)
- The demand matrix is *intuitively* hard: it forces pairs to communicate solely between each other
- The demand matrix does not give you any path diversity out-of-the-box: you must find it

Part C: **What are the best K paths to select under optimal allocation?**



Random regular graph

Part C: **What are the best K paths to select under optimal allocation?**

- There are 50 random regular graphs of n=120, d=8
- Your goal is to calculate the best paths between all pairs
- On the leaderboard there are two scores:
  - C-public: you have the demand matrix
  - C-hidden: you don’t have the demand matrix
- For C-hidden, we run the LP over your paths and it is updated not as frequent. C-hidden is your score.

Final remarks

- This is the task, looking forward to track your progress on the leaderboard! I recommend you start early: do not leave this project to the last moment.
- Read the entire project assignment PDF
- Pay attention to the file formats! Think about which data structure would be convenient to support printing
- Feedback on the LP Python solver is welcome!
- Feedback on the project in general is welcome!